

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 – 2012

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

СЕКЦИЯ

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

подсекция

**ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ И
АВТОМАТИЗИРОВАННЫХ СИСТЕМ**

Барнаул – 2012

УДК 004

IX Всероссийская научно-техническая конференция студентов, аспирантов и молодых ученых "Наука и молодежь – 2012". Секция «Информационные технологии». Подсекция «Программное обеспечение вычислительной техники и автоматизированных систем». / Алт. гос. техн. ун-т им. И.И.Ползунова. – Барнаул: изд-во АлтГТУ, 2012. – 103 с.

В сборнике представлены работы научно-технической конференции студентов, аспирантов и молодых ученых, проходившей 27 апреля 2012 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

Акинъшин А.А., Голубятников В.П. Математическое и численное описание фазовых портретов некоторых нелинейных динамических систем.....	6
Бутин А.Ю., Крайванова В.А. Комплекс сбора статистической информации с автоматических телефонных станций	9
Речкунов Д.И., Боровцов Е.Г. Разработка и реализация системы контроля технологических процессов на основе платформы Node.js.....	11
Казанцев М.В., Леонов С.Л. Дифференциальная диагностика заболеваний по данным компьютерных томограмм	14
Ворона О.И., Леонов С.Л. Анализ данных компьютерных томограмм	16
Гордиенко Е.Ю., Сучкова Л.И. Разработка программного обеспечения для проведения капнографии пациентов на основе оем-модуля измерения параметров газообмена	17
Витвицкий А.А., Сучкова Л.И. Разработка среды для клеточно-автоматного моделирования на базе механизма параллельных подстановок	19
Слепенков М.И., Сорокин А.В. Разработка программного обеспечения для автоматизации документооборота складского учета государственной аптеки	22
Кротова А.О., Крючкова Е.Н., Крайванова В.А. Проектирование и реализация системы управления семантическим контентом на основе обработки естественных языков.....	23
Удалова А.В., Ананьев П.И. Автоматизация процесса приема заявок на производство в торгово-производственной компании «Завод окон О»	26
Зинченко Д.А., Крайванова В.А. Определение паттернов поведения пользователей на сайте	28
Поканов Ю.А., Сорокин А.В. Разработка системы автоматизированного учета и инвентаризации компьютерной техники.....	30
Трофимчук А.А., Вальков Д.В. Разработка программного комплекса для предоставления информации мобильным клиентам.....	31
Малахов Д.Е., Крючкова Е.Н. Некоторые вопросы разработки системы выявления плагиата в исходных кодах на языке программирования высокого уровня C++	33
Теплякова М.С., Крючкова Е.Н. Использование ассоциативных связей в задачах поиска .	37
Краюшкин М.С., Крайванова В.А. Алгоритм расстановки вершин графа на плоскости, основанный на применении физического взаимодействия.....	38
Фаст Ю.С., Ананьев П.И. Учебные планы ФГОС ВПО третьего поколения в АлтГТУ им. И.И. Ползунова	40
Есипенко С.П. Подходы к решению задачи поиска пары ближайших точек в пространствах с большой размерностью	43
Цисык Р.О., Крючкова Е.Н. Архитектура распределенной системы для обработки событий и сообщений в режиме реального времени.....	44
Деменева Н.В., Сучкова Л.И. Автоматизация учета показателей внеурочной деятельности в рамках муниципальной системы оценки качества образования	48

Избышев А.О., Крючкова Е.Н. Методы перехвата системных сообщений и вызовов в ОС Windows NT	50
Петренко О.Р., Бубнова Н.Д. Разработка автоматизированной системы по работе с пациентами больницы (на примере краевого государственного бюджетного учреждения здравоохранения «Центральная районная больница имени А.Ф. Воробьева»).....	53
Овчинникова А.А., Бразовская Н.В. Разработка структуры финансово – коммерческих операций	56
Шатилина Ю.Е., Крючкова Е.Н. Реализация системы проектирования и тестирования каркасов программных систем	58
Трегубова Ю.Б., Пышнограй Г.В. Стохастическая динамика линейной макромолекулы ...	60
Савченко В.В. , Крючкова Е. Н. Применение нейронных сетей для рубрикации текстов на естественном языке	63
Мезенцев В.Ю., Сучкова Л.И. Разработка инструментальной системы для исследования нечетких временных рядов с применением правил свободной семантики	64
Козлова И.И., Сучкова Л.И. Разработка модуля визуализации для учета посещения массовых мероприятий с произвольной схемой размещения зрителей.....	66
Черкашин А.А., Бубнова Н.Д., Ловцкая О.В. Разработка ГИС на базе картографического сервера с открытым кодом.....	68
Пайвин М.И., Андреева А.Ю. Разработка модуля препарирования изображений для системы распознавания на базе Web-интерфейса	69
Троицкий Д.В., Андреева А.Ю. Разработка программного обеспечения для исследования методов предварительной сегментации символов текста	71
Смирнова Е.Ю., Бубнова Н.Д. Технорабочий проект комплекса прикладных задач «Оперативное управление основным производством» (На материалах компании «Западно-сибирские железные дороги моторвагонное депо ст. Алтайская» ОАО РЖД)	73
Юношев Я.Ю., Бубнова Н.Д. Разработка технического проекта «Автоматизация учета и анализа электрической энергии» (на материалах ОАО «ВЭС»).....	75
Устинов А.В., Старолетов С.М. Реализация интерактивного общения между пользователями социальной сети вуза включая видео-семинары.....	77
Поповцев А.В., Сорокин А.В. Разработка программного обеспечения для реализации интервальных вычислений	79
Горяев С.Л., Егорова Е.В. Разработка системы автоматизированной подготовки и размещения сборников научных конференций на Web-сайте АлтГТУ	80
Лелетко В.В. Подходы к разработке изометрического движка для создания интерактивных приложений.....	81
Селезнев К.К., Сорокин А.В. Разработка программного обеспечения для реализации системы тестирования студентов.....	84
Норсеев А.В., Ананьев П.И. Разработка программного обеспечения для формирования статистической отчетности по рынку труда Кыргызской республики	87
Вотьяков Р.А., Старолетов С.М. Новый подход к программированию искусственного интеллекта.....	88

Сикерин А.С., Крючкова Е.Н. Параллельный автомат как средство анализа недетерминированного поведения параллельных систем	90
Силин Д.И., Уваров Д.А., Крючкова Е.Н. Эффективные алгоритмы обработки Mind Map	97
Барильник К.С., Чемяков А.О. Создание дистрибутива на основе свободно-распространяемого программного обеспечения для реализации учебной деятельности школьника	98
Гичёва Н.И., Кизбикенов К.О. Задача о фигуре, содержащейся в любом треугольнике единичной ширины	99
Дерипаскин В.С., Ефремова Е.Н., Мальцев Ю.Н. Об одном неравенстве и некоторых тождествах треугольника	100
Кадиров Р.В. О некоторых неравенствах в геометрии треугольника	102
Мальцев Р.А., Мальцев Ю.Н. Об одной задаче из журнала «Сгих Mathematicorum»	103

МАТЕМАТИЧЕСКОЕ И ЧИСЛЕННОЕ ОПИСАНИЕ ФАЗОВЫХ ПОРТРЕТОВ НЕКОТОРЫХ НЕЛИНЕЙНЫХ ДИНАМИЧЕСКИХ СИСТЕМ

Акинъшин А.А. – аспирант, Голубятников В.П. – д.ф.-м.н., профессор*
Алтайский государственный технический университет (г. Барнаул)
Новосибирский государственный университет* (г. Новосибирск)

Введение

Среди задач биоинформатики одной из наиболее интересных является задача изучения периодических режимов функционирования генных сетей, которые соответствуют различным биоритмам организма. В настоящей работе рассматриваются некоторые классы моделей генных сетей. В качестве приложений часто рассматриваются циклические системы с простейшими комбинациями отрицательных и положительных обратных связей. Подробный анализ такого рода систем можно найти в [1,2].

Анализ генных сетей является очень важной задачей. В качестве примеров практических приложений исследований можно привести: изучение механизма циркадных ритмов, анализ апоптоза и геномной нестабильности, исследования гемопоэза, лечение заболеваний крови, терапия онкологических заболеваний, терапия нейродегенеративных заболеваний.

В процессе анализа естественным образом возникает обилие подзадач: нахождение стационарных точек и периодических траекторий, исследование их свойств, описание фазовых портретов исследуемых моделей, определение их качественных характеристик.

Ручные методы обчёта зачастую являются непригодными, ввиду их сложности и объёма. Поэтому в работе создаётся специализированный программный комплекс для автоматизированного анализа исследуемых моделей. Но прежде необходимо проработать математический аппарат, который послужил бы основой для компьютерного моделирования.

Математическое моделирование

В настоящей работе рассматриваются произвольные нелинейные динамические системы химической кинетики. Каждая такая система задаётся системой дифференциальных уравнений, определяющих, каким образом скорость изменений концентраций одних веществ зависит от концентраций других веществ.

Особый интерес при исследовании таких систем представляют стационарные точки (они соответствуют гомеостазу организма) и циклические траектории (они соответствуют разным биологическим циклам, например, циркадным ритмам).

Рассмотрим некоторую стационарную точку и её инвариантную окрестность. Проведём через эту точку n гиперплоскостей, параллельных координатным гиперплоскостям. Они разобьют окрестность на 2^n частей, которые мы будем называть кластерами. Кластер примечателен тем, что внутри него производные всех функций в правых частях уравнений сохраняют свой знак. Стоит отметить, что поиск таких кластеров в случае произвольной системы является достаточно сложной математической задачей. Для простых систем часто удаётся выписать уравнения, определяющие область кластера. Анализ кластеров является достаточно важной задачей, т.к. позволяет описать качественное поведение траекторий вблизи стационарных точек.

На границе двух кластеров векторное поле, соответствующее динамической системе, ориентировано в одну сторону по всей области соприкосновения. Таким образом, можно говорить о графе кластеров. Удобным будет рассмотрение траекторий, возникающих в системе, как движение по графу кластеров. В силу однородности внутри отдельно взятого кластера не может существовать замкнутых траекторий. А значит, каждому циклу динамической системы будет соответствовать цикл в графе кластеров. Поэтому поиск замкнутых траекторий удобней начинать с анализа такого графа.

Для более детального изучения окрестности стационарной точки строится линеаризация системы с последующим нахождением собственных чисел и векторов. При построении проекций траекторий на плоскости, построенные на линейных комбинациях собственных векторов, получаются очень показательные иллюстрации, дающее представление об устройстве векторного поля. Собственные числа тоже содержат в себе ценную информацию. Если все действительные части собственных чисел меньше нуля, то стационарная точка является притягивающей. Более интересным является случай отталкивающей стационарной точки. В этой ситуации благодаря некоторым дополнительным рассуждениям часто удаётся доказать существование по меньшей мере одного цикла вблизи рассматриваемой точки. В некоторых частных случаях даже для систем небольших размерностей доказывалось существование двух или трёх циклов, а также делаются выводы об их устойчивости.

Стоит заметить, что так же очень важными являются вопросы о количестве циклов и об их устойчивости. Но для ответа на них приходится использовать куда более сложные математические рассуждения.

Компьютерное моделирование

Для автоматизации описанного выше анализа динамических систем была разработана специальная компьютерная программа.

Программа позволяет выполнять следующие действия:

- *Базовое конструирование системы.* Задание размерности системы и исследуемых функций в аналитическом виде. Для задания системы можно использовать определённые пользователем переменные и вспомогательные функции.
- *Анализ стационарных точек.* Поиск точек, построение линеаризации системы, поиск собственных чисел и векторов.
- *Моделирование траекторий.* Задание характеристик генерации стартовой точки, алгоритмы автоматического подбора шага.
- *Построение изображений системы.* Отображение стационарных точек и траекторий. Проецирование на координатные 2D и 3D плоскости, построенные на заданных векторах (могут использоваться: декартовы вектора, собственные вектора, их линейные комбинации).
- *Построение графа кластеров.* Кроме аналитического построения графа предусмотрена графическая визуализация.
- *Построение различных графиков.* Рассматриваются всевозможные функции, возникающие в ходе исследования.
- *Сохранение и загрузка системы.* Используются файлы собственного формата, которые можно редактировать как средствами программы, так и в любом текстовом редакторе.
- *Отчёт о проделанных вычислениях.* Предусмотрена генерация отчёта в разнообразных форматах (в том числе, в формате LaTeX).

Программа написана на языке C# под платформу .NET 4 с использованием технологии WPF. Для генерации изображений графов используется пакет graphviz.

Результаты моделирования

Устройства сложных многомерных моделей генных сетей можно изучать по сгенерированным программой графическим представлениям проекций на пространство собственных векторов. Важно иметь возможность посмотреть проекции траекторий на различные плоскости.

На рисунках 1, 2 приведены примеры фазового пространства 15-мерной системы. На рисунке 3 изображён фрагмент графа кластеров 9-мерной системы. На рисунке 4 продемонстрирован интерфейс программы.

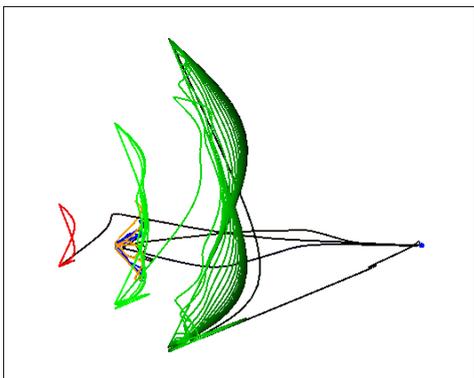


Рисунок 1 – Распределение нескольких циклов

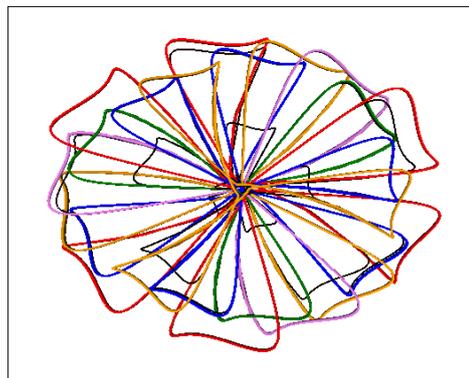


Рисунок 2 – Пример сложного аттрактора

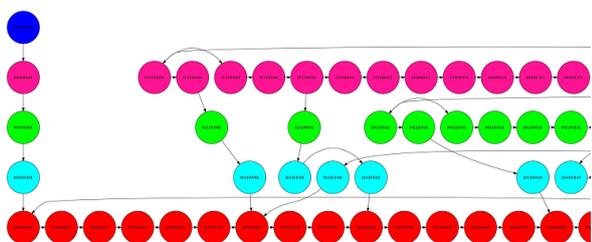


Рисунок 3 – Фрагмент графа кластеров

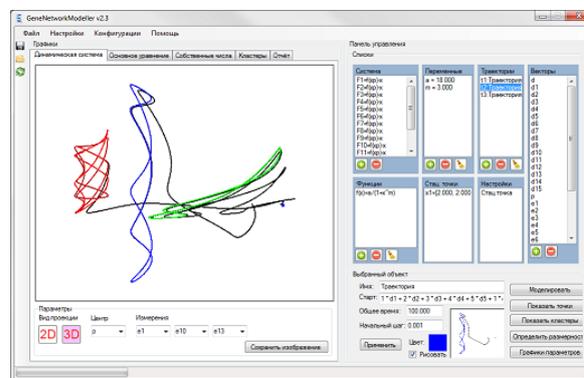


Рисунок 4 – Главное окно программы

Перспективы развития программы

К настоящему моменту программа обладает достаточно богатыми функциональными возможностями. Но, тем не менее, количество задач биоинформатики и их вычислительная сложность настолько велики, что рамки развития программы практически не ограничены. Рассмотрим список того, что хотелось бы реализовать в ближайшем будущем:

- *Поддержка запаздывающего аргумента.* В программе рассматриваются произвольно заданные функции на основе их текстового аналитического представления, что позволяет конструировать практически произвольные динамические системы. Однако, в реальных приложениях реализация обратных связей не является мгновенной: эффекторы оказывают воздействие на определённый белок по прошествии некоторого интервала времени. Поэтому необходимо включить в систему формирования функции запаздывающий аргумент.
- *Углубленное исследование топологии генных сетей.* Существует много сложных для вычисления задач от поиска полного списка всех траекторий системы до определения их бассейнов притяжения.
- *Повышение точности расчётов.* В некоторых случаях достаточно при погрешности задания начальной точки даже в 50-ом знаке качественная картина фазового портрета совершенно изменяется. Для обхода таких ситуаций необходимо использовать сложные алгоритмы вычислительной математики.

- *Работа с существующими базами данных генных сетей.* Существует множество баз реальных генных сетей, записанных на языках формализованных описаний. Предполагается реализовать импорт из этого формата.

Список литературы

1. Системная компьютерная биология. Интеграционные проекты. Выпуск 14. Редколлегия: Н.А.Колчанов, С.С.Гончаров, В.А.Иванисенко, В.А.Лихошвай. Новосибирск: Издательство СО РАН, 2008.
2. Gaidov Yu.A., Golubyatnikov V.P. On the existence and stability of cycles in gene networks models with variable feedbacks // *Contemp. mathematics*, 2011, 553: 61-74.

КОМПЛЕКС СБОРА СТАТИСТИЧЕСКОЙ ИНФОРМАЦИИ С АВТОМАТИЧЕСКИХ ТЕЛЕФОННЫХ СТАНЦИЙ

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

Операторам, работающим с телефонными станциями, необходимо отслеживать статистику подключений: количество успешных занятий, распределение неуспешных соединений по различным причинам, вычисление часа наибольшей нагрузки на АТС. Удобнее всего это делать с персонального компьютера или ноутбука. Использование компьютера позволяет автоматизировать большую часть рутинных операций по расчету различных показателей на основе полученной статистики.

Разработанная клиентская система позволяет, подключаясь к станции с помощью стационарного ПК или ноутбука, собирать информацию о ее работе за продолжительный период времени. Также система выполняет операции агрегации и анализа полученной информации на предмет определения ЧНН (часа наибольшей нагрузки). Программа осуществляет ежедневную выдачу отчетов в xls формате о ЧНН включающих в себя суммарные значения счетчиков за ЧНН. Программный продукт реализован на платформе Java. Все настройки клиентской системы хранятся в xml файлах. В качестве протокола связи станция использует SNMP (простой протокол управления сетью). Этот протокол является одним из первых протоколов управления, в настоящий момент актуальной является третья версия протокола, поддерживаемая в очень большом количестве устройств.

Также для проверки работоспособности клиентской системы был разработан эмулятор ЦАТС mc240, который позволяет продемонстрировать работу клиента без использования реальной АТС. Задача эмулятора – генерировать новые значения показателей работы станции, эмулируя различные ситуации, возникающие при работе ЦАТС в штатном режиме. Это могут быть такие штатные ситуации как перегрузка оборудования, отказы в обслуживании, выход из строя каких-то абонентских линий, суточный трафик телефонных разговоров и т.д.

Взаимодействие системы сбора статистики и стационарного эмулятора осуществляется по протоколу SNMP, также как и взаимодействие с реальной ЦАТС. Все значения, получаемые от АТС, кодируются в шести SNMP-таблицах:

- PlanIn – Входящие вызовы планов нумерации;

- PlanOut - Исходящие вызовы планов нумерации;
- TgIn - Входящие вызовы транк-групп;
- TgOut – Исходящие вызовы транк-групп;
- TgTrIn - Входящие транзитные вызовы транк-групп;
- TgTrOut - Исходящие транзитные вызовы транк-групп.

Каждая строка этих таблиц определяет или вызовы для определенной транк-группы (группы каналов по которым происходит переброска транзитного соединения между абонентами разных станций), или все осуществлённые через эту станцию вызовы для конкретного плана нумерации.

Каждая строка таблицы содержит следующие показатели:

- Count – Общее число занятий;
- Answer – Число занятий, окончившихся ответом;
- Uncomplete – Число неуспешных занятий (неполный номер);
- Unassigned – ЧНЗ номер не назначен;
- Unspecified – ЧНЗ другая причина;
- Busy – ЧНЗ из-за занятости вызываемого абонента;
- OutOfOrder – ЧНЗ из-за неисправности окончного оборудования;
- Unanswer – ЧНЗ из-за не ответа вызываемого абонента;
- Unavailable – ЧНЗ из-за недоступности промлинии;
- Access – ЧНЗ в доступе отказано;
- Vchan – ЧНЗ — из-за недоступности V-channel;

Система агрегирует полученную информацию в базу данных и вычисляет необходимые статистические таблицы:

- Отчет по потерям - для входящей, исходящей и суммарной нагрузок;
- N5 – для каждой транк-группы собственной и транзитной нагрузки, отдельная строка для всех танк-групп;
- N6 – для каждого плана нумерации и отдельная строка общая для всех планов;

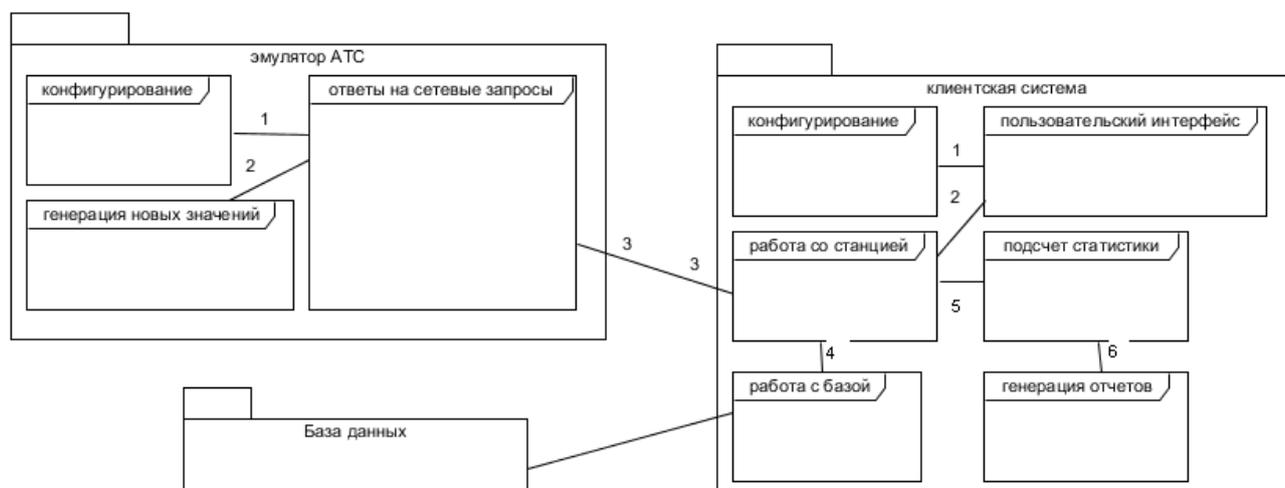


Рисунок 1 – Схема взаимодействия основных модулей системы

Работа клиентской системы происходит следующим образом.

1. Загружаются настройки из xml файлов.

2. Система ожидает времени следующего сеанса со станцией.
3. Со станции считываются новые данные.
4. Новые данные сохраняются в БД.
5. Использую все данные, собранные за новый период расчета, выполняется подсчет статистики.
6. На основании посчитанной статистики генерируются отчеты для планов и транк-групп.

Работа эмулятора АТС происходит следующим образом.

1. Загружаются настройки из xml файлов.
2. Генерируются новые значения счетчиков на основании старых и xml файлов конфигурации.
3. Система ожидает следующего считывания счетчиков.

В дальнейшем планируется разработать Web интерфейс к клиентской системе на Tomcat JSP, что позволит настраивать и получать данные со станции удаленно.

Список литературы

1. Статьи и файлы об snmp протоколе – Портал [Электронный ресурс] - Электрон. дан. — М., [2009-2011]. — Режим доступа: <http://snmp.ru>, свободный. — Загл. с экрана.
2. Java Source code/java Documentation – Портал [Электронный ресурс] - Электрон. дан. — М., [2009-2011]. — Режим доступа: <http://www.java2s.com/Open-Source/Java-Documents/Net/snmp4j/org.snmp4j.agent.htm>, свободный. — Загл. с экрана.
3. Inserting log statements into your code is a low-tech method for debugging it – Портал [Электронный ресурс] - Электрон. дан. — М., [2009-2011]. — Режим доступа: <http://logging.apache.org/log4j/1.2/>, свободный. — Загл. с экрана.

РАЗРАБОТКА И РЕАЛИЗАЦИЯ СИСТЕМЫ КОНТРОЛЯ ТЕХНОЛОГИЧЕСКИХ ПРОЦЕССОВ НА ОСНОВЕ ПЛАТФОРМЫ NODE.JS

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

В настоящее время активно предпринимаются меры по автоматизации различных технологических процессов на предприятиях: будь то производственные предприятия или коммунальные службы. Такие организации располагают множеством объектов, к примеру, скважинами, водонапорными башнями, тепловыми пунктами, автоматизированными станками и конвейерами. Подобные объекты требуют постоянного внимания и контроля, но, в тоже время, постоянное присутствие человека на них нежелательно, а в некоторых случаях является вредным для здоровья. Отсутствие систем удаленной диспетчеризации объектов ведет к излишним экономическим и трудовым затратам, что сказывается на себестоимости производства или предоставления каких-либо услуг. И, напротив, использование систем контроля технологических процессов позволяет обеспечить безопасность персонала, консолидировано наблюдать за группой удаленных объектов в рамках одного рабочего

места, прогнозировать износ оборудования, анализировать эффективность его использования и повышать показатели производительности за счет экономии времени и рабочей силы.

Целью работы является разработка системы контроля технологических процессов в масштабе реального времени в виде веб-сервиса по принципу SaaS (SaaS, англ. Software as a Service).

Современный уровень развития веб-технологий позволяет разрабатывать интерактивные приложения, не уступающие по своему функционалу обычным приложениям, и при этом использование веб-ориентированных систем несет за собой много преимуществ, как для пользователя, так и для разработчика:

1. отсутствие необходимости установки дистрибутива приложения на компьютер пользователя;
2. возможность использования приложения на большинстве операционных систем, включая мобильные устройства;
3. поддержка централизованной системы обновлений приложений;
4. удобство оказания технической поддержки (так как версия приложения у всех гарантировано одна и та же);
5. гибкость разработки графического интерфейса, использование дизайнерских решений;
6. низкая стоимость сопровождения.

При таком подходе мы должны решить задачу разработки веб-приложения, способного обслуживать большое количество пользователей, которые должны с минимальной задержкой получать данные об изменении показателей датчиков на их объектах и наблюдать наглядную визуализированную мнемосхему технологического процесса.

Для решения такой задачи было принято решение использовать платформу `node.js`. Платформа представляет собой фреймворк для разработки серверных приложений на языке программирования JavaScript. Node.js имеет архитектуру, позволяющую справляться с большими нагрузками, имеет в своем составе реализации основных сетевых протоколов, алгоритмов криптографии, средства для работы с файловой системой, а также имеет в своем репозитории множество готовых библиотек-модулей для работы с базами данных, прикладными протоколами, такими как XMPP, IRC и многое другое. Основным фактором, определяющим выбор этой платформы, является библиотека Socket.IO, разработанная для платформы `node.js` и обеспечивающая связь между серверным и клиентским веб-приложениями по наиболее эффективному протоколу, поддерживаемому браузером клиента. Если это современный браузер, то соединение будет устанавливаться по WebSockets, если браузер их не поддерживает, но поддерживает Flash, то будут использоваться сокет технологии Flash, иначе JSONP или, в крайнем случае, бесконечный `iframe`. Библиотека поддерживает множество браузеров, в том числе и мобильные платформы, а также обеспечивает надежный канал для обмена данными в реальном масштабе времени.

Заметим, что процесс `node.js` может удерживать тысячи одновременных соединений, но, тем не менее, все операционные системы имеют ограничение по количеству открытых соединений на одном компьютере, и рано или поздно количество пользователей может превысить этот лимит. Для решения этой проблемы решено использовать распределенную архитектуру веб-сервиса, которая представляет собой кластер с одним мастер-узлом и множеством рабочих узлов, находящихся на разных физических машинах.

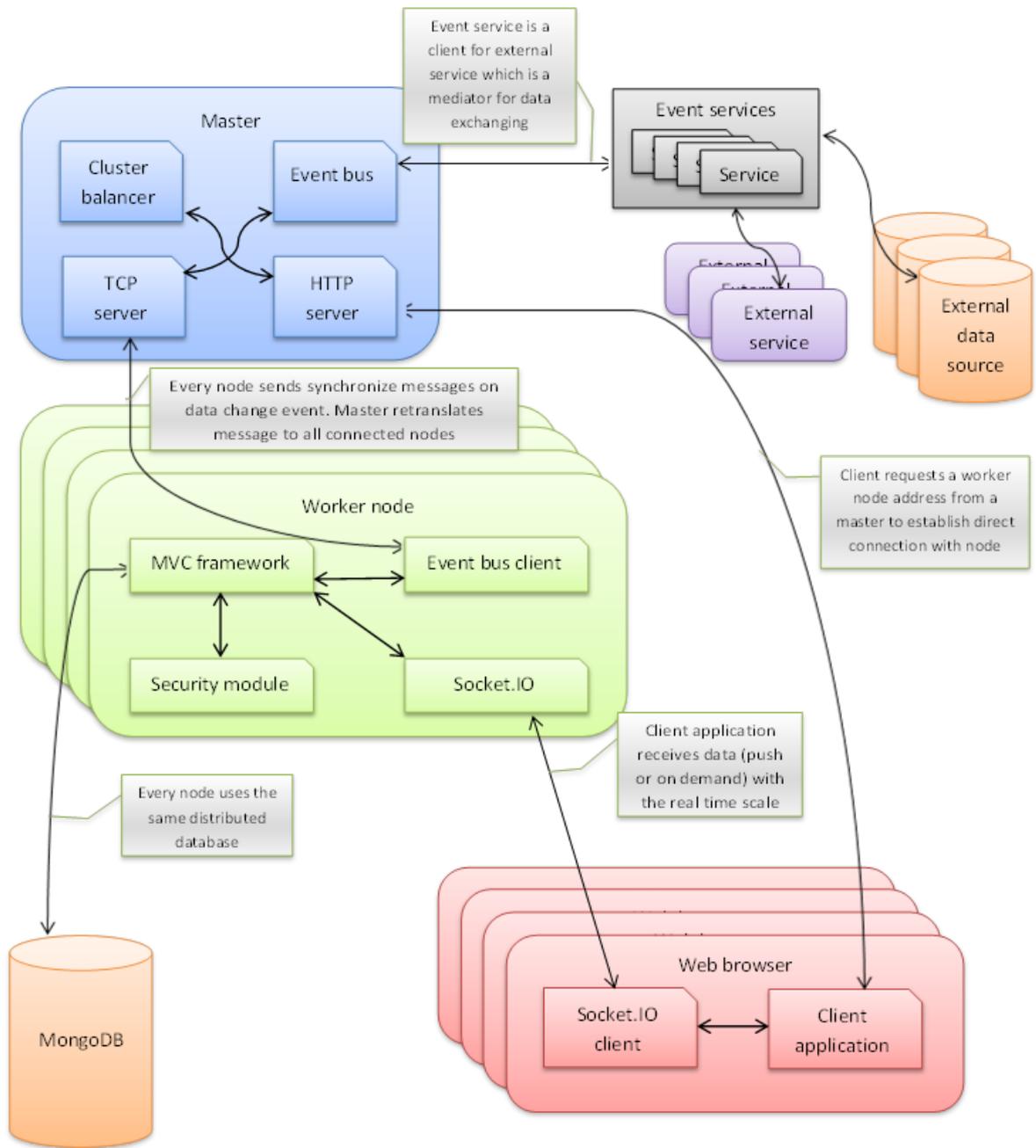


Рисунок 1 – Принципиальная схема архитектуры веб-сервиса

Мастер-узел обеспечивает «шину событий», на которую каждый узел присылает информацию об изменении пользователем данных, а мастер-узел ретранслирует события на все остальные узлы, чтобы синхронизировать состояние системы. Каждый рабочий узел реализовывает шаблон проектирования MVC (MVC, англ. Model-View-Controller), где представлением является клиентское веб-приложение, а модель сама уведомляет представление об изменении своих данных для обновления интерфейса пользователя. Мастер-узел при старте запускает сервисы, которые являются внешними источниками событий для всей системы, а также транслирует входящие события на «шину событий». Внешними источниками событий могут служить такие сервисы как клиент технологического

сервера, который занимается сбором информации от приборов, находящихся на технологических объектах пользователей системы или клиент для сервиса SMS команд, который можно организовать в будущем. Все данные система хранит в NoSQL базе данных MongoDB, которая имеет отличные механизмы масштабирования, распределенную архитектуру, высокую скорость записи, что в нашем случае играет решающую роль. Принципиальная схема веб-сервиса указана на рисунке 1.

Клиентское веб-приложение разработано с использованием фреймворка knockout.js, который позволяет использовать декларативные привязки данных для интерактивного обновления интерфейса пользователя и помогает разрабатывать приложения по шаблону проектирования MVVM (MVVM, англ. Model-View-ViewModel). Такой подход в разработке графического интерфейса хорошо знаком разработчикам WPF/Silverlight и значительно облегчает задачу представления данных.

В результате работы получена система контроля технологических процессов в масштабе реального времени на основе платформы node.js. Система имеет распределенную архитектуру, что позволяет масштабировать ее под возрастающие нагрузки и обслуживать большое количество одновременных соединений для доставки данных с веб-сервиса в клиентское веб-приложение. Исходные коды веб-сервиса распространяются на ресурсе <https://github.com/DenisRechkunov/blinker> под лицензией MIT, как программное обеспечение с открытым исходным кодом.

ДИФФЕРЕНЦИАЛЬНАЯ ДИАГНОСТИКА ЗАБОЛЕВАНИЙ ПО ДАННЫМ КОМПЬЮТЕРНЫХ ТОМОГРАММ

Казанцев М.В. – студент, Леонов С.Л. – д.т.н., профессор
Алтайский государственный технический университет (г. Барнаул)

Дифференциальная диагностика заболеваний предполагает изучение ряда показателей пациента с целью установления диагноза. Среди множества предполагаемых нозологических форм требуется выбрать ту, которая в наибольшей степени соответствует наличествующему набору снятых показателей. Для оценки вероятностей наличия той или иной нозологической формы возможно использовать системы искусственного интеллекта.

В рамках данной работы ставится задача проведения дифференциальной диагностики заболевания по характеристикам, полученным на основании численного анализа компьютерных томограмм. Современное штатное программное обеспечение, которым сопровождаются компьютерные томографы, позволяет рассчитывать количественные характеристики части одного среза и не дают возможности изучать динамику изменения структуры образования в объеме [1]. Методика ориентирована на анализ шаровидных образований в лёгких (рис. 1). Рассматриваются три нозологические формы: рак, пневмония, туберкулёз. Актуальность рассматриваемой задачи заключается в том, что существующий на данный момент набор штатного программного обеспечения для обработки компьютерных томограмм дает возможность производить измерения параметров шаровидных образований, но не позволяет оценивать принадлежность его к одной из нозологических форм.

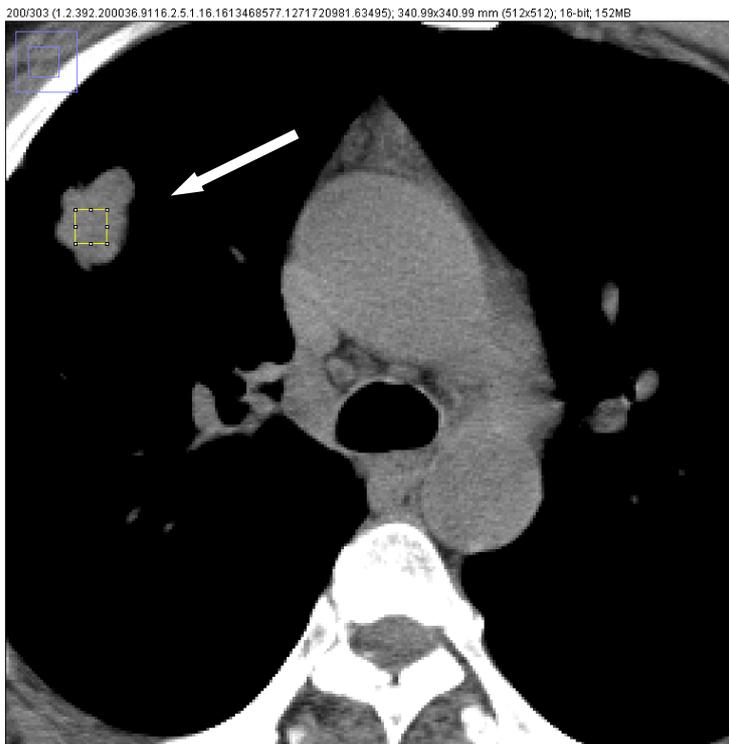


Рисунок 1 – Пример компьютерной томограммы шаровидного образования в легком

Способ решения задачи заключается в следующем:

1. По набору срезов (томограмм) шаровидного образования рассчитываются его объемные характеристики, такие как:
 - 1.1. средние и среднеквадратичные отклонения денситометрических плотностей и динамика их изменения от среза к срезу;
 - 1.2. фрактальные характеристики образований;
 - 1.3. параметры локализации и др.
2. Для решения задачи об отнесении данной томограммы к одной из нозологических форм предлагается использовать искусственную нейронную сеть [2]. Предлагается использовать обучение сети с учителем, используя в качестве обучающей выборки набор данных о пациентах с верифицированными диагнозами.

Для автоматизации анализа описанных выше параметров предлагается разработать следующие программные продукты:

1. Конструктор нейронных сетей, позволяющий выполнять следующие действия:
 - 1.1. Построение нейронных сетей произвольной топологии.
 - 1.2. Обучение построенной сети на выборке данных.
 - 1.3. Тестирование обученной сети на выборке данных и оценку качества её работы.
 - 1.4. Сохранение и загрузку сетей из файла.
2. Прикладное ПО, использующее обученную нейронную сеть для оценки вероятности отнесения набора входных данных к тем или иным нозологическим формам.

Перспективы развития

В перспективе данный подход к решению задачи компьютерной дифференциальной диагностики может быть обобщён на другие нозологические формы, на другие форматы изображений, полученных не только с помощью компьютерного томографа. Конструктор нейронных сетей может быть улучшен добавлением функции автоматического построения сети топологии, которая как можно эффективнее решает данную задачу после обучения на

данной выборке. Для этого предполагается использовать методы эволюционного программирования.

Список литературы

1. Прокоп М., Галански М. Спиральная и многослойная компьютерная томография. Том 1. – М.: МЕДпресс-информ, 2009. – 416 с.
2. Яхьева Г.Э. Нечеткие множества и нейронные сети: Учебное пособие – М: Интернет-Университет Информационных Технологий; БИНОМ. Лаборатория знаний, 2006. – 316 с.

АНАЛИЗ ДАННЫХ КОМПЬЮТЕРНЫХ ТОМОГРАММ

Ворона О.И. – студент, Леонов С.Л. – д.т.н., профессор
Алтайский государственный технический университет (г. Барнаул)

Рентгеновская компьютерная томография – способ визуализации внутренней структуры с получением отдельных срезов вплоть до ее объемной реконструкции [1]. Существующие аппараты для томографии обладают большими возможностями и оснащены мощным штатным программным обеспечением. В качестве примера можно привести программу Vitrea, которая позволяет:

- проводить анализ отдельных срезов тела пациента;
- осуществлять объемную реконструкцию с выводом 3D изображений;
- производить замеры денситометрических плотностей и рассчитывать их стохастические параметры в пределах каждого среза;
- выполнять сравнение отдельных срезов в пределах одного или разных исследований.

Однако данное программное обеспечение недостаточно специализировано в области количественных измерений объемных плотностей образований с получением их стохастических характеристик. Кроме того, штатное программное обеспечение поставляется вместе с самим томографом и не позволяет проводить анализ изображений автономно. Это затрудняет проведение дифференциальной диагностики заболеваний.

Существует свободно распространяемое программное обеспечение для анализа томограмм, которое хоть и обладает меньшими возможностями, но позволяет проводить их анализ независимо от аппаратуры компьютерного томографа. К такому программному обеспечению относится ImageJ. Это общедоступная, основанная на Java программа для анализа и обработки изображений, разработанная в National Institutes of Health. Программа имеет открытую архитектуру, которая обеспечивает расширяемость при помощи плагинов Java, а также записываемых макросов [2].

Планируется доработка данной программы для расчета с ее помощью:

- средних и среднеквадратичных отклонений денситометрических плотностей и динамики их изменения от среза к срезу;
- фрактальных характеристик образований;
- параметров локализации образований и др.

Разрабатываемое программное обеспечение ориентировано на изучение шаровидных образований легких но в перспективе может быть масштабировано на исследование и других органов пациентов.

Список литературы

1. Прокоп М., Галански М. Спиральная и многослойная компьютерная томография. Том.1. – М.: МЕДпресс-информ, 2009. - 416 с.
2. [<http://rsb.info.nih.gov/>] - Ресурс разработчиков ImageJ

РАЗРАБОТКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ДЛЯ ПРОВЕДЕНИЯ КАПНОГРАФИИ ПАЦИЕНТОВ НА ОСНОВЕ OEM-МОДУЛЯ ИЗМЕРЕНИЯ ПАРАМЕТРОВ ГАЗООБМЕНА

Гордиенко Е.Ю. – студент, Сучкова Л.И. – к.т.н., профессор
Алтайский государственный технический университет (г. Барнаул)

Исследование легочной вентиляции играет ведущую роль в дифференциальной диагностике заболеваний дыхательной системы человека. Проведение этого исследования во многих случаях позволяет верифицировать клинический диагноз, спланировать оптимальные режимы лечения и контролировать его эффективность, в известной мере спрогнозировать течение патологического процесса.

Наряду с классической спирографией – традиционным методом определения функции внешнего дыхания, который хорошо зарекомендовал себя на практике, в последнее время появились новые методы как для массового скринингового или индивидуального обследования, так и для использования в стационарных условиях (спирография и капнография с анализом соотношения кривой «поток-объем», бодиплетизмография и пр.). Капнограмма, кривая изменения концентрации CO_2 во времени, является важным диагностическим средством, так как ее форма практически одинакова у здоровых людей. Поэтому следует анализировать любое изменение формы капнограммы. Современная медицинская техника содержит микропроцессор или соединена с персональным компьютером, использующим для анализа функции внешнего дыхания многие параметры и показатели, которые обязательно фиксируются и/или записываются в память компьютера [1].

С возникновением и развитием вычислительной техники появилась возможность переложить все больше функций по анализу и интерпретации данных, получаемых при обследовании, на компьютеры. Это позволяет не только облегчить труд специалистов в области диагностики состояния дыхательной системы, но и сократить время обработки и количество человеческих ошибок, возникающих в ходе обследований.

В ходе работы были рассмотрены программные разработки, позволяющие проводить капнографию пациентов, такие как Лечебно-диагностический комплекс «Карбоник» и сопровождающее программное обеспечение, а также тестовое программное обеспечение, поставляемое в комплекте с модулем измерения параметров газообмена «Микролюкс OEM_CO2+O2» (разработан ООО «МИКРОЛЮКС», Россия, Челябинск) [2, 3]. Основными недостатками обоих аналогов является недоступность исходного кода для корректировки и расширения функционала, невозможность доступа к данным, передающимся от прибора для их анализа и обработки, отличной от стандартно предоставленной, несоответствие внешнего вида приложений современным требованиям к пользовательскому интерфейсу, а также наличие ошибок в работе программ.

Рассматриваемый в данной работе OEM-модуль – это полнофункциональный модуль мониторинга всех основных параметров газообмена, в том числе:

- $EtCO_2$ (Концентрация CO_2 в конце выдоха).
- $FiCO_2$ (Концентрация CO_2 на вдохе).
- Частоты дыхания.
- Капнограммы.

Во время работы прибора он передает по виртуальному СОМ интерфейсу пакеты, содержащие служебную информацию, а также данные о параметрах газообмена. Для получения данных от модуля главный процессор посылает модулю команду «Запрос данных». В ответ модуль посылает пакет данных фиксированной длины (8 байт), то есть обмен ведется по принципу «Запрос - Ответ». Время ответа модуля после получения запроса (с учетом физического времени передачи всего пакета данных) не превышает 1.5 мс, что позволяет обрабатывать и визуализировать данные в реальном времени. Модуль также посылает пакет данных в ответ на любую другую команду управления, адресованную ему. Любая команда управления также является одновременно и командой запроса данных. [4]

Модуль имеет «неонатальный» и «взрослый» режимы работы. Отличия эти режимов связаны с обработкой сигналов при высокой частоте дыхания. Во «взрослом» режиме максимальная частота дыхания ограничивается значением 90 дых./мин., в «неонатальном» режиме измерения продолжаются до 150 дыханий в минуту. Также поддерживается режим «Горы», который используется в тех ситуациях, когда очень быстро меняется барометрическое давление, например, во время подъема на вертолете санавиации или в горах. В этом случае модуль гораздо чаще (1 раз в минуту) производит автоматическую калибровку нулевой концентрации CO_2 и измеряет текущее барометрическое давление [4].

Гибкая система взаимодействия с модулем с использованием двунаправленного асинхронного последовательного интерфейса (UART) позволяет очень быстро и с минимальными затратами встроить его практически в любую мониторинговую систему. Цифровая система обработки сигналов и мощный RISC-процессор обеспечивают качественное измерение концентрации CO_2 и частоты дыхания в любых клинических ситуациях. Алгоритм обнаружения дыхательных циклов разработан и протестирован с учетом всевозможных аномалий дыхания и различных форм кривых (капнограмм), встречающихся в медицине критических состояний [4].

На основании параметров, получаемых от модуля, можно рассчитывать показатели основного обмена, анализ значений которых делает возможным формирование медицинских заключений и прогнозирование диагнозов.

В ходе данной работы были спроектированы необходимые экранные формы, структура базы данных, обеспечивающая эффективную обработку и хранение данных о пациентах и результатах обследований. В итоге разработан программный продукт для организации обмена данными между модулем «Микролюкс OEM_CO2+O2» измерения параметров газообмена и персональным компьютером, их обработки, анализа и визуализации.

Функционал разработанного программного продукта включает:

- Мониторинг концентрации CO_2 в выдыхаемом пациентом воздухе.
- Мониторинг частоты дыхания пациента.
- Регистрация и сохранение данных капнографических обследований.
- Построение графика капнограммы в реальном времени в ходе обследования.
- Расчет основных показателей функционирования дыхательной системы.
- Формирование заключения о состоянии дыхательной системы пациента на основании данных обследования.
- Управление и настройка OEM-модуля.
- Формирование и сохранение отчетов по обследованиям, а также возможность импорта в Word, Excel.
- Ведение базы данных пациентов и результатов капнографических обследований.

Разработанный программный продукт имеет эргономичный интерфейс, позволяющий пользователям эффективно осуществлять диагностику состояния дыхательной системы пациентов, отслеживать динамику изменений основных показателей, мгновенно после проведения диагностики получать предварительное медицинское заключение по полученным данным.

Можно выделить следующие перспективы развития работы:

- Расширение возможностей программного продукта посредством разработки программных модулей для проведения терапевтических тренировок дыхательной системы, а также повышение интерактивности для ведения «диалога» с пациентом во время таких тренировок.
- Расширение функционала программного продукта для анализа и обработки данных, получаемых от дополнительно подключаемых к рассмотренному модулю датчиков.
- Дополнение программной части, отвечающей за формирование заключений посредством разработки интеллектуальных алгоритмов глубокого анализа получаемых от модуля данных.
- Разработка программных модулей, позволяющих осуществлять формирование индивидуальных тренировок пациента по результатам его диагностик с учетом динамики изменений показателей основного обмена.

Список литературы

1. Окорочков А.Н. Диагностика болезней внутренних органов: Т. 3. Диагностика болезней органов дыхания. [Текст] – М.: Мед. лит., 2000. – 464 с.: ил.
2. Сайт «Микролюкс» [Электронный ресурс] Режим доступа: <http://www.microlux.ru>.
3. Сайт «Карбоник» [Электронный ресурс] Режим доступа: <http://www.carbonic.ru>.
4. OEM-Модуль измерения параметров газообмена: Руководство по применению. [Текст] – Челябинск: Микролюкс, 2008. – 35 с.

РАЗРАБОТКА СРЕДЫ ДЛЯ КЛЕТОЧНО-АВТОМАТНОГО МОДЕЛИРОВАНИЯ НА БАЗЕ МЕХАНИЗМА ПАРАЛЛЕЛЬНЫХ ПОДСТАНОВОК

Витвицкий А.А. – студент, Сучкова Л.И. – к.т.н., профессор
Алтайский государственный технический университет (г. Барнаул)

Введение

В настоящее время существует огромный класс вычислительно трудоемких задач, начиная от обработки изображений и заканчивая моделированием физических и биологических процессов. Структуры данные этих задачи представимы в виде больших массивов из однотипных элементов, а алгоритмы решения, как правило, нуждаются в распараллеливании. Клеточные автоматы (КА), воплощающие в себе однородность данных и естественный параллелизм [1], представляют собой подходящий инструмент для решения подобных задач. К сожалению, промышленному применению КА на сегодняшний момент мешает нехватка полноценных сред разработки и языков программирования, способных предоставить весь потенциал КА.

Целью данной работы являлось создание высокоуровневого языка программирования и среды разработки, служащих для полноценного программирования КА-моделей.

Описание языка

Разработанный язык является Си-подобным компилируемым языком и основан на математической модели Алгоритма Параллельных Подстановок (АПП), позволяющей описывать формально КА-модели [2]. Условно, синтаксис языка можно разделить на две составляющие – синтаксические конструкции языка Си (описание стандартных типов, операторы выбора, циклов и др.) и синтаксис описаний КА, основанный на модели АПП.

Для большей гибкости описание клеточно-автоматных моделей было разделено на три составляющих:

- *Описание клеточных массивов.* Определяет структуры данных, на которых будут работать клеточные автоматы. При создании клеточного массива требуется указать тип решетки (гексагональная, октогональная и др.), форму пространства (тор, плоскость), размеры пространства, а также поля данных, которые будут находиться в узлах решетки. Тип решетки определяет мерность пространства и структуру соседства клеток, в то время как форма пространства задает граничные условия клеточного массива. Поля данных (клетки) могут иметь любой стандартный или пользовательский тип данных. Также существует возможность задавать шаблоны соседства, представляющие собой указатели на определенную группу соседей, и классы клеточных массивов, объединяющие клеточные массивы схожей структуры.
- *Описание клеточного автомата.* Задает правила переходов для клеточно-автоматной модели. Клеточный автомат представляет собой функцию, которая по заданным правилам преобразует указанные клеточные массивы. Описание клеточного автомата состоит из систем параллельных подстановок, функций переходов и контекстных подстановок.
- *Запуск клеточного автомата.* Запуск напоминает вызов функции, в которую передаются фактически параметры и опции, такие как размер эволюции и режим работы КА. Клеточный автомат может работать в одном из 4 режимах работы: синхронный, асинхронный, асинхронный упорядоченный и блочно-синхронный.

Язык также позволяет описывать различные типы композиции КА: последовательную локальную, последовательную глобальную, параллельную однонаправленную и параллельную двунаправленную [3]. Композиция совместно с режимами работы, позволяет создавать различные комбинации клеточно-автоматных моделей (например, параллельно функционирующие взаимодействующие клеточные автоматы). Также существует возможность многопоточного распараллеливания КА-моделей, в том числе и блочно-синхронных клеточных автоматов по алгоритму, предложенному и описанному в [4].

Описание среды разработки

Созданная среда разработки работает в среде Windows и состоит из отладчика, компилятора, генератора кода, компоновщика и RAD-среды. На выходе можно получить как скомпилированную программу в виде exe-файла, так и сгенерированный C++ код, который можно использовать в других приложениях. Генератор кода использует как стандартные методы оптимизации (SSE/SSE2), так и специализированные клеточно-автоматные методы, такие как буфер подкачки (buffer swap), метод разреженных матриц (sparse matrix techniques) и др. Встроенная RAD-среда позволяет быстро создавать оконные приложения и включает в себя такие компоненты как ViewOpenGL, который позволяет визуализировать работу клеточных автоматов используя библиотеку OpenGL. Помимо этого, в среду встроен набор системных функций для работы с КА, таких как булева дискретизация (конвертирование вещественных и целочисленных массивов в булевы), осреднение (обратное предыдущему преобразование) и некоторые другие. Также при написании кода существует возможность делать вставки на языках C++ и assembler.

Тестирование и анализ результатов

Тестирование созданного языка и среды показало, что предложенный язык позволяет полноценно описывать клеточно-автоматные модели и обладает полнотой по Тьюрингу. Процесс программирования клеточных автоматов значительно ускоряется и упрощается, когда с пользователя снимаются такие задачи, как низкоуровневое описание клеточных структур данных (например, описание гектогональной решетки в виде тора и

соответствующей структуры соседства клеток), задание алгоритмов композиции и распараллеливания (например, алгоритм распараллеливания блочно-синхронных клеточных автоматов). Методы низкоуровневой и клеточно-автоматной оптимизации, производимые генератором кода, показали хорошие результаты в скорости работы клеточно-автоматных моделей.

В качестве примера использования разработанного языка и среды приведем описание клеточного автомата, моделирующего химическую реакцию окислению окиси углерода на катализаторе. Код представлен в листинге 1, а результат работы откомпилированной программы приведен на рисунке 1.

Листинг 1

```
CellAuto CO2(TArray1 A, float pa)
{
  System (A) //Система параллельных подстановок, которая
  { //задает правила переходов описываемого КА
    Q1: (0, A.m) * (true, ma) -> (1, A.m);
    Q2: (0, A.m), (0, A.T[k].m) *(true, ma), (out k,mk)->(2,A.m), (2,A.T[k].m);
    Q3: (1, A.m), (2, A.T[k].m) *(out k, mk) -> (0, A.m), (0, A.T[k].m);
    Q4: (2, A.m), (1, A.T[k].m) *(out k, mk) -> (0, A.m), (0, A.T[k].m);
  }
  (bool, ma) //Контекстная подстановка, задающая вероятностное
  { //срабатывание подстановок
    if (randf() < pa) return true;
    else return false;
  }
  (int, mk) //Контекстная подстановка, задающая
  { //случайный выбор соседа клетки
    float k = randf();
    if (k < 0.25) return 1;
    else if (k < 0.5) return 2;
    else if (k < 0.75) return 3;
    else if (k < 1.0) return 4;
  }
};
CO2(A1, pa : CA.Mode.ASYNCH, 1000); //Запуск КА в асинхронном режиме
```

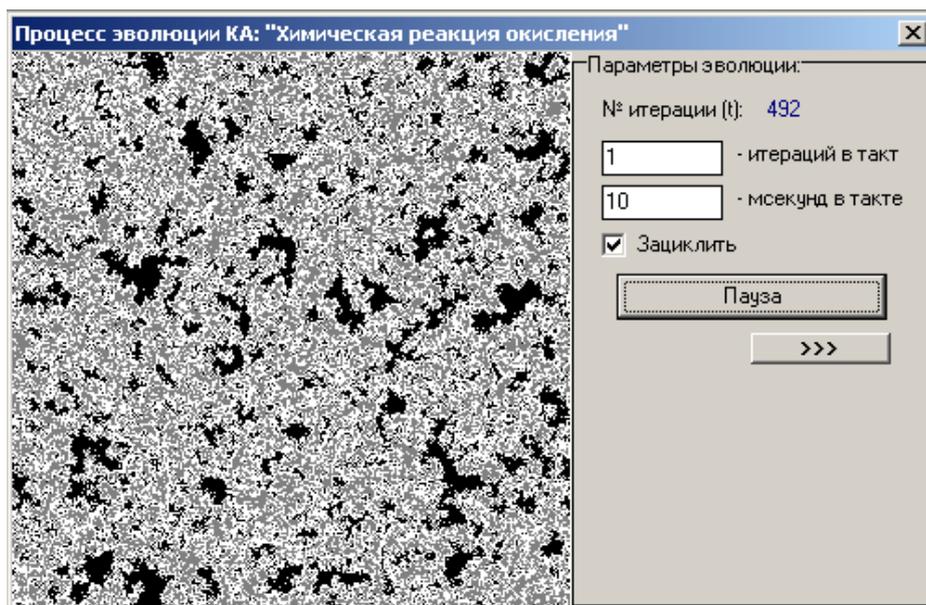


Рисунок 1 – Клеточный автомат, моделирующий химическую реакцию окисления окиси углерода на катализаторе

Заключение

Намечены следующие цели для дальнейшей работы: внедрение возможности распараллеливания работы клеточных автоматов на кластер на основе протокола TCP, поддержка технологии MPI, а также анализ возможностей языка для работы с растровой графикой и алгоритмами вычислительной математики (непрерывные КА).

Список литературы

1. Toffoli T., Margolus N. Cellular Automata Machine. USA: MIT Press, 1987.
2. Achasova S., Bandman O., Markova V. Piskunov S. Parallel Substitution Algorithm. Theory and Application. Singapore: World Scientific, 1994.
3. Бандман О.Л. Метод построения клеточно-автоматных моделей процессов формирования устойчивых структур // ПДМ, 2010, № 4, 91–99.
4. Бандман О.Л. Параллельная реализация клеточно-автоматных алгоритмов моделирования пространственной динамики // Сиб. журн. вычисл. математики / РАН. Сиб. отделение. – Новосибирск, 2007. – Т. 10, №4. – С.335–348.

РАЗРАБОТКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ДЛЯ АВТОМАТИЗАЦИИ ДОКУМЕНТООБОРОТА СКЛАДСКОГО УЧЕТА ГОСУДАРСТВЕННОЙ АПТЕКИ

Слепенков М.И. – студент, Сорокин А.В. – к.т.н., доцент
Алтайский государственный технический университет (г. Барнаул)

Автоматизация в различных сферах деятельности человека посредством разработки специализированного программного обеспечения является одной из важных задач в настоящий момент времени, поскольку и упрощает процесс работы в определенной сфере и в какой-то мере позволяет его ускорить. В данной работе производится исследование возможности процесса автоматизации в сфере документооборота складского учета в государственных аптеках.

Коснемся кратко работы государственных аптек, которые ведут следующую деятельность:

- реализация лекарственных средств населению за наличный расчет;
- реализация лекарственных средств населению по рецептам.

Реализация лекарственных средств по рецептам осуществляется без наличного расчета. Нужно фиксировать информацию по рецептам, чтобы установить однозначное соответствие между отпущенным лекарственным средством и изъятым рецептом. В дальнейшем государство возместит аптеке стоимость лекарственных средств, отпущенных по рецептам. Поэтому, важно вести четкий учет всех лекарственных средств, отпущенных по рецептам.

Другим видом лекарственных средств является коммерческий товар. Его аптека реализует населению за наличный расчет. Необходимо строго разграничивать все лекарственные средства, полученные по разным направлениям, так как они должны быть реализованы населению строго по тем же направлениям.

Для реализации указанных выше операций в государственных аптеках, в данной работе разрабатывается специализированное программное обеспечение. Разрабатываемое программное обеспечение должно обращаться к базе данных находящейся на сервере, для этого был выбран вид системы: клиент-сервер. Пользователю для входа в систему будет необходимо сначала авторизоваться, после чего он может выполнять определенные

операции, в зависимости от того под какой учетной записью он зашел. Различаются три основных состояния провизор, кладовщик и администратор.

Таким образом, функциональность разрабатываемой программы выглядит следующим образом:

- Формирование розничных цен;
- Учет поступления, отпуска в розницу;
- Учет льготного отпуска;
- Расчёт отчетов за произвольный период: остатков, прихода, расхода;
- Работа в локальной сети на нескольких рабочих станциях.

Для разработки программы было принято решение использовать следующее программное обеспечение: Microsoft Visual C# 2010 Express Edition. Для создания базы данных выбрана свободная система управления базами данных MySQL.

Для функционирования приложения предположительно необходимо следующее программное обеспечение: Операционная система Microsoft Windows XP/2003/Vista/7, Microsoft .NET Framework версии 4.0, Сервер БД MySQL.

Список литературы

1. Вендров А.М. Case-технологии. Современные методы и средства проектирования информационных систем: Учебник [текст] / Вендров А.М. – М.: Финансы и статистика, 2008. – 176 с.
2. Липаев В.В. Системное проектирование сложных программных средств для информационных систем. – 2-е изд. – М.: СИНТЕГ, 2002.
3. Электронная энциклопедия "Википедия" [Электронный ресурс] – Режим доступа: <http://ru.wikipedia.org/>

ПРОЕКТИРОВАНИЕ И РЕАЛИЗАЦИЯ СИСТЕМЫ УПРАВЛЕНИЯ СЕМАНТИЧЕСКИМ КОНТЕНТОМ НА ОСНОВЕ ОБРАБОТКИ ЕСТЕСТВЕННЫХ ЯЗЫКОВ

Кротова А.О. – студент

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

Рассмотрим вид информационного поиска в Интернет, основной целью которого является совершение определенного выбора на основе некоторых приблизительно сформулированных требований к предмету поиска. Примерами таких задач являются поиск подарка на определенный праздник, поиск товара в интернет-магазинах, поиск нового сотрудника. Эта разновидность поиска обладает следующими особенностями:

1. у пользователя имеются некоторое представление о предмете поиска, однако свои требования он не может точно описать;
2. на основе имеющегося опыта пользователь может иметь ассоциативные представления о предмете поиска, которые достаточно сложно сформулировать в форме поискового запроса, например, в Google ;
3. успешное решение приведенных задач требует от пользователя достаточно полного представления об информационной структуре предметной области;
4. каждый раз при обращении к этим задачам пользователь может обновлять свои знания о предметной области.

Получать информацию из текстов, размещенных в Интернет, для данной проблемы очень трудоемко. Система поиска, основанная на связях между понятиями предметной области и их визуализации, может существенно ускорить и облегчить выполнение описанных задач.

Сформулируем требования к модели знаний:

1. Модель должна быть достаточно универсальной относительно предметной области.
2. Модель должна обладать сравнительно простой структурой (что позволит составить алгоритмы автоматизированного извлечения знаний из текстов на естественном языке).
3. Должны существовать алгоритмы автоматического построения физического представления модели.
4. Модель должна быть ориентирована на построение по достаточно большому корпусу текстов (чтобы обеспечить достоверность содержащихся в ней знаний)
5. Модель должна иметь возможности репрезентативной визуализации.

Основой предлагаемой модели является лексикон на основе универсальных лингвистических словарей. Рассматриваемая структура лексикона представляет собой развитие модели, описанной в работах [1] и [2]. Источником лексикона могут служить существующие лингвистические словари. В данной работе рассматривается система, построенная на автоматической обработке толкового словаря Ожегова [3] и словаря синонимов [4]. Возможно использование любых других неадаптированных для автоматической обработки словарей.

Пусть A - алфавит лексического уровня применяемого естественного языка (ЕЯ). Тогда формально словом будем называть конечную упорядоченную последовательность непустых цепочек над этим алфавитом, разделенных пробелами. Пусть WA^+ - лексикон модели. Слова в лексиконе W могут являться как отдельными словами ЕЯ, так и устойчивыми словосочетаниями. Абстрагируемся от морфологических особенностей словоформ, и будем считать, что слова ЕЯ приведены к словарной форме.

Строго говоря, лексикон ЕЯ - бесконечное, динамически изменяющееся множество, но множество слов, содержащихся в словарях, является конечным. Поэтому множество W также будем считать конечным.

На множестве W зададим три типа нечетких отношений, которые мы можем извлечь из словарей:

1. отношение синонимии - $Syn: WW \rightarrow [0..1]$ определяет степень уверенности системы в том, что некоторое слово w_B W является синонимом к слову w_A W ;
2. отношение определения - $Def: WW \rightarrow [0..1]$ определяет степень уверенности системы в том, что некоторое слово w_A W является определением к слову w_B W ;
3. отношение ассоциации - $Acc: WW \rightarrow [0..1]$ определяет степень уверенности системы в том, что некоторое слово w_A W является ассоциацией к слову w_B W .

Введенные отношения позволяют установить степень близости слов на основе словарей. Назовем величину $Dict(w_A, w_B)$ **словарным уровнем близости слов** w_A и w_B . $Dict(w_A, w_B) = Max(Syn(w_A, w_B), Def(w_A, w_B), Acc(w_A, w_B))$.

Выбор именно этих видов отношений мотивируется в первую очередь тем, что приведенные типы связи естественным образом извлекаются из словарей.

Слова, связанные отношениями Acc , Def и Syn , образуют взвешенный граф. На основе трёх рассмотренных отношений возможно построение более строгих и семантически сложных отношений, таких как отношение обобщения, отношение часть-целое и др.

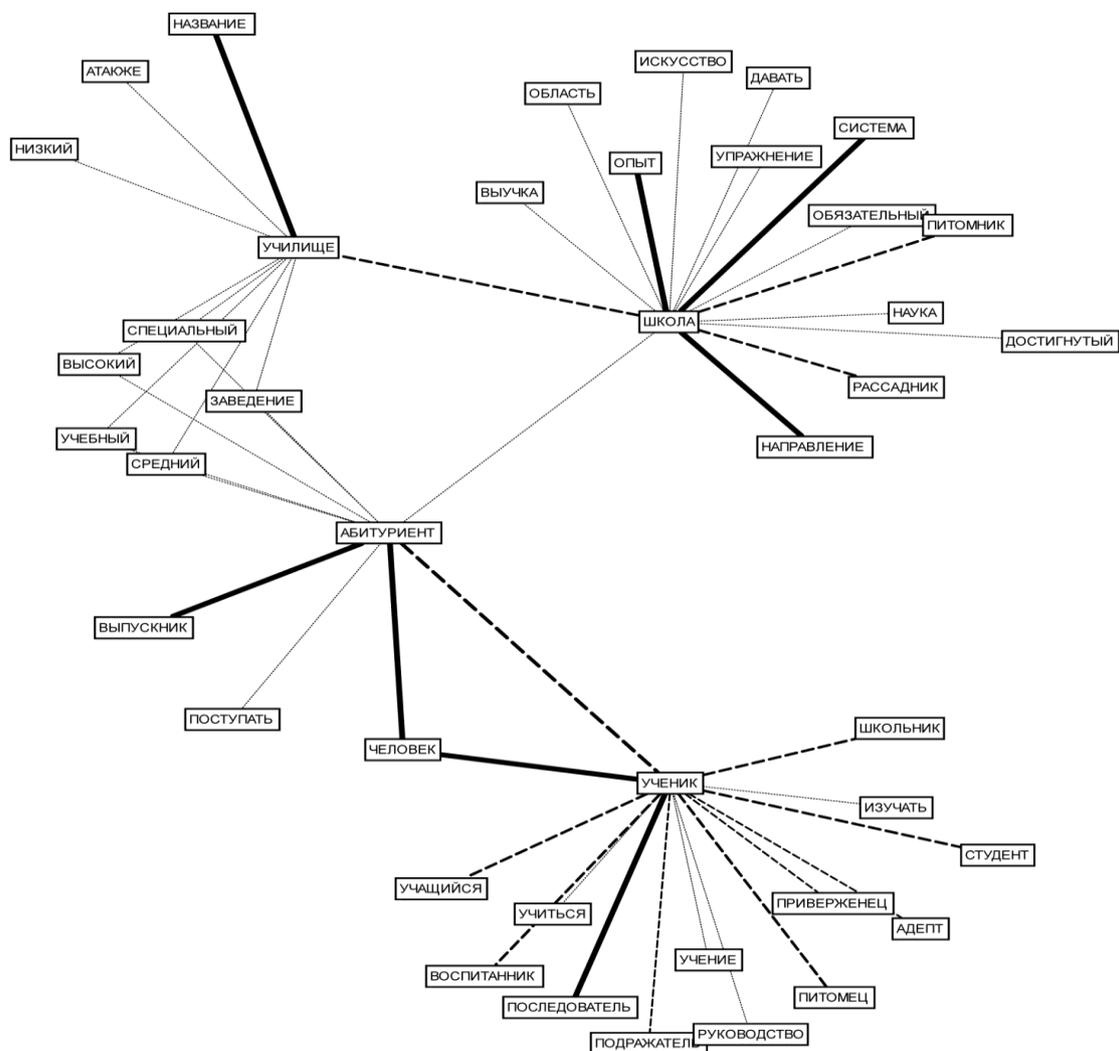


Рисунок 1 – Часть графа вокруг слова «Абитуриент»

На рисунке 1 представлена часть словарного графа. Сплошными линиями обозначены связи *Def*, пунктиром – связи *Syn*, точками – связи *Acc*. Толщина линии соответствует весу связи. Для построения графов использовалась программа AiSee[6].

Лингвистические словари позволяют построить некоторый достаточно строгий и релевантный окружающей действительности каркас лексических знаний для автоматизации смыслового поиска информации. Однако, модель несовершенна - существуют некоторые проблемы, связанные с существованием множества предметных областей. Появление таких проблем обусловлено причинами, перечисленными ниже:

- Недостаточный лексический запас словарей, отсутствие значительного количества неологизмов, заимствованных слов и жаргонизмов.
- Невозможность разрешения омонимии для конкретной предметной области.

Действительно, для предметной области «Аукцион произведений искусств» определение слова «КАРТИНА» через слово «ОБСТАНОВКА» будет значительно менее релевантным, чем через слово «ИЗОБРАЖЕНИЕ».

Источником дополнительной информации для уточнения лексикона в пределах конкретной предметной области могут быть специализированные словари и тексты на естественном языке.

Список литературы

1. Krayvanova V., Kryuchkova E. The mathematical model of the semantic analysis of phrases based on the trivial logic // In Proceedings of the 13-th International Conference "Speech and Computer" SPECOM'2009, pp. 543-546.
2. Крайванова В.А., Крючкова Е.Н. Проблема извлечения знаний в условиях динамически изменяющейся внешней среды на примере модели логического анализа текстов на естественном языке // Вычислительные технологии. 2010. Т. 15. № 3. С. 99-108.
3. Ожегов С.И., Шведова Н.Ю. Толковый словарь русского языка [Электронный ресурс]. Издательство "Азъ", 1992 – Режим доступа: <http://lib.ru/DIC/OZHEGOW/>
4. Абрамов Н. Словарь русских синонимов и сходных по смыслу выражений [Электронный ресурс]. Издательство Русские словари, 2007 г – режим доступа: <http://dict.buktopuha.net/data/abr1w.zip>
5. Сайт рабочей группы «Автоматическая обработка текстов» [Электронный ресурс] – Режим доступа: <http://aot.ru>
6. Программа визуализации графов AiSee [Электронный ресурс] – Режим доступа: http://www.absint.com/aisee/index_ru.htm

АВТОМАТИЗАЦИЯ ПРОЦЕССА ПРИЕМА ЗАЯВОК НА ПРОИЗВОДСТВО В ТОРГОВО-ПРОИЗВОДСТВЕННОЙ КОМПАНИИ «ЗАВОД ОКОН О»

Удалова А.В. – студент, Ананьев П.И. – доцент
Алтайский государственный технический университет (г. Барнаул)

В настоящее время существует множество производственных предприятий, отличающихся своей структурой, направленностью и типом производства. Однако основной функцией любого предприятия является производство и сбыт собственной продукции.

Рассматриваемое предприятие занимается следующими видами деятельности: производство, реализация и установка пластиковых окон.

Обычно, для постановки производственного процесса необходимо рассмотреть следующие этапы:

- Снабжение;
- Производство;
- Реализация.

Для любого производственного предприятия важно выполнить свои задачи в максимально сжатые сроки, качественно, с минимальными затратами и удовлетворять потребности клиентов. Для этой цели все больше и больше используют автоматизацию всех производственных процессов.

Автоматизация производства – процесс трудоемкий, долгий и дорогостоящий, так как на производстве необходимо постоянное вмешательство соответствующего специалиста. Поэтому начальный этап автоматизации часто касается не самого производства, а смежных подсистем.

В рамках дипломного проекта, ведется разработка программного обеспечения для приема заказов на производство пластиковых конструкций с последующей генерацией отчетов по результатам производства.

Программное обеспечение представляет из себя веб-приложение, разработанное с помощью технологий SharePoint.

Технологии SharePoint включают в себя:

Windows SharePoint Services 3.0 - набор служб для совместной работы и основа для построения веб-приложений на базе Windows Server.

Microsoft Office SharePoint Server 2007 - независимое интегрированное приложение, предоставляющее сотрудникам возможность взаимодействовать с членами рабочих групп, выполнять поиск организационных ресурсов, управлять содержимым и рабочим процессом.

Microsoft Office SharePoint Designer 2007 – средство для разработчиков и веб-дизайнеров, которое позволяет быстро создавать решения, включающие реализацию логики рабочего процесса и разработку пользовательского интерфейса.

Разработанное веб-приложение обеспечивает возможность:

- авторизации диллера в системе размещения заказов;
- размещения нового заказа, прикрепляя файл, полученный путём экспорта из Optima Win;
- приобретения статуса заказа «Размещён»;
- принятия интерфейса, позволяющий сотруднику предприятия консолидировано видеть заказы всех дилеров за произвольный период. Данный интерфейс должен иметь максимальную функциональность в плане применения фильтров, сортировок, групповых операций над заказами;
- проставления менеджером статусов, дат готовности, либо отклонения заказов с указанием причины;
- выгрузки файлов заказов клиентов в папку на компьютере сотрудника компании.

С помощью импорта файлов заказов, полученных посредством разработанного веб-приложения, в программный продукт Optima Win, использующийся на предприятии, происходит заполнение базы данных. Для новых заказов, с помощью Optima Win, генерируется стандартный отчет «Задание в цех», который отправляется на производство.

Далее, по результатам производства формируются отчеты. Для построения отчетов используется программа для расчета окон Optima Win и генератор отчетов Fast Report. Модуль построителя отчетов интегрирован с Optima Win и использует общую базу данных.

Разработаны следующие отчеты:

1. Отчёт, который позволяет формировать полную сводную информацию по одному или нескольким контрагентам (заказчикам) в разрезах:
 - Заказ;
 - Дата изготовления;
 - Статус заказа.
2. Отчет, который по заданному интервалу дат, по фактической дате пункта диспетчеризации «изготовление», делает выборку необходимых заказов и в выбранных заказах производит калькуляцию стоимости изделия по пунктам с учетом количества данного изделия.
3. Отчет, который по выделенному пользователем заказу будет предоставлять следующую информацию: номер изделия, профиль изделия, количество данного изделия, тип изделия, размер изделия, цвет изделия и схема изделия.
4. Исправить уже существующий отчет «Коммерческое предложение». Для этого, необходимо добавить вывод дополнительной информации по монтажу и комплектующим, номер изделия, количество комплектующего, цена за штуку, стоимость.

Разрабатываемое ПО внедряется в работу торгово-производственной компании «Завод окон О».

ОПРЕДЕЛЕНИЕ ПАТТЕРНОВ ПОВЕДЕНИЯ ПОЛЬЗОВАТЕЛЕЙ НА САЙТЕ

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

Для современных Интернет-ресурсов важно разрабатывать максимально эргономичный дизайн. Исходя из психических и физических особенностей человека специалисты Usability, используя различные методики и статистики, решают поставленную задачу. Учитывать интересы большинства пользователей позволит понимание того, кто и как использует информационную систему. Для эффективного поиска ответа на этот вопрос предлагается создать автоматизированный инструмент определения шаблонов поведения пользователей на сайте.

Любой сайт представляет собой систему web-страниц, связанных гиперссылками, которая может быть представлена в виде графа $G = \langle V, E \rangle$. Здесь множество вершин V – это страницы сайта. Будем считать, что страница определяется своим URL. Из вершины $a \in V$ в вершину $b \in V$ существует дуга $e \in E$, если на странице a существует ссылка, перейдя по которой пользователь попадет на страницу b .

Зачастую сайты содержат много однотипных страниц, которые для выявления общих шаблонов поведения пользователей целесообразно объединить в одну мультивершину. Пусть во множестве всех вершин V графа G существует такое подмножество $V_m \subset V$, что все вершины из V_m – это страницы сайта, обладающие однотипной функциональностью и объединенных общим дизайном. В этом случае сконденсируем все элементы из V_m в одну вершину m . Тогда будем говорить, что m – мультивершина. Исходное множество V можно заменить следующим образом:

$$V_{new} = V - V_m + m$$

Примером мультивершин могут служить товары в интернет - магазине или новости на новостном портале. В большинстве случаев URL страниц, входящих в мультивершину, имеют сходную структуру.

Заметим, что в мультивершинах могут образовываться ребра-петли.

Подграф $G_1 = \langle V_1, E_1 \rangle$ графа G будем называть пользовательским подграфом или поведением пользователя, если любая вершина $a \in V_1$ соответствует странице, которая была просмотрена пользователем, а каждая дуга $e \in E_1$, соединяющая a и $b \in V_1$, означает, что пользователь осуществлял непосредственный переход со страницы a на страницу b . В этом случае будем говорить, что G_1 характеризует поведение одного пользователя в информационной системе. Каждой вершине из V_1 поставим в соответствие метки времени: t_{begin} и t_{end} – время начала и завершения просмотра соответственно, а также метку об отказе (в случае, если пользователь прекратил работу с сайтом на этой странице).

Пусть имеется множество B , которое включает k пользовательских подграфов графа G , то есть $B = G_1 \cup G_2 \cup \dots \cup G_k$. Тогда задача определения паттернов поведения на сайте сводится к задаче разбиения множества B на группы (кластеры). Внутри каждой группы должны оказаться пользовательские подграфы, расстояние между которыми минимальное, а объекты разных групп должны быть как можно более удалены друг от друга.

Расстоянием $\rho(G_n, G_m)$ между подграфами G_n и G_m будем считать сумму:

$$\rho(G_n, G_m) = \rho_v(G_n, G_m) + \rho_t(G_n, G_m)$$

где ρ_v – мера различия наборов вершин графов G_n и G_m , а ρ_t – мера различия времени просмотра одинаковых вершин этих же графов.

$$d = |V_n \cup V_m - V_n \cap V_m|$$

$$\rho_v(G_n, G_m) = 2\alpha \frac{d}{|V_n| + |V_m|}$$

$$\rho_t(G_n, G_m) = \tau \sum_{i=1}^d \frac{|t_{1i} - t_{2i}|}{d}$$

Здесь V_n и V_m – множества вершин графов G_n и G_m соответственно, $|V_n|$ и $|V_m|$ – их мощности, t_{1i} и t_{2i} – время просмотра вершины i пользователем 1 и 2, α и τ – управляющие коэффициенты (подбираются эмпирически).

Каждый кластер будет характеризовать шаблон поведения некоторого типа пользователей. Предлагаемый подход даст возможность наглядно отобразить кластеры и выявить проблемы у пользователей, которые относятся к различным типам поведения.

На основе описанной выше модели предлагается разработать программное обеспечение, состоящее из четырех логических модулей: модуль сбора и хранения статистики посещений; интеллектуальный модуль обработки первичных данных и выделения паттернов поведения (МВПП); модуль графической визуализации пользовательских подграфов; интеллектуальный модуль определения проблем в дизайне сайта и формирования конкретных рекомендаций и замечаний по модернизации навигации на сайте.

Логический модуль сбора статистики должен быть отделен от остальной части ПО и вынесен как отдельный физический модуль. В его ответственность будет входить сбор необходимой информации, а также возможность конфигурации и регулирования количества собранных записей с учетом настраиваемого фильтра. Это позволит наиболее правильно формировать группы пользователей, за поведением которых будет вестись наблюдение и, как следствие, точно выявить шаблоны поведения и сформировать адекватные рекомендации.

Информация, которая будет записываться модулем статистики, будет включать: идентификационные данные пользователя (ip-адрес); страница, с которой произошел переход; текущая страница; время просмотра; время окончания просмотра. Этот список может быть дополнен другими показателями.

Модули алгоритмического определения паттернов поведения, генерации советов по улучшению, а также модуль графической визуализации пользовательских подграфов следует объединить физически в одном приложении. В задачи модуля МВПП входит автоматическое построение графа сайта (с заданной глубиной просмотра), формирование пользовательских подграфов на основе информации, собранной модулем статистики, а также их кластеризация. Модуль генерации советов отвечает за выявление проблемных мест в структуре сайта, таких как: малопосещаемые страницы, слишком удаленные страницы, страницы, на которых происходят частые отказы, ошибки в навигации (пользователи часто возвращаются назад из одной вершины), тупиковые ветки сайта и другое. За наглядное представление полученных данных отвечает модуль графической визуализации графов.

Описанная структура разделения ПО на логические и физические модули позволит создать программный продукт, который поможет специалистам по Usability перейти на новый качественный уровень работы. Он позволит существенно снизить затраты времени на сбор необходимой информации о пользователях и повысить объективность и точность описания существующей ситуации на оптимизируемом интернет – ресурсе.

Список литературы

1. Usability [Электронный ресурс] – М. 2012. – Режим доступа: <http://en.wikipedia.org/wiki/Usability>
2. Обзор алгоритмов кластеризации [Электронный ресурс] – М. 2012. – Режим доступа: [http:// http://habrahabr.ru/post/101338/](http://habrahabr.ru/post/101338/)

РАЗРАБОТКА СИСТЕМЫ АВТОМАТИЗИРОВАННОГО УЧЕТА И ИНВЕНТАРИЗАЦИИ КОМПЬЮТЕРНОЙ ТЕХНИКИ

Поканов Ю.А. – студент, Сорокин А.В. – к.т.н., доцент
Алтайский государственный технический университет (г. Барнаул)

Компьютер как любое сложное устройство состоит из множества элементов, называемых комплектующими, среди которых: материнская плата, процессор, винчестер и т.д. Для обычного пользователя компьютера тип процессора или сетевого контроллера, используемых в системном блоке компьютере, как правило, не представляет интереса. Для системного администратора эти характеристики являются важными, поскольку, обеспечивают бесперебойную и надежную работу компьютерной сети в целом. Данная работа направлена для создания программного обеспечения, позволяющего вести учет комплектующих компьютеров конкретной организации и установленного на них лицензионного программного обеспечения.

Задача учета компьютерной техники в организации очень важна по разным причинам:

1. во-первых, учет необходим для нормальной работы бухгалтерии. Поскольку каждый компьютер должен учитываться, как единое целое, так как бухгалтеру нужно всегда быть готовым к задаче инвентаризации;
2. во-вторых, учет необходим для нормальной работы системного администратора. В этом случае хорошее ПО для учета должно: учитывать компьютер не только как единое целое, но и как сборное изделие, состоящее из конкретных комплектующих деталей; наглядно представлять расположение компьютера или комплектующего элемента, четко отражать за каким пользователем он закреплён. При этом становится возможным быстрый поиск интересующего нас устройства (комплектующего) или групп устройств (комплектующих) по определенному критерию. Всегда можно оперативно узнать, где находится конкретный компьютер или определенных комплектующий элемент.

Таким образом, учитывая выше изложенное, разрабатываемая система автоматизированного учета должна содержать следующие функции:

- а) ведение списка производителей компьютерного оборудования;
- б) ведение списка поставщиков компьютерного оборудования;
- в) ведение списка компьютеров предприятия с закреплением их по подразделениям и сотрудникам;
- г) учет и списание компьютеров в виде комплектующих;
- д) ведение списка документов покупки оборудования, просмотр и печать содержимого каждого документа;
- е) вывод подробной информации о составе комплектующих компьютера с подробным указанием реквизитов элементов (серийные и инвентарные номера, технические параметры), а также данных об их покупке (документ, поставщик, цены, гарантия);
- ж) ведение списка программного обеспечения, установленного на компьютерах;
- и) поиск, отбор и фильтрация оборудования по любому критерию, его принадлежности, установленному ПО и работающим пользователям, а также по совокупности этих критериев;
- к) составление отчетов о компьютерах предприятия.

Реализация данной системы осуществляется в виде программы, для разработки которой было принято использовать следующее программное обеспечение: Microsoft Visual C# 2010 Express Edition. Для создания базы данных выбрана свободно распространяемая система управления базами данных MySQL.

Список литературы

1. Вендров А.М. Case-технологии. Современные методы и средства проектирования информационных систем: Учебник [текст] / Вендров А.М. – М.: Финансы и статистика, 2008. – 176 с.
2. Норенков И. П. Основы автоматизированного проектирования: учеб. для вузов. – 4-е изд., перераб. и доп. – М.: Изд-во МГТУ им. Н. Э. Баумана, 2009. – 430 с.
3. Электронная энциклопедия "Википедия" [Электронный ресурс] – Режим доступа: <http://ru.wikipedia.org/>

РАЗРАБОТКА ПРОГРАММНОГО КОМПЛЕКСА ДЛЯ ПРЕДОСТАВЛЕНИЯ ИНФОРМАЦИИ МОБИЛЬНЫМ КЛИЕНТАМ

Трофимчук А.А. – студент, Вальков Д.В. – ассистент
Алтайский государственный технический университет (г. Барнаул)

В настоящее время остро стоит проблема оперативного информирования. От того насколько своевременно будет получена информация зависит скорость принятия решений и соответствующее дальнейшее развитие событий. Одним из путей решения данной проблемы, в условиях современного общества, является использование для предоставления информации мобильных устройств и предоставляемых ими сервисов, получивших на сегодняшний день широкое распространение. В связи с этим актуальной является задача разработки программного комплекса для предоставления информации мобильным клиентам.

На сегодняшний день на рынке программного обеспечения в области предоставления информации мобильным клиентам существует множество проектов, предназначенных для передачи контента мобильным клиентам по средствам различных технологий. К сожалению, существующие решения не лишены недостатков, не способны комплексно решать поставленные задачи, и являются дорогими. Среди основных недостатков, представленных на рынке решений можно выделить: слабую масштабируемость, отсутствие централизованного управления и узкую специализацию.

Проанализировав потребность рынка и существующие технологии позволяющие решать поставленные задачи, была сформулирована основная цель данной работы. Таким образом, целью данной работы является разработка программного комплекса для предоставления информации мобильным клиентам по средствам технологии SMS и BLUETOOTH.

В ходе анализа был сформулирован ряд основных требований к реализуемому программному комплексу:

- Возможность масштабирования системы;
- Наличие централизованного механизма управления комплексом;
- Наличие механизма для визуализации статистики предоставления информации;
- Возможность подписки на информационные рассылки;
- Доставка текстовых сообщений по технологии SMS;
- Предоставления доступа к контенту по технологии BLUETOOTH всем устройствам, находящимся в зоне покрытия

На этапе проектирование комплекса была сформулирована концепция аппаратной части комплекса (рисунок 1). Состоящая из отказа устойчивого серверного решения в связке с

GSM-шлюзом, выполняющего роль централизованного механизма управления комплексом а так же функционал предоставления информации по технологии SMS, и компактных решений на базе одноплатных компьютеров с BLUETOOTH выполняющих роль точек доступа, и берущих на себя функционал предоставления контента по технологии BLUETOOTH.

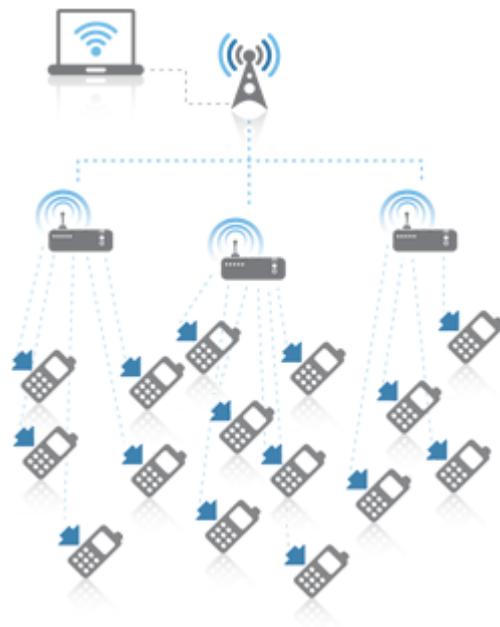


Рисунок 1 – Концепция аппаратной части комплекса

После чего была спроектирована архитектура комплекса, позволяющая оперативно выполнять поставленные задачи и гибко масштабировать систему. Проведя анализ, в качестве базовой операционной системы был выбран linux дистрибутив Debian.

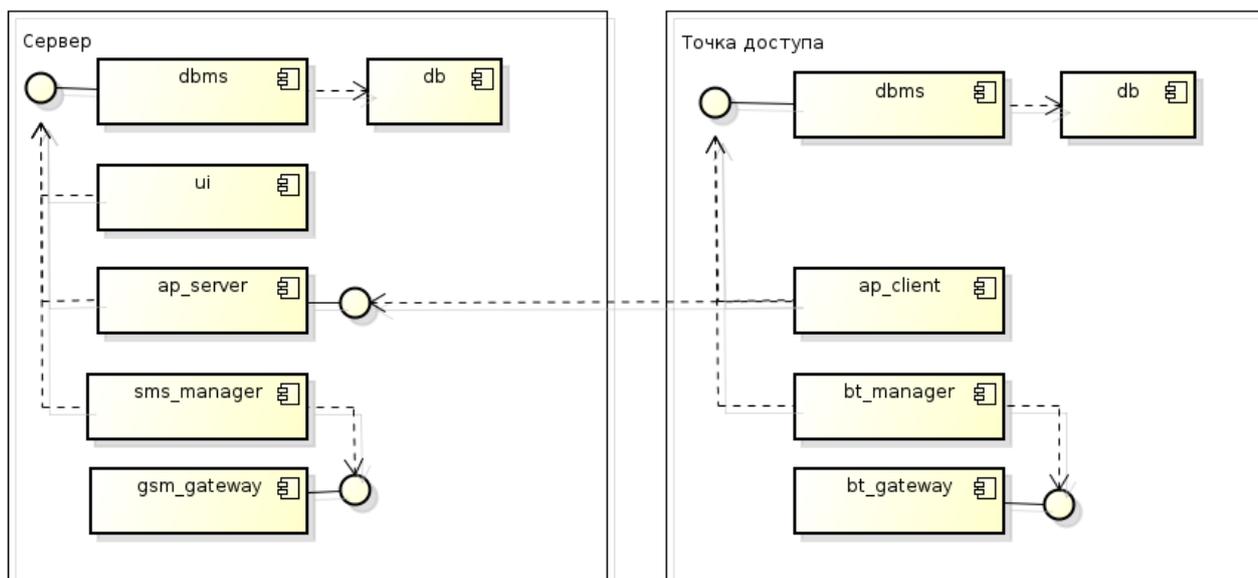


Рисунок 2 – Диаграмма компонентов комплекса

В комплексе можно выделить пять основных групп компонентов (рисунок 2).

Первая группа компонентов это компоненты осуществляющие работу с базами данных(dbms и bd). На сервере в этой роли выступает СУБД Oracle, на точках доступа SQLite. Через заданный администратором промежуток времени данные из базы точек доступа выгружаются в базу находящийся на сервере.

Второй группой компонентов комплекса являются компоненты реализующие пользовательский интерфейс взаимодействия с комплексом (ui). Данная часть комплекса реализована на Oracle Application Express , что значительно упрощает интеграцию с выбранной СУБД, а так же позволило реализовать более тонко настраиваемые политики доступа к данным. Так как данная технология предназначена для разработки веб-приложений, также её использование позволяет выполнить требования наличия в комплексе централизованных механизмов управления и визуального отображения статистики.

В третью группу компонентов можно объединить модули отвечающие за предоставление информации по технологии SMS (sms_manager и sms_gateway). Основным является модуль осуществляющий отправку и прием и обработку SMS сообщений взаимодействуя с GSM-шлюзом через последовательный порт. Данный компонент реализован на C++ с использованием библиотеки QT и QextSerialPort.

Четвертая группа компонентов реализует клиент северную архитектуру(ap_server и ap_client), и позволяет осуществлять централизованное управление и обмен данными с точками доступа. Для реализации данных компонентов так же реализован на C++ и использует библиотеки QT.

В пятую группу компонентов объединены модули отвечающие за предоставление информации по технологии BLUETOOTH (bt_manager и bt_gateway). Основным в данной группе является модуль осуществляющий предоставление контента по технологии BLUETOOTH и взаимодействующий с BLUETOOTH-шлюзом. Данный компонент так же реализован на C++ и использует библиотеки QT и BlueZ.

В дальнейшем планируется доработка комплекса для повышения его отказоустойчивости , возможности работы в режиме 24 на 7, а также увеличения его производительности. Так же планируется комплексное тестирование работа способности комплекса и его производственные испытания.

НЕКОТОРЫЕ ВОПРОСЫ РАЗРАБОТКИ СИСТЕМЫ ВЫЯВЛЕНИЯ ПЛАГИАТА В ИСХОДНЫХ КОДАХ НА ЯЗЫКЕ ПРОГРАММИРОВАНИЯ ВЫСОКОГО УРОВНЯ C++

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

1. Введение

Плагиат – это умышленное присвоение авторства на чужое произведение науки, литературы или искусства в целом или в частности. Проблема плагиата в образовании достойна отдельного внимания и заключается не только в незаконном присвоении авторских произведений, но и в подрыве самой сути образовательного процесса. И поэтому рост объема плагиата в письменных работах студентов, в том числе, в программных кодах, приводит к необходимости разработки и внедрения в вузах специализированного инструментария – систем поиска плагиата.

Целью данной работы является разработка системы выявления плагиата в исходных кодах именно на языке программирования высокого уровня C++, поскольку данный язык на данный момент является основным языком программирования в технических вузах.

2. Обзор детекторов плагиата

В настоящее время для обнаружения фактов списывания программ студентами друг у друга разработано некоторое количество программ. Такие приложения позволяют сравнить две работы между собой, или работу с внутренним архивом. Наиболее часто программы подвергаются плагиату на уровне исходного текста, реже – на уровне исполняемого кода. В связи с этим, подавляющее большинство детекторов осуществляет поиск плагиата в исходном и промежуточном представлениях программ. Ниже приведены краткие описания характеристик некоторых детекторов плагиата:

- SIM (Software Similarity Tester) – детектор с открытым кодом, разработанный Диком Грюном. Работа детектора SIM основывается на алгоритме поиска по матрице совпадений подстрок; используется представление программного кода в виде токенов. SIM поддерживает такие языки программирования как C, Java, Pascal, Modula-2, Lisp, Miranda, а также естественные языки. Недостатком данного детектора является невозможность выявления плагиата, полученного путем перемещения блоков кода.
- JPlag – детектор, доступный в режиме online. Разработчиком является Guido Malpohl. В основе работы детектора лежит алгоритм жадного строкового замощения, что требует предварительной токенизации исходного программного кода. Jplag поддерживает следующие языки программирования: C, C++, Scheme, Java.
- MOSS (Measure of Software Similarity) – детектор, доступный в сети Интернет в режиме online. Для поиска плагиата используется алгоритм просеивания для построения идентификационных меток. Поддерживает большое количество языков, в частности: C, C++, Lisp, Haskell, Pascal, Python, Assembler и др.
- SID (Software Integrity System) – доступный в режиме online детектор, основанный на использовании Колмогоровской сложности, но требующий предварительной токенизации текста программы. Поддерживает языки Java и C++.

3. Разбор кода

Для анализа кода необходимо средство, позволяющее производить разбор и выделение конструкций языка. Язык C++, как и все языки программирования, является языком, тексты на котором структурированы и составляются согласно набору правил построения синтаксических конструкций. Такой набор правил называется грамматикой языка. Полная грамматика C++ содержит порядка 200 правил, из них в действительности используется около половины. Тем не менее, такое количество правил велико. К тому же грамматика была спроектирована так, что в некоторых правилах присутствуют неоднозначности, что, как следствие, серьёзно осложняет процесс реализации синтаксического анализатора языка. Поэтому, было решено поступить следующим образом:

- Составить грамматику нисходящего разбора (LL) подмножества языка C++, оперирующую с идентификаторами (переменные, константы, классы, метки и др.).
- Записать получившуюся грамматику в системе ANTLR, средстве анализа грамматики и генерации по ней исходного модуля синтаксического анализатора (парсера) и лексического сканера (лексера). Система является freeware.
- Программа, содержащая модули лексера и парсера, должна составлять по анализируемому тексту статистику, которая представляет собой список всех использованных идентификаторов, сгруппированных по типам.

Решение не давало ожидаемых результатов. Программа не всегда корректно находила и группировала идентификаторы из-за оставшихся в грамматике неоднозначностей.

Для того чтобы реализовать максимальную гибкость и настраиваемость системы, а также упростить процесс сбора данных об идентификаторах и структуре кода, на данный момент было решено спроектировать парсер языка на основе нисходящего алгоритма

разбора (LL) с использованием принципа прозрачного синтаксиса и синтаксически-управляемого перевода (СУП) с дельта-функциями, указывающими на структуру программы и данных (Начала цикла, Докончания цикла, Дописания и т.д.). Алгоритм LL-анализа позволяет размещать эти функции в любом месте правил грамматики.

4. Методы выявления плагиата

4.1 Анализ стиля программирования

Данный метод основан на термине «родимые пятна» («birthmarks»), обозначающий прошлые авторские наработки или особенности, которые присущи стадии разработки программы и которые в дальнейшем становятся неотъемлемой частью этой программы. Признано, что родимые пятна существуют объективно. Это обстоятельство позволяет использовать анализ стиля программирования для доказательства факта заимствования.

4.2 Специальные интеллектуальные методики

К таковым относятся: токенизация, параметризованный поиск, сравнение деревьев разбора. Реализация таких методик трудоёмка и осуществляется посредством создания и обучения нейронных сетей.

4.3 Анализ характеристик программы

Данный метод подразумевает собой сравнение числовых рядов-характеристик исходных кодов между собой. К такой характеристике, например, можно отнести распределение переменных по операторам. Если представить данную характеристику в виде графика, то получится следующее. По оси абсцисс откладываются токены (лексемы) программы, а по оси ординат типы операторов. График будет представлять ломанную, т.к. не на каждую лексему будет приходиться оператор.

Сравнение графиков двух подозрительно похожих друг на друга программных текста производится следующим образом. Сначала графики совмещаются так, чтобы они занимали одну область определения. Этого возможно достичь двумя вариантами: либо область определения одного графика расширяется (сжимается) до размеров другого, либо области определения обоих графиков расширяются на величину наименьшее общее кратное размеров областей. При изменении размеров областей необходимо применять интерполяцию (линейную, сплайн или др.) для вычисления значений в промежуточных точках.

Следующий этап это непосредственно само сравнение. При этом результат сравнения не является однозначным утверждением, что один график абсолютно равен другому. Здесь определяется лишь степень похожести. Есть несколько вариантов сравнения:

1. Использовать библиотеку компьютерного зрения OpenCV. Библиотека содержит функцию сравнения двух гистограмм *double cvCompareHist(...)* с применением одной из четырех метрик сравнения (корреляция, хи-квадрат, пересечение, расстояние Бхаттачария). На вход функции необходимо подать кривые в виде гистограмм. На выход функция возвращает значение от 0 до 1, и чем число ближе к 0, тем больше гистограммы совпадают.
2. Вычисление интегральной характеристики – среднего квадратичного отклонения или разности знаков производных по всей области определения.
3. Использовать нейронные сети, хорошо обученных на сравнение изображений.

4.3 Анализ кода посредством метрик

Метрика программного обеспечения (англ. software metric) — это мера, позволяющая получить численное значение некоторого свойства ПО или его спецификаций. Поскольку количественные методы хорошо зарекомендовали себя в других областях, многие теоретики и практики информатики пытались перенести данный подход и в разработку ПО.

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

Рассмотрим группы метрики, наиболее актуальные для программных проектов:

- **Метрики размера**

К этому классу относят, например, различные показатели количества строк кода, степень его комментированности, а также показатели, отражающие количественную оценку отдельных составляющих проекта. В большинстве случаев главное предназначение этой группы метрик, независимо от способа их вычисления, состоит в том, чтобы оценить объем работ по проекту, и, соответственно, быть основой для таких показателей, как стоимость и длительность его реализации.

- **Метрики сложности**

Основная цель метрик сложности – выявить наиболее критичные участки программного проекта, которые являются потенциальными источниками ошибок и повышенных рисков. Одна из самых распространенных таких метрик – цикломатическая сложность. Данная метрика предназначена для оценивания сложности потока управления программы и вычисляется на основе ориентированного графа, где вычислительные операторы или выражения представляются в виде узлов, а передача управления между узлами – в виде дуг.

- **Объектно-ориентированные метрики**

Поскольку основные элементы конструирования объектно-ориентированных (ОО) программ классы, реализующие функциональность посредством методов, то именно классы и методы являются категориями, которыми преимущественно оперируют ОО метрики. Помимо таких очевидных метрик, как общее число классов, методов, атрибутов, средних показателей числа методов и атрибутов на класс и пр., существуют комплексные метрики, с помощью которых можно судить не только об объеме исходного кода ОО проекта, но и о его сложности, качестве, соответствии основным принципам ОО парадигмы.

Наш проект анализирует программные тексты на основе грамматик, построенных по принципу прозрачного синтаксиса таким образом, чтобы каждая грамматика соответствовала какой-то метрике. В связи с большим количеством метрик, было решено спроектировать масштабируемую архитектуру системы на основе поведенческого шаблона проектирования Посетитель (Visitor), который позволяет выполнять над каждым объектом некоторой произвольной структуры одну или более операций. Данный шаблон позволяет определить и задать на выполнение новую операцию, не изменяя ни классы этих объектов, ни их структуры, что подходит для задач вычисления различных метрик над одним кодом.

Список литературы

1. Макаров В.В. Идентификация дублирования и плагиата в исходном тексте прикладных программ. [Электронный ресурс] / В.В. Макаров. – Режим доступа: <http://lab18.ipu.rssi.ru/projects/conf2006/1/V.V.Макаров.htm>
2. Иванчегло С. Методы выявления плагиата в программировании. [Электронный ресурс]/Сергей Иванчегло. – Режим доступа: <http://www.kv.by/index2000491105.htm>
3. Колдовский В. Разработка ПО: метрики программных проектов [Электронный ресурс] / Режим доступа: http://itc.ua/articles/razrabotka_po_metriki_programmnyh_proektov_27774
4. Википедия – свободная энциклопедия. [Электронный ресурс] / Режим доступа: <http://ru.wikipedia.org>

5. Библиотека компьютерного зрения с открытым исходным кодом OpenCV (Open Source Computer Vision Library) [Электронный ресурс] / Режим доступа: <http://opencv.ru/>

ИСПОЛЬЗОВАНИЕ АССОЦИАТИВНЫХ СВЯЗЕЙ В ЗАДАЧАХ ПОИСКА

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

В настоящее время идет непрерывный и быстрый рост объемов информации. Значительную часть этой информации составляют текстовые данные, причем многие тексты хранятся не структурировано. В этом огромном массиве информации очень сложно найти то, что необходимо.

Дело в том, что технически существует возможность доступа к данным, но практически доступна лишь малая часть. Это обусловлено несовершенством средств доступа. Современные средства поиска текстов не удовлетворяют в полной мере требованиям пользователей. Необходимо их развитие и усовершенствование в направлении повышения эффективности поиска информации и упрощения взаимодействия с пользователем.

Одним из возможных способов решения проблемы доступа к необходимой информации является использование ассоциативных связей в алгоритме поиска.

Составной частью предлагаемой системы является система синтаксического анализа Dialing. Единицей синтаксического анализа является отдельное предложение текста, для которого строится дерево зависимостей между составляющими его единицами, обычно словами. По результатам анализа может быть получена следующая информация:

- Все слова с указанием части речи и синтаксической роли в предложении.
- Все слова, синтаксически подчиненные данному слову, с указанием типа синтактико-семантической связи.

Дерево синтаксических зависимостей – есть дерево, множество узлов которого служит множеством вхождения слов в предложение. Слова связаны друг с другом отношениями подчинения. Сказуемое обычно помещается в корень дерева, а все остальные слова зависят от него как непосредственно, так и опосредованно. Ребра дерева обычно помечаются символами синтаксических отношений и характеризуются определенным весовым коэффициентом.

В ходе работы были рассмотрены связи определения, синонимии и ассоциации с различными весовыми коэффициентами. Для простоты все связи можно привести к ассоциативным связям.

Был проведен анализ баз данных, полученных в результате обработки словарей естественного языка. Базы содержат связи и весовые коэффициенты.

Например, в базе имеются следующие связи со словом абстракционизм:

абстракционизм 1 = направление 1 $i=1.00$, последователь 1 $a=0.50$, изображать 4 $a=0.50$, реальный 2 $a=0.50$, мир 1 $a=0.50$, сочетание 1 $a=0.50$, отвлеченный 2 $a=0.50$, форма 1 $a=0.50$, цветовой 2 $a=0.50$, пятно 1 $a=0.50$;



Таким образом, на основе исходного запроса и баз данных можно построить граф запроса, основанный на структуре запроса и дополненный узлами, которые связаны ассоциативной связью со словами - элементами запроса. Дальнейшая работа по кластеризации текстов и выявлению текстов, наиболее точно соответствующих запросу, основана на обработке полученного графа.

Необходимо определить меру близости текста большого объема и поискового запроса. Если при анализе текста выявляются слова, соответствующие узлам исходного графа, мы прокрашиваем ребра графа, которые связаны с найденными словами, затем анализируем веса всех покрашенных ребер и вычисляем коэффициент соответствия текста запросу. Если рассчитанный коэффициент превысит некоторое пороговое значение, то текст будет включен в результат поиска.

В ходе работы были рассмотрены ассоциативные правила, как один из инструментов поиска информации, позволяющие эффективно обрабатывать большие объемы данных и находить необходимую информацию.

АЛГОРИТМ РАССТАНОВКИ ВЕРШИН ГРАФА НА ПЛОСКОСТИ, ОСНОВАННЫЙ НА ПРИМЕНЕНИИ ФИЗИЧЕСКОГО ВЗАИМОДЕЙСТВИЯ

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

Графы – распространенный способ представления информации, например, при работе с сетями или рассмотрении зависимостей между какими-либо объектами. В настоящее время при решении многих прикладных задач возникает проблема визуализации графов. В данной работе рассматривается так называемый «силовой» алгоритм (Force-based) расстановки вершин. Приводятся полученные эвристические улучшения алгоритма. Например, улучшение результата в зависимости от начального положения вершин перед запуском алгоритма. Применяемый алгоритм является достаточно простым, но при этом дает хорошие результаты. Также к плюсам можно отнести, что на выход он может принимать любые графы, независимо от их вида.

Алгоритм использует два физических закона: закон Гука и закон Кулона. Ребра графа представляют собой пружины, которые подчиняются силе упругости. Они стягивают вершины между собой. Вершины представляют собой точечные заряды, имеющие один знак. Поэтому они отталкиваются. Каждую итерацию алгоритма мы для каждой вершины просчитываем силы, влияющие на нее, и сдвигаем ее в плоскости в соответствии с вектором этой силы. Алгоритм продолжает свою работу до тех пор, пока система не примет состояние равновесия. Сила, действующая на вершину v вычисляется по формуле:

$$F(v) = \sum_{(u,v) \in E} f(u,v) + \sum_{(u,v) \in V \times V} g(u,v),$$

где $f(u,v) = k_1(d(u,v) - L_{uv}) \frac{x_v - x_u}{d(u,v)}$, $g(u,v) = \frac{k_2}{(d(u,v))^2} \frac{x_v - x_u}{d(u,v)}$, $d(u,v)$ – расстояние между вершинами u и v .

Приведена формула проекции силы на ось OX , для проекции на ось OY формула аналогична. Коэффициенты k_1 и k_2 подбираются эмпирически, в зависимости от необходимых характеристик получаемого изображения. Параметр L_{uv} – это желаемая длина ребра (расстояние) между вершинами u, v .

Качество получаемых результатов существенно зависит от начальной расстановки вершин. В литературе [1] рекомендуется изначально задать координаты вершин случайно. Но как показали тесты, зачастую случайный выбор координат не дает хорошего итогового изображения из-за того, что некоторые вершины занимают изначально неудачные позиции. Поэтому были проведены различные тесты, которые помогли выявить некоторые полезные эвристики изначальной расстановки вершин. Наиболее удачной оказалась изначальная расстановка вершин в порядке их обхода алгоритмом поиска в ширину. При выборе очередной вершины все ее соседи равномерно расставляются в правой полуокружности с радиусом L и центром в этой вершине. Такая расстановка изначально уже дает довольно неплохой результат, а после запуска алгоритма, мы получаем на выходе равномерно распределенный на плоскости граф. Пример можно увидеть на рисунке 1.

Асимптотика алгоритма: $O(k * N^2)$ (без учета начальной расстановки), где N – количество вершин. k – количество итераций, которое понадобится для приведения системы в состояние равновесия. Параметр k – можно задать и явно.

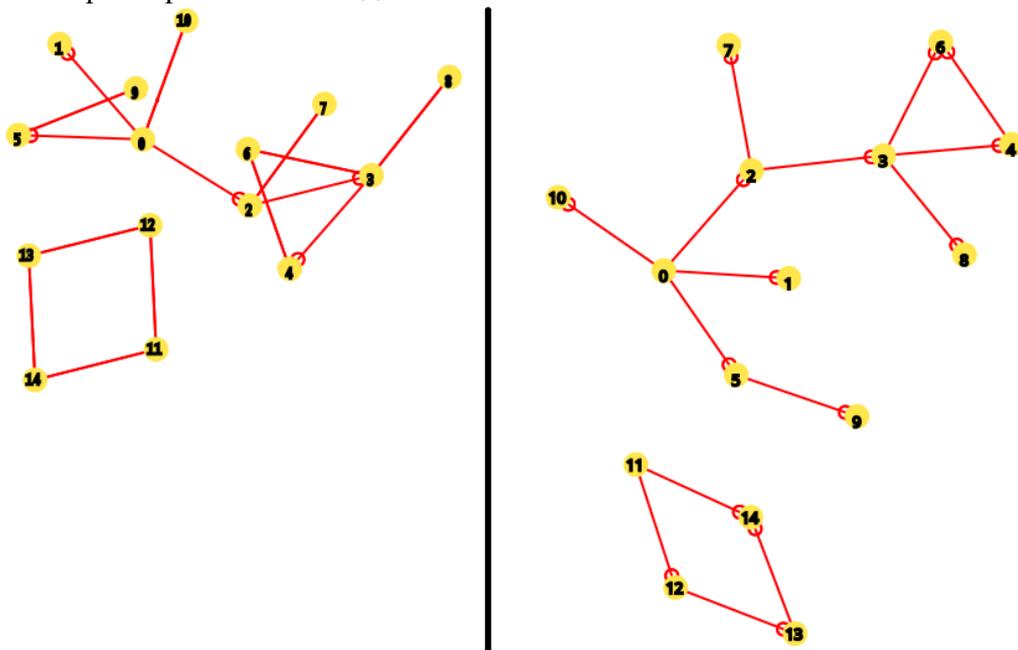


Рисунок 1 – Результаты со случайной (слева) и «качественной» (справа) начальной расстановкой

Для отображения результатов работы алгоритма был написан визуализатор на языке программирования C++. Он использует кроссплатформенную библиотеку Gtkmm 3.0 для отрисовки графа.

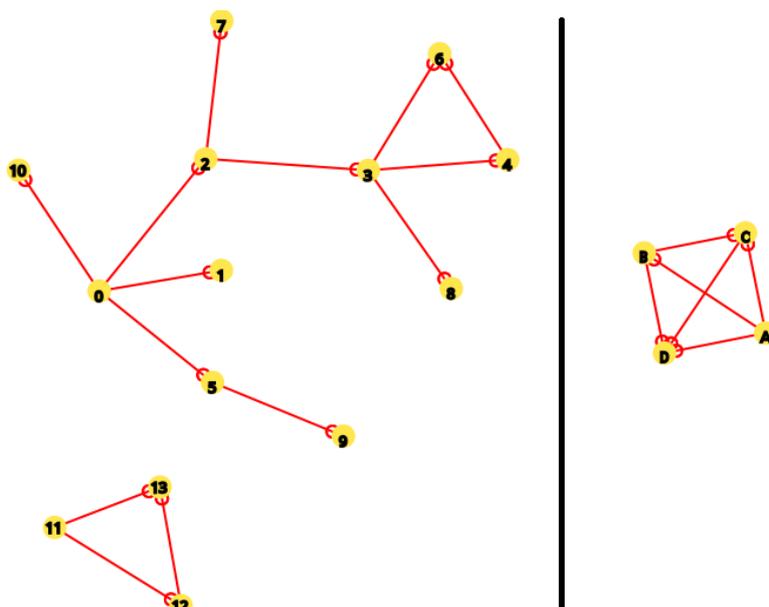


Рисунок 2 – Примеры работы алгоритма

В планах на данный момент значится работа над решением вопроса применения алгоритма на несвязных графах. При наличии нескольких компонент связности встает проблема выбора позиций расположения для каждой компоненты. На данный момент этот вопрос решен следующим образом: вводятся дополнительные «фиктивные» ребра, которые соединяют каждую компоненту с какой-либо другой. Но, к сожалению, такое прямолинейное решение дает не очень хороший результат в плане использования площади.

В будущем алгоритм будет оформлен в виде библиотеки вместе с некоторыми другими алгоритмами обработки графов, которая будет выложена в открытый доступ.

Список литературы

1. Battista G. Algorithms for the visualization of graphs – New Jersey: Prentice-Hall Inc., 1999. - 398 с.

УЧЕБНЫЕ ПЛАНЫ ФГОС ВПО ТРЕТЬЕГО ПОКОЛЕНИЯ В АЛТГТУ ИМ. И.И. ПОЛЗУНОВА

Фаст Ю.С. – студент, Ананьев П.И. – доцент
Алтайский государственный технический университет (г. Барнаул)

Федеральные государственные образовательные стандарты (ФГОС) высшего профессионального образования (ВПО) представляют собой совокупность требований, обязательных при реализации основных образовательных программ высшего профессионального образования образовательными учреждениями, имеющими государственную аккредитацию.

Федеральные государственные образовательные стандарты должны обеспечивать:

- единство образовательного пространства Российской Федерации;
- преемственность основных образовательных программ высшего профессионального образования.

В данное время Алтайский государственный технический университет им. И.И. Ползунова переходит на учебные планы нового поколения. Таким образом, в 2010-ые годы планируется переход всех специальностей вуза на систему бакалавриата/магистратуры. В связи с этим, разрабатываются новые учебные планы согласно ФГОС ВПО третьего поколения. Согласно временному положению о разработке рабочих учебных планов в соответствии с федеральными государственными образовательными стандартами высшего профессионального образования в ГОУ ВПО «Алтайский государственный технический университет им. И.И. Ползунова» от 03.03.2011 были установлены требования к составлению рабочих учебных планов в АлтГТУ. При разработке новых учебных планов направлений подготовки (специальностей) следует использовать формат РУП ФГУ «ИМЦА» (г. Шахты) для предварительной формальной проверки (в частности, для контроля числа зачетных единиц (ЗЕТ) в году). Для деканатов обязательным является требование о предоставлении учебных планов в учебный отдел в двух форматах: ФГУ «ИМЦА» и АлтГТУ. Учебный отдел, в свою очередь, обязан формировать рабочие учебные планы (РУП) в формате АлтГТУ.

Существует пакет «GosInsp», разработанный в городе Шахты и предназначенный для набора рабочих учебных планов направлений и специальностей ВПО, с целью представления их в Информационно-методический центр аттестации образовательных организаций (ИМЦА) для проверки на соответствие государственным стандартам высшего профессионального образования. Пакет «GosInsp» свободно распространяется среди ВУЗов, проходящих процедуру аттестации или самоаттестации в рамках ИМЦА. Согласно временному положению о разработке РУП возникает необходимость преобразования учебных планов, сформированных с помощью пакета GosInsp в формат АлтГТУ.

В рамках дипломного проекта, разрабатывается комплекс программного обеспечения (ПО) для учебного отдела и деканатов АлтГТУ. ПО обеспечивает возможность экспорта данных, сформированных с помощью пакета GosInsp, в существующую в АлтГТУ базу данных (БД), которая организована с помощью СУБД Oracle. Для хранения учебных планов нового поколения, БД была дополнена необходимыми таблицами. Программное обеспечение, используемое сотрудниками АлтГТУ для обработки различных внутренних данных и документов, реализовано преимущественно на языке программирования Java. IDE NetBeans 7.0 для Java дает возможность разработки ПО с интуитивно-понятным пользовательским интерфейсом, кроме того язык программирования Java имеет широкие возможности интеграции с СУБД Oracle. Также, преимуществом Java над другими языками программирования является то, что приложения Java обычно компилируются в специальный байт-код, поэтому они могут работать на любой виртуальной Java-машине (JVM) независимо от компьютерной архитектуры. Исходя из вышесказанного, для разработки дипломного проекта были выбраны именно эти средства. Разрабатываемое ПО представляет собой приложение с оконным интерфейсом.

Для экспорта данных из файла формата XML, сформированного пакетом GosInsp в БД, были написаны процедуры на языке PL/SQL. Структура XML файла фиксирована и открыто представлена на сайте разработчиков в файле VSD. Согласно представленной структуре, была написана процедура, выполняющая непосредственно разбор XML с сохранением содержимого в БД. Часть данных, относящихся к РУП содержится в этом же файле, часть уже имеется в БД, но некоторые данные требуют заполнения учебным отделом. Таблицы, для хранения этой информации также были добавлены в структуру БД. Для заполнения этих таблиц известной информацией, также была написана процедура на PL/SQL. Недостающая информация может быть заполнена сотрудником учебного отдела через прикладное ПО.

Разрабатываемое прикладное ПО должно отвечать всем требованиям учебного отдела по формированию необходимых документов. Деканаты должны иметь возможность

формирования учебных поручений на кафедры. В то же время, пользователи ПО – сотрудники деканатов, не должны иметь возможности просмотра и редактирования данных, относящихся к другим факультетам, однако сотрудники учебного отдела имеют право просматривать и корректировать информацию по всем учебным планам вуза.

Учебные поручения формируются каждым деканатом для одной кафедры на конкретный учебный год. Существует возможность корректировки либо заполнения недостающих данных. На данный момент, сотрудники деканатов формируют учебные поручения на кафедры вручную, что является трудоемким процессом, отнимает большое количество времени и не исключает человеческих ошибок.

Учебному отделу предоставляется возможность формирования множества документов.

Для конкретного учебного плана формируются:

- Рабочие семестровые учебные планы для студентов;
- Линейный график учебного процесса;
- План учебного процесса;
- Расчетная нагрузка преподавателей на группу или направление.

Существует возможность формирования сводных ведомостей для всех учебных планов, составленных на один учебный год:

- Структура учебной нагрузки по вузу;
- Данные к составлению расписаний.

Для всех формируемых документов существует возможность печати или экспорта в MS Excel.

В текущее время, сотрудники учебного отдела получают бумажную копию заполненного учебного плана, либо файл MS Excel, сформированный пакетом GosInsp. Другие необходимые документы формируются сотрудниками учебного отдела вручную, частично копированием данных из одного файла в другой. Необходимость формировать документы возникает часто в ходе учебного процесса, а непосредственно формирование отнимает время.

Разрабатываемое ПО внедряется в работу учебного отдела и в работу деканатов. По мере возможности учитываются пожелания сотрудников и вносятся изменения в функционал и интерфейс программы.

После внедрения ПО, часть работы сотрудников деканатов и учебного отдела будет автоматизирована, что позволит оптимизировать затраты рабочего времени. Благодаря единой базе данных сотрудники разных отделов не будут многократно вводить повторяющиеся данные. Формирование большинства документов, относящихся к учебным планам и РУП, будет автоматизировано. Автоматическое формирование документов уменьшит «человеческий фактор», то есть ценное время сотрудников не будет расходоваться не только на заполнение документов, но и на многократную проверку корректности информации. У сотрудников отпадет необходимость задерживаться на работе во внеурочное время, а значит уменьшатся затраты вуза на оплату труда.

Список литературы

1. Официальный сайт министерства образования и науки РФ www.mon.gov.ru
2. Официальный сайт лаборатории ММИС – разработчика программного комплекса «GosInsp» www.mmis.ru
3. Официальный сайт Алтайского государственного технического университета им. И.И. Ползунова www.altstu.ru

ПОДХОДЫ К РЕШЕНИЮ ЗАДАЧИ ПОИСКА ПАРЫ БЛИЖАЙШИХ ТОЧЕК В ПРОСТРАНСТВАХ С БОЛЬШОЙ РАЗМЕРНОСТЬЮ

Есипенко С.П. – студент

Алтайский государственный технический университет (г. Барнаул)

Введение

Во многих прикладных задачах требуется для заданного объекта находить самый похожий объект из некоторого множества. Например, такая задача возникает при распознавании образов, классификации объектов, сжатии и поиске данных, создании искусственного интеллекта. Можно привести и конкретные примеры: пусть требуется распознать сканированный текст, представленный в виде битового образа. Несложно выделить в нем отдельные битовые образы символов. Затем достаточно распознать эти образы, то есть для каждого найти самый похожий символ. Точно также распознается и рукописный текст, только образы представлены набором кривых.

Данные задачи имеют эффективные решения, потому что объекты в них достаточно простые. Но есть и такие, в которых необходимо работать с очень сложными объектами, которые характеризуются большим числом параметров. В общем случае, можно представить эти объекты как точки в некотором многомерном пространстве, и ввести в нем расстояние (меру близости). Тогда необходимо найти просто ближайшую точку к данной.

Также стоит отметить, что на практике часто требуется искать ближайшие объекты сразу для группы объектов. Отсюда вытекает следующая задача.

Постановка задачи

Дано два множества точек в k -мерном пространстве. Пусть их размеры N_1 и N_2 соответственно. Предполагается, что точки имеют вещественные координаты и равномерно распределены в k -мерном квадрате со стороной 1. Требуется найти пару точек из разных множеств, евклидово расстояние между которыми минимально. Конечно, далеко не во всех практических задачах объекты лежат в обычном евклидовом пространстве, но в рамках данной статьи будет рассмотрен этот вариант.

Обзор основных методов решения

Очевидно, что самый простой алгоритм, перебирающий всевозможные пары точек будет работать за время $O(N_1 \cdot N_2 \cdot k)$. Далее будем называть его базовым.

В пространствах с небольшой размерностью существуют более эффективные алгоритмы. Все эти алгоритмы можно разбить на две группы:

- алгоритмы работающие с разбиением пространства (kD -деревья, R -деревья, Morton order);
- алгоритмы непрерывного хэширования (Locality-sensitive hashing).

Идея этих методов заключается в группировке близко расположенных точек, что позволяет значительно сократить число пар-кандидатов. При небольших размерностях пространства (до 10) данные алгоритмы позволяют добиться времени работы порядка $O(\min(N_1, N_2) \cdot \log(\max(N_1, N_2)))$ в среднем, и даже в худшем случае для ряда алгоритмов. Но в данной оценке опускается размерность пространства, которую принимают за константу. Проблема заключается в том, что при больших размерностях данные алгоритмы работают не лучше базового (обычно хуже из-за скрытых констант). Эту проблему также называют «проклятием размерностей» («curse of dimensionality»). Фактически, эффективно сгруппировать точки в пространстве можно, если их количество экспоненциально растет с ростом размерности.

Таким образом эффективного алгоритма решения поставленной задачи на данный момент не существует. Но можно ослабить задачу, и попытаться найти пару достаточно

близких точек. Данная задача имеет более менее эффективные решения. Далее рассмотрим упрощенный вариант одного из них.

Алгоритм случайного проецирования

Идея данного алгоритма проста – если две точки расположены близко, то любые их проекции также будут располагаться на небольшом расстоянии друг от друга. Самый простой способ, проецировать точки на случайный вектор. Тогда несложно найти пару ближайших точек, проекции которых близко расположены. Делается это так: находятся проекции, точки обоих множеств сортируются по величине проекции (в любом порядке). Далее делается прямой проход по точкам, и точки одного множества заносятся в стек. Для каждой точки второго множества, просматривается несколько кандидатов с вершины стека. Затем делается аналогичный обратный проход. Необходимо выполнить несколько итераций данного алгоритма для различных случайных векторов, чтобы найти более менее оптимальный ответ. В целом, данный алгоритм ведет себя достаточно неплохо, даже на больших наборах данных, но все же он находит далеко не оптимальный ответ. Как бы то ни было, он является одним из лучших на данный момент (для произвольных, равномерно распределенных точек).

Вывод

Оценим эффективность алгоритма. Одна итерация требует подсчет всех проекций за $O(k \cdot (N_1 + N_2))$ (в случае неевклидовых пространств оценка может быть хуже), сортировку за $O((N_1 + N_2) \cdot \log(N_1 + N_2))$, и два линейных прохода. В итоге получаем сложность $O(I \cdot (k \cdot (N_1 + N_2) + (N_1 + N_2) \cdot \log(N_1 + N_2)))$. Требования к памяти небольшие – $O(k \cdot (N_1 + N_2))$. Итерации можно выполнять в разных потоках, причем для этого не потребуются существенных модификаций алгоритма.

В данной статье рассматривалось самое простое решение поставленной задачи, которое находит субоптимальный ответ. Тем не менее, оно дает неплохой результат. Более того, оно легко обобщается на различные пространства (достаточно лишь эффективно считать проекцию).

Список литературы

1. Препарата Ф., Шеймос М. Вычислительная геометрия: Введение/ пер. с англ. – М.: Мир, 1989. – 478 с.
2. Connor M., Kumar P. Fast construction of k-Nearest Neighbor Graphs for Point Clouds. – IEEE Transactions on visualization and computer graphics, 2009.

АРХИТЕКТУРА РАСПРЕДЕЛЕННОЙ СИСТЕМЫ ДЛЯ ОБРАБОТКИ СОБЫТИЙ И СООБЩЕНИЙ В РЕЖИМЕ РЕАЛЬНОГО ВРЕМЕНИ

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

В настоящее время мы можем наблюдать за проникновением различных электронных устройств практически во все сферы человеческой деятельности. Можно ожидать, что в обозримом будущем на каждого человека будет приходиться несколько десятков различных датчиков и «умных» устройств [1]. Сразу же возникает вопрос о том, как такое многообразие систем будет взаимодействовать друг с другом. В данной работе предлагается архитектура

программного комплекса, открывающего новые возможности в области организации взаимодействия устройств и обработки событий и сообщений в режиме реального времени.

Разрабатываемый программный комплекс включает в себя черты брокера обмена сообщениями, пространственно-временной базы данных и ядра автоматизированной системы управления. Предлагаемый комплекс направлен на решение следующих задач:

1. Организация Machine-to-Machine взаимодействия устройств и приложений посредством обмена сообщениями;
2. Обработка возникающих событий с обеспечением достаточной скорости реакции, необходимой для управления жизненно важными процессами;
3. Надежное хранение получаемых данных с устройств и сенсоров в нормализованном виде и превращение сырых данных в полезную информацию.

1. Состав комплекса

Программный комплекс состоит из двух основных компонентов: распределенная серверная часть и интерфейс управления. Серверная часть реализована на основе Erlang/OTP и обеспечивает все необходимые функции для организации M2M-взаимодействия. Интерфейс управления реализован с использованием веб-фреймворка Python/Django и позволяет производить настройку серверной части. Связь между двумя составляющими осуществляется через общую реляционную базу данных конфигурации.

2. Организация кластера

Серверная часть комплекса имеет распределенную кластерную архитектуру. Каждый узел кластера соответствует узлу Erlang VM [2]. Внутри узла запускается несколько приложений (модулей) системы, реализующих отдельные сервисы. Взаимодействие между модулями осуществляется при помощи механизмов обмена сообщениями Erlang/OTP.

В ходе работы кластера формируется ячеистая логическая топология сети (англ. Mesh), в которой каждый узел соединяется с несколькими другими. При этом весь кластер разбивается на области по физическому и логическому расположению узлов, образуя **зоны**. Предполагается, что время обращения к ресурсам в пределах одной зоны должно быть значительно меньше, чем время обращения к ресурсам других зон. Например, в отдельную зону могут быть выделены узлы, расположенные в рамках одного центра обработки данных. В процессе работы кластер будет минимизировать пересылки данных между зонами.

Добавление новых узлов в кластер осуществляется без остановки всей системы. Управляющий модуль кластера отслеживает проблемы связности и неисправности узлов, изменяя активную схему взаимодействия модулей при необходимости. Обмен данными осуществляется преимущественно в асинхронном режиме. Для обеспечения качества обслуживания производится сбор статических данных о нагрузке системы в целом [3].

3. Модули

Серверная часть программного комплекса имеет модульную структуру и состоит из четырех основных модулей: core, storage, frontend, script. Каждый модуль представляет из себя отдельное приложение Erlang со своим отдельным OTP-деревом процессов и супервизоров [4]. Модули взаимодействуют с друг с другом исключительно при помощи обмена сообщениями, не опираясь на данные о внутренней структуре, что обеспечивает слабую связность. Приложение модуля core запускается на каждом узле кластера, остальные же модули могут быть запущены в необходимом количестве в зависимости от нагрузки и настроек узлов.

Модуль core координирует работу кластера и реализует такие функции, как поддержка связности между узлами, разрешение имен, подписка на уведомления, поиск ближайших узлов, работа с конфигурацией системы и т.д. Каждый экземпляр **core** поддерживает локальную карту топологии кластера и реестр списка запущенных приложений, синхронизируя информацию с остальными узлами. Для вызова тех или иных функций модулей системы необходимо обращение к **core** для получения идентификатора процесса-обработчика. Дальнейшее взаимодействие осуществляется напрямую, минуя **core**.

Модуль storage реализует пространственно-временной хранилище. API модуля предоставляет функции для создания таблиц определенной структуры, получения данных по временному диапазону, а также сортировке и фильтрации результата по значениями индексированных полей. Каждый экземпляр приложения **storage** работает с определенными группами таблиц (каналов). По идентификатору канала осуществляется балансировка нагрузки, при этом для надежности данные каждого канала хранятся на нескольких узлах, обеспечивая необходимую степень избыточность (определяется конфигурацией).

Модуль frontend реализует сеансовые протоколы, обеспечивающие взаимодействие клиентов с подсистемами и функциями комплекса. Процесс **transport_base** предоставляет «сырое» Erlang API, процессы **transport_http**, **transport_mqtt**, **transport_snmp** предоставляют доступ через соответствующие протоколы. Клиенты получают одинаковые возможности при подключении к любому из запущенных экземпляров модуля в рамках одного кластера.

Модуль script отвечает за выполнение кода обработчиков, вызываемых при возникновении событий. Приложение подгружает исходные коды из конфигурационной базы, компилирует их в машинный или байт-код (зависимости от используемого языка) и запускает с данным об возникшем событии. Каждый скрипт запускается в «песочнице» (англ. Sandbox) с ограниченными правами, имея доступ только к модулю **frontend** на уровне обычного клиента.

4. Обработка событий

Все получаемые модулем **frontend** данные нормализуются и преобразуются в сообщения определенной структуры. Для определения формата сообщений вводится понятие «канал». Каждый канал определяется уникальным идентификатором и списком полей тела сообщения. Формат описания полей сравним со структурой таблицы базы данных. В частности, возможно указание имени поля, типа данных, размерности и значения по умолчанию. Формат данных строго проверяется для всех типов протоколов.

После получения и разбора нового события в канале модуль **frontend** передает данные для записи ближайшим экземплярам процесса **storage**. Сообщение также параллельно отправляется в модуль **script**, который запускает необходимые скрипты-обработчики. Скрипты реализуют определенную логику обработки событий и могут записывать данные в другие каналы, а также обращаться к внешним сервисам, если это разрешено в рамках «песочницы». Политика ограничения доступа определяется в зависимости от типа песочницы.

Выборка данных также осуществляется при помощи обращения к **frontend**. При выполнении запросов необходимо указывать идентификатор канала и интересующий временной диапазон. Результаты запроса могут быть отфильтрованы и отсортированы по значениям полей. Клиенты сервиса могут подписаться на получение сообщений. Появление новых данных в канале может служить в качестве сигнала о необходимости выполнения тех

или иных действий устройством. В зависимости от используемого протокола поддерживается как схема с опросом (англ. Polling), так и работа через сокеты.

5. Управление системой

Параметры конфигурации программного комплекса сохраняются в реляционной базе данных. Модуль **core** предоставляет доступ к параметрам настройки для остальных модулей. Серверная часть использует таблицы конфигурации исключительно для чтения, тогда как интерфейс администрирования позволяет изменять необходимые параметры.

В конфигурационной базе содержится следующая информация:

1. Информация о пользователях;
2. Данные авторизации;
3. Права доступа;
4. Описания каналов и их полей;
5. Исходные коды скриптов;
6. Параметры и условия запуска обработчиков;
7. Справочные данные.

Реляционная база данных предназначена исключительно для работы с конфигурацией системы и не используется для хранения данных каналов. Часть конфигурации кешируется экземплярами **core** для ускорения доступа. Для увеличения степени надежности комплекса возможен запуск slave-серверов в режиме чтения в каждой из зон кластера.

Заключение

Рассмотрена архитектура программного комплекса для обработки событий и сообщений в режиме реального времени. Разрабатываемая система представляет из себя новый класс ПО, включающий в себя черты брокера обмена сообщениями, пространственно-временной базы данных и ядра автоматизированной системы управления. Заложенный уровень абстракции позволяет поддерживать несколько различных по своей природе протоколов. Предложенный механизм обработки событий и сообщений при помощи скриптов позволяет использовать комплекс в качестве фреймворка для разработки более сложных систем.

Список литературы

1. Цисык Р.О. Принципы реализации сервисов предоставления информационных услуг в среде глобальных вычислительных сетей, [Текст] / Р.О. Цисык, Е.Н. Крючкова // НАУКА И МОЛОДЕЖЬ – 2010: VII Всероссийская научно-техническая конференция студентов, аспирантов и молодых ученых. — Барнаул: изд-во АлтГТУ, 2010. – С. 81–88.
2. Armstrong, J. Programming Erlang: Software for a Concurrent World, [Текст] / J. Armstrong. — Pragmatic Bookshelf, 2007.
3. Цисык Р.О. Оценка качества обслуживания в распределенных системах, [Текст] / Р.О. Цисык, Е.Н. Крючкова // XVII Международная научно-практическая конференция студентов, аспирантов и молодых ученых «Современные техника и технология». — Томск: изд-во Томского политехнического университета, 2011. – С. 450–451.
4. Supervisor Behaviour, [Электронный ресурс] // Erlang/OTP Design Principles User's Guide. Режим доступа: http://www.erlang.org/doc/design_principles/des_princ.html

АВТОМАТИЗАЦИЯ УЧЕТА ПОКАЗАТЕЛЕЙ ВНЕУРОЧНОЙ ДЕЯТЕЛЬНОСТИ В РАМКАХ МУНИЦИПАЛЬНОЙ СИСТЕМЫ ОЦЕНКИ КАЧЕСТВА ОБРАЗОВАНИЯ

Деменева Н.В. – студент, Сучкова Л.И. – к.т.н., профессор,
Алтайский государственный технический университет (г. Барнаул)

Повышение качества образования признано приоритетным направлением государственной политики в сфере образования. На современном этапе вопросы объективной оценки качества образования являются актуальными как для педагогической общественности, так и для родителей, т. е. в целом для всего общества.

Учитывая, что образовательное учреждение представляет собой сложную социальную систему, выделяют три основных объекта контроля качества:

1. **Качество условий для осуществления образовательного процесса** - контроль за соответствием и соблюдением: лицензионных условий и требований, законодательства РФ, противопожарной безопасности т.п.
2. **Качество образовательного процесса** - контроль за соответствием направления ФГОС, базисному учебному плану, учебного плана ОУ, рабочих учебных программ примерным учебным программам; выполнение учебного плана, рабочих учебных программ.
3. **Качество результатов образовательного процесса** - контроль за уровнем подготовки выпускников и обучающихся ОУ по урочной и внеурочной деятельности.

Внеурочная деятельность занимает значительную часть в образовательном процессе, поиске личности и профориентации школьников, а также мотивации учителей [1].

Внеурочная работа тесно связана с дополнительным образованием школьников – составной частью системы образования и воспитания детей и подростков, которая ориентирована на свободный выбор и освоение учащимися дополнительных образовательных программ. Связующим звеном между внеурочной работой и дополнительным образованием детей выступают различные факультативы, школьные научные общества, объединения профессиональной направленности, учебные курсы по выбору .

Учет показателей внеурочной деятельности в учебном заведении в электронном виде практически отсутствует. В настоящее время известны программные системы, предназначенные для автоматизации процессов управления учебным заведением общего среднего образования. Большая часть таких систем поддерживает достаточно ограниченный круг функций, связанных с управлением учебным заведением, имеет собственную структуру данных, которая не позволяет обеспечить горизонтальный (от одного учреждения другому) и вертикальный (информация для органов управления образованием на уровне города или района) обмен данными. В основном, системы направлены на управление учебным процессом и объединяют в себе такие функции как: управление учебными материалами, управление учебными планами, тестирование и оценка, отчетность по успеваемости учащихся.

Был реализован программный продукт, направленный на автоматизацию учета показателей внеурочной деятельности в рамках муниципальной системы качества образования. Он представляет собой web-приложение, реализованное на языках HTML и PHP.

Для того, чтобы система оценки качества образования выполняла свое назначение, предусмотрено осуществление четырех основных технологических функций:

1. **получение (сбор)** фактических данных;
2. статистическую и аналитическую **обработку** собранных данных;
3. **предоставление** обработанных данных потребителям;
4. **хранение** данных в течение установленных сроков [2].

Их функциональные связи представлены на рисунке 1



Рисунок 1 – Функциональные связи основных технологических процессов

Сбор данных, их обработка, а также порядок предоставления и хранения описаны в методических указаниях [3]. Там содержится перечень показателей оценки и методика их расчета, источники информации и регламент проведения сбора, порядок предоставления информации (уровень доступа).

Спроектированная база данных содержит информацию о педагогах, об обучающихся и о конкурсах. О педагоге хранятся сведения об образовании, список должностей, преподаваемый предмет, классное руководство, дата последней аттестации, стаж, имеющиеся награды, сведения о пройденных курсах, отношение к методическим объединениям и вклад в их работу, участие в сетевых сообществах. Информация об обучающихся включает в себя данные о социальном положении, форме обучения, успеваемости, отношении ученика к классу и школе, информация о кружках. Информация о конкурсах включает название конкурса, период проведения, территориальную принадлежность, данные об организаторе конкурса. В базе отмечается участие педагогов и обучающихся в различных конкурсах, с указанием предмета, по которому проводился конкурс и полученные результаты.

Приложение предусматривает авторизованный вход. В зависимости от прав становятся доступны следующие функции:

1. Просмотр показателей внеурочной деятельностью (всем);
2. Просмотр информации о педагоге, в том числе участие в конкурсах (группа «Педагоги»);
3. Просмотр информации об обучающемся, в том числе участие в конкурсах(группа «Обучающиеся»);
4. Просмотр информации о конкурсах (группы «Педагоги» и «Обучающиеся»);
5. Просмотр информации о методических объединениях(группа «Педагогов»);
6. Заполнение информации о педагоге (группа «Модераторы»);
7. Заполнение информации об обучающемся(группа «Модераторы»);
8. Заполнение и изменение всех данных (администратор).

Список литературы

1. Развитие региональной системы оценки качества образования Алтайского края. Сборник документов. Часть 1. / Сост. Ю.Г. Абдуллаев. – Барнаул: Азбука, 2009. – 204 с.
2. Региональная система оценки качества образования в Алтайском крае. Сборник методических материалов. Часть 2. – Барнаул, 2011 – 204 с.
3. Организация внутреннего мониторинга качества образования в образовательном учреждении. Методические рекомендации для руководителей образовательных учреждений, специалистов муниципальных органов управления образованием. / Сост. С.Н. Кучер, Т.М. Гозман. - Барнаул, 2010. – 130 с.

МЕТОДЫ ПЕРЕХВАТА СИСТЕМНЫХ СООБЩЕНИЙ И ВЫЗОВОВ В ОС WINDOWS NT

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

Введение

Задача сбора данных о приложении непосредственно в процессе его выполнения представляет большой практический интерес. Полученная информация может быть полезна при тестировании, отладке, профилировании, исследовании структуры и состава приложения (reverse engineering), изучении взаимодействия программы и операционной системы, а также анализе действий пользователя при использовании приложения.

Одним из способов сбора такой информации является перехват системных сообщений и вызовов. Основными преимуществами этого подхода являются отсутствие необходимости внесения изменений в исходный и исполняемый код целевой программы, что фактически позволяет осуществлять мониторинг любого запущенного приложения, и возможность проследить за взаимодействием компонентов приложения не только между собой, но и с операционной системой. Безусловным недостатком является техническая сложность, вызванная тем, что большинство существующих методов решения задачи предполагают низкоуровневое взаимодействие с целевым процессом посредством системного API.

В данной работе рассмотрено несколько методов реализации перехвата системных сообщений и вызовов в ОС Windows NT, а также описаны их преимущества и недостатки.

Общие вопросы реализации

Любая система слежения за приложением состоит из двух ключевых компонентов (см. рисунок 1):

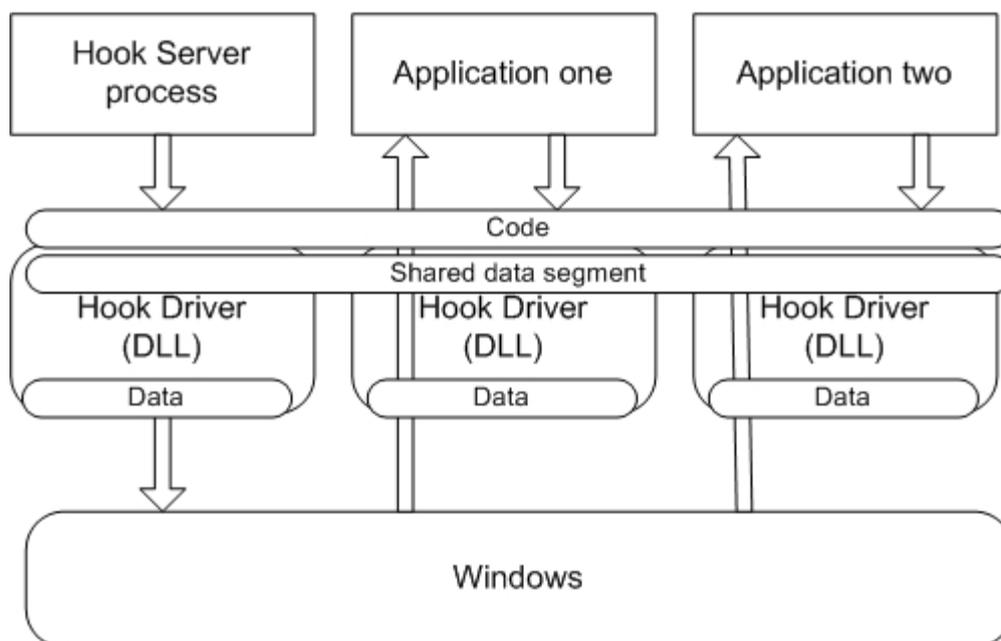


Рисунок 1 — Схема системы слежения

1. Сервер слежения (*hook server*)

Осуществляет мониторинг выполняемых процессов, внедряет следящий агент в нужные процессы, передаёт команды агенту и получает от него данные с помощью межпроцессного взаимодействия.

2. Агент слежения (*hook driver*)

Выполняет непосредственный перехват сообщений и вызовов API, работая в адресном пространстве целевого процесса, и обменивается данными с сервером. Код агента содержится в динамически подключаемой библиотеке (DLL).

Методы реализации каждого из этих компонентов и способ их взаимодействия выбираются в зависимости от целей системы слежения. В некоторых случаях можно полностью избежать реализации агента, используя определённые API ОС (например, UI Automation). Если свой агент необходим, необходимо выбрать как метод его внедрения, так и способ перехвата сообщений и вызовов. Кроме того, необходимо определиться, требуется ли слежение за всеми приложениями либо за одним или несколькими.

Внедрение агента в процесс

Так как агент является динамически подключаемой библиотекой (DLL), то задача сводится к нахождению способов подключения DLL к произвольному процессу. Рассмотрим несколько способов.

Системный реестр

В Windows существует стандартный механизм подгрузки дополнительных библиотек в приложения, использующие стандартную библиотеку USER32.DLL. Для этого имя требуемой библиотеки должно быть указано в значении определённого ключа в реестре. Преимущество данного метода — использование стандартных средств системы, что не требует усилий со стороны программиста. Из недостатков можно выделить отсутствие контроля за процессом подгрузки (библиотека загружается при запуске любого процесса и выгружается только при его завершении), а также невозможность внедрения в консольные приложения (они, как правило, не используют USER32.DLL).

Windows hooks

Механизм Windows Hooks (hook – перехватчик) позволяет внедряться в обработку событий любого потока. Используя функцию SetWindowsHookEx, можно задать процедуру обратного вызова (callback), находящуюся в DLL, которая будет выполняться при возникновении определённого события в указанном потоке. Windows автоматически подгрузит указанную DLL при первом возникновении требуемого события. Заметим, что данный механизм сам по себе является достаточно мощным средством, которое можно использовать для реализации агента, однако в данном случае мы используем лишь одно свойство этого механизма — то, что он позволяет выполнить произвольный код в любом процессе. Таким образом, мы можем выполнить вызов функции LoadLibrary в контексте нужного процесса, загрузив в него требуемую DLL, и затем удалить теперь уже ненужный перехватчик (UnhookWindowsHookEx). Более того, мы можем поместить код агента в ту же DLL, в которой находится callback-функция — в этом случае LoadLibrary просто увеличит количество ссылок на данную DLL, поэтому она не выгрузится при вызове UnhookWindowsHookEx.

Как и использование реестра, Windows hooks позволяют внедрить DLL стандартными средствами без лишних усилий. Однако преимуществом данного механизма является возможность выгрузить DLL по требованию — для этого лишь нужно установить ещё один перехватчик, выполняющий вызов FreeLibrary. Недостатком является тот факт, что при необходимости внедрения агента в новые процессы сервер слежения должен либо обнаруживать появление процессов, используя дополнительные механизмы, либо использовать перехватчик для всех потоков системы, не удаляя его. Последний способ может серьёзно повлиять на производительность ОС в целом.

Функции CreateRemoteThread и LoadLibrary

Функция CreateRemoteThread позволяет создать поток во внешнем процессе. Однако в качестве параметра ей должен быть передан адрес функции ThreadProc, также находящийся во внешнем процессе и имеющей определённую сигнатуру, поэтому выполнить произвольный код, используя только эту функцию, не удастся. Для внедрения агента необходимо использовать тот факт, что функция LoadLibrary присутствует практически в любом процессе и, более того, имеет везде один и тот же адрес (это следствие нахождения этой функции в базовой системной библиотеке KERNEL32.DLL). Заметим также, что сигнатура этой функции аналогична сигнатуре ThreadProc. Таким образом, мы можем выполнить LoadLibrary в любом процессе, что нам и требуется. Точно так же можно выполнить и выгрузку библиотеки (FreeLibrary имеет совместимую сигнатуру).

Данный метод является наиболее гибким из всех рассмотренных, однако требует ручной загрузки библиотеки в каждый процесс.

Перехват сообщений и вызовов

Внедрение агента позволяет получить доступ к адресному пространству процесса, что, в свою очередь, даёт возможность выполнять перехват системных сообщений и вызовов. Существует множество способов сделать это, рассмотрим лишь некоторые из них.

UI Automation

Технология UI Automation позволяет получать информацию об элементах графического интерфейса пользователя (GUI), используемых приложениями, и подписываться на события, связанные с этими элементами. При этом не требуется реализация агента и межпроцессного взаимодействия — всё это предоставляется системой. Технология основана на COM и имеет высокоуровневую реализацию в .NET. Перечисленные факторы существенно облегчают труд программиста, однако данный подход имеет и недостатки. Можно выделить недостаточную поддержку WPF-элементов, невозможность работы со сторонними GUI-фреймворками (Swing, Qt и т.д.) и проблемы с получением информации об источнике события в некоторых

случаях. Технология тесно связана с GUI и не позволяет перехватывать низкоуровневые сообщения или вызовы API.

Windows hooks

Рассмотренный в предыдущем разделе механизм Windows hooks предназначен именно для перехвата сообщений, причём на достаточно низком уровне. В тех случаях, когда нужен тщательный анализ взаимодействия пользователя с интерфейсом, данный механизм хорошо подходит. Однако при его использовании для всех потоков системы возможно замедление производительности, так как дополнительная работа будет производиться для каждого сообщения.

Изменение таблицы адресов импортированных функций

Данный метод перехвата вызовов API основывается на факте, что адреса всех внешних функций, используемых некоторой библиотекой, хранятся в специальной таблице в исполняемом коде программы (Import Address Table). При доступе к памяти процесса эти адреса могут быть изменены на адреса функций-посредников (проxy), фиксирующих интересующие нас вызовы. Так как таблицу адресов сравнительно легко найти и изменить, данный способ является одним из самых простых и надёжных среди всех способов подмены вызываемых функций. Однако, безусловно, он требует достаточно трудоёмких и легко подверженных ошибкам манипуляций с памятью процесса.

Заключение

Рассмотренные методы перехвата системных сообщений и вызовов могут быть использованы при реализации широкого спектра ПО — от систем учёта рабочего времени до средств автоматизированного тестирования интерфейса пользователя.

Список литературы

1. Ivanov I. API hooking revealed [Электронный ресурс] / I. Ivanov. – Режим доступа: <http://www.codeproject.com/Articles/2082/API-hooking-revealed>
2. Kuster R. Three Ways to Inject Your Code into Another Process [Электронный ресурс] / R. Kuster. – Режим доступа: <http://www.codeproject.com/Articles/4610/Three-Ways-to-Inject-Your-Code-into-Another-Process>
3. Richter J. Windows via C/C++ / J. Richter. – 5-е изд. – Microsoft Press, 2007. – 900 с.

РАЗРАБОТКА АВТОМАТИЗИРОВАННОЙ СИСТЕМЫ ПО РАБОТЕ С ПАЦИЕНТАМИ БОЛЬНИЦЫ (НА ПРИМЕРЕ КРАЕВОГО ГОСУДАРСТВЕННОГО БЮДЖЕТНОГО УЧРЕЖДЕНИЯ ЗДРАВООХРАНЕНИЯ «ЦЕНТРАЛЬНАЯ РАЙОННАЯ БОЛЬНИЦА ИМЕНИ А. Ф. ВОРОБЬЕВА»)

Петренко О.Р. – студент, Бубнова Н.Д. – ст. преподаватель
Алтайский государственный технический университет (г. Барнаул)

Актуальность создания информационной системы в медицинских учреждениях обусловлена сегодня необходимостью использования больших и постоянно растущих объемов информации при решении диагностических, терапевтических, статистических, управленческих и других задач.

Ни для кого не секрет, что большая часть времени приема пациента уходит не на решение клинических вопросов, а на сопроводительную и далеко не самую основную работу – оформление поликлинических талонов и другой отчетной документации, записей в амбулаторной карте или истории болезни, назначений консультаций или обследования и т.д.

Уже не вызывает сомнений, что наиболее эффективным инструментом для облегчения труда медицинских сотрудников и повышения его эффективности являются компьютерные технологии. Автоматизация способна не просто облегчить работу, она должна освободить персонал от рутинной работы и дать ему принципиально новый инструмент, который прямо или косвенно, но приведет к сокращению нецелевого расхода интеллектуального багажа, реализации желания работать и заниматься именно медициной.

На сегодняшний момент существует множество программных продуктов, которые значительно упрощают работу медицинских сотрудников, например, такие как:

- Автоматизированная информационно-поисковая система "Регистратура-Экономист" от компании ООО «ФОБОС»;
- qMS компании «СП. АРМ».

Недостатками рассмотренных продуктов являются:

- Высокая стоимость;
- Отсутствие необходимых функций или же наоборот – слишком широкий набор функций, в некоторых из которых нет надобности;
- Невозможность развития системы. Программное обеспечение является готовым запатентованным продуктом, поэтому нет возможности изменять содержание программного продукта под требования конкретной организации - пользователя.

При обследовании изучаемого объекта, а именно Центральной районной больницы, был сделан вывод, что все не функции рассмотренных продуктов действительно необходимы данной организации, поэтому возникла необходимость разработки программного обеспечения с необходимым функционалом, более подходящим для конкретной организации.

Разработанный программный продукт имеет 3 режима доступа: медрегистратор, врач и администратор, каждому соответствует определенный набор прав доступа.

Автоматизация рабочего места медрегистратора включает следующий набор функций:

- Ведение электронных амбулаторных карт с возможностью изменения персональных данных;
- Запись пациентов на прием;
- Выдача талонов;
- Распределение пациентов по участкам, учет пациентов на каждом участке;
- Составление отчетов по приемам врачей за промежуток времени.

Автоматизация рабочего места врача включает следующее:

- Планирование и составление расписания;
- Оформление направлений;
- Составление отчетов по прогнозированию сезонной либо другой массовой заболеваемости;
- Составление отчетов о возможных посещениях пациентов, по запланированным лечением;
- Внесение информации, связанной с обследованиями и лечением;
- Ведение электронной истории болезни;
- Отчеты по проведенным лечением за промежуток времени.

Администратор имеет право добавлять пользователя и устанавливать права доступа для пользователя, может фиксировать действия пользователя: вход в систему, выход из системы, выполняемые операции, такие как добавление, изменение, удаление. Это позволит отслеживать действия сотрудника в случае несанкционированного доступа к данным.

В целом, функционал программы состоит из следующих пунктов:

- Ведение базы данных, отражающей деятельность учреждения по учету пациентов;
- Запись пациентов на прием с учетом занятости врачей;
- Ведение амбулаторной карты пациента с отметками обо всех визитах, проведенных осмотрах и назначениях;
- Регистрация поставленных диагнозов;
- Ведение электронной истории болезни;
- Ведение учета пациентов на каждом участке;
- Составление необходимых внутренних отчетов о работе клиники, таких, как:
 - Приемы выбранного врача (всех врачей) за заданный промежуток времени;
 - Возможные посещения пациентов, по запланированным процедурам;
 - История проведенных процедур за заданный промежуток времени
 - Запланированные посещения выбранного врача (всех врачей) за заданный промежуток времени;
 - Прогнозы сезонной либо другой массовой заболеваемости.

Главной задачей поликлиники является обеспечение пациентов высококвалифицированной, быстрой и эффективной медицинской помощью. В условиях постоянно растущего потока пациентов и увеличивающегося объема информации, система позволит при высоком качестве обслуживания больных повысить качество обработки информации, снизить нагрузку на сотрудников, избавив их от выполнения многочисленных рутинных операций.

Разработанный программный продукт предназначен для сбора и хранения информации с последующей ее обработкой. Программный продукт предоставляет возможность вводить, изменять и хранить персональные данные обо всех клиентах больницы, диагнозах и прививках, приемах и оказанных услугах, упрощать поиск необходимой информации и предоставлять необходимые печатные формы документов.

В программе имеется ряд справочников: справочник областей, справочник районов, справочник городов, справочник специальностей врачей, справочник социального статуса, справочник цели посещения, справочник статуса посещения, справочник диагнозов, справочник характера заболевания. Так как организация работает с несколькими районами, предусмотрен справочник районов. При работе с программой часто требуется информация, уже введенная в справочники. Например, при заполнении диагноза пациента необходимо ввести диагноз, вызвав при этом справочник диагнозов. Если в справочнике отсутствует информация, предусмотрено оперативное добавление ее без прерывания основной работы.

Обоснование выбора комплекса технических и программных средств

Для разработки программы было использовано следующее программное обеспечение.

Программа «Microsoft Visual C# 2008 Express Edition». Для функционирования Microsoft Visual Studio 2008 Express Edition требуется операционная система семейства Windows.

Для создания базы данных выбрана свободная система управления базами данных MySQL. MySQL имеет API для языков Delphi, C, C++, Эйфель, Java, Лисп, Perl, PHP, Python,

Ruby, Smalltalk и Tcl, библиотеки для языков платформы .NET, а также обеспечивает поддержку для ODBC посредством ODBC-драйвера MyODBC. Таким образом, с MySQL можно работать и из Microsoft VisualC # 2008. СУБД MySQL очень популярна на сегодняшний день и знакома многим разработчикам программного обеспечения.

Для функционирования данного приложения необходимо следующее программное обеспечение: операционная система Microsoft Windows XP/2003/Vista/Seven, Microsoft .NET Framework не ниже версии 2.0, текстовый редактор MicrosoftWord 2002 и выше, сервер БД MySQL 5.5.3 и ПО, указанное выше.

Для нормального функционирования ПО к оборудованию предъявляются следующие требования: необходим ПК с процессором семейств Intel®Pentium®/Celeron®/Itanium®/Xeon™, AMD K6/Athlon™/Duron™/Turion™ или совместимым с ними процессором, тактовая частота которого составляет не менее 1 ГГц, оперативная память 512 МБ и выше, любой жесткий диск со свободным местом не менее 100 Мб, видеоплата и монитор с разрешением не менее 800х600 точек, клавиатура, мышь.

Список литературы

1. Медицинский портал. Организация работы регистратуры поликлиники [Электронный ресурс]. – Режим доступа:
http://medvuz.info/load/ozz_obshhestvennoe_zdorove_i_zdravookhranenie/organizacija_raboty_registratury_polikliniki/36-1-0-665.

РАЗРАБОТКА СТРУКТУРЫ ФИНАНСОВО – КОММЕРЧЕСКИХ ОПЕРАЦИЙ

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

Коммерческая деятельность всегда связана с поиском более выгодного распределения ресурсов или денежных средств. Любая фирма хочет получить как можно больше прибыли, затратив при этом как можно меньше денежных и трудовых ресурсов. Эти задачи наилучшим образом можно решить с помощью экономико – математических моделей, которые позволяют моделировать реальные ситуации финансово – коммерческой деятельности.

Любая финансово-коммерческая операция предполагает какие-то условия, которые должны быть согласованы участвующими сторонами. К условиям могут относиться денежные суммы, процентные ставки и т.д. И все операции требуют определенных знаний и навыков об их применении и расчете. Например, процентная ставка может быть рассчитана более десяти различными способами (платежи могут даваться на различное время, сумма выплат может меняться в течение времени, а может оставаться постоянной). В рамках одной финансовой операции все экономические показатели образуют взаимосвязанную систему. Естественно, что при изменении одного из параметров этой системы, даже на незначительную величину, изменятся и остальные параметры системы. Следовательно, именно такие системы и должны являться объектом исследования количественного финансового анализа. При этом фактор времени (даты, сроки выплат, периодичность поступления денежных средств, отсрочка платежей и т.д.) зачастую играет более важную роль, чем стоимостные характеристики финансовой операции, поскольку именно он определяет конечный финансовый результат.

Для построения экономико – математической модели необходимо описать эти факторы, установить связи (информационную, логическую, временную) между ними.

На основе изученного материала, исходя из требования оптимизации финансово – коммерческих операций была построена укрупненная структура системы задач финансовой математики (Рис.1.).

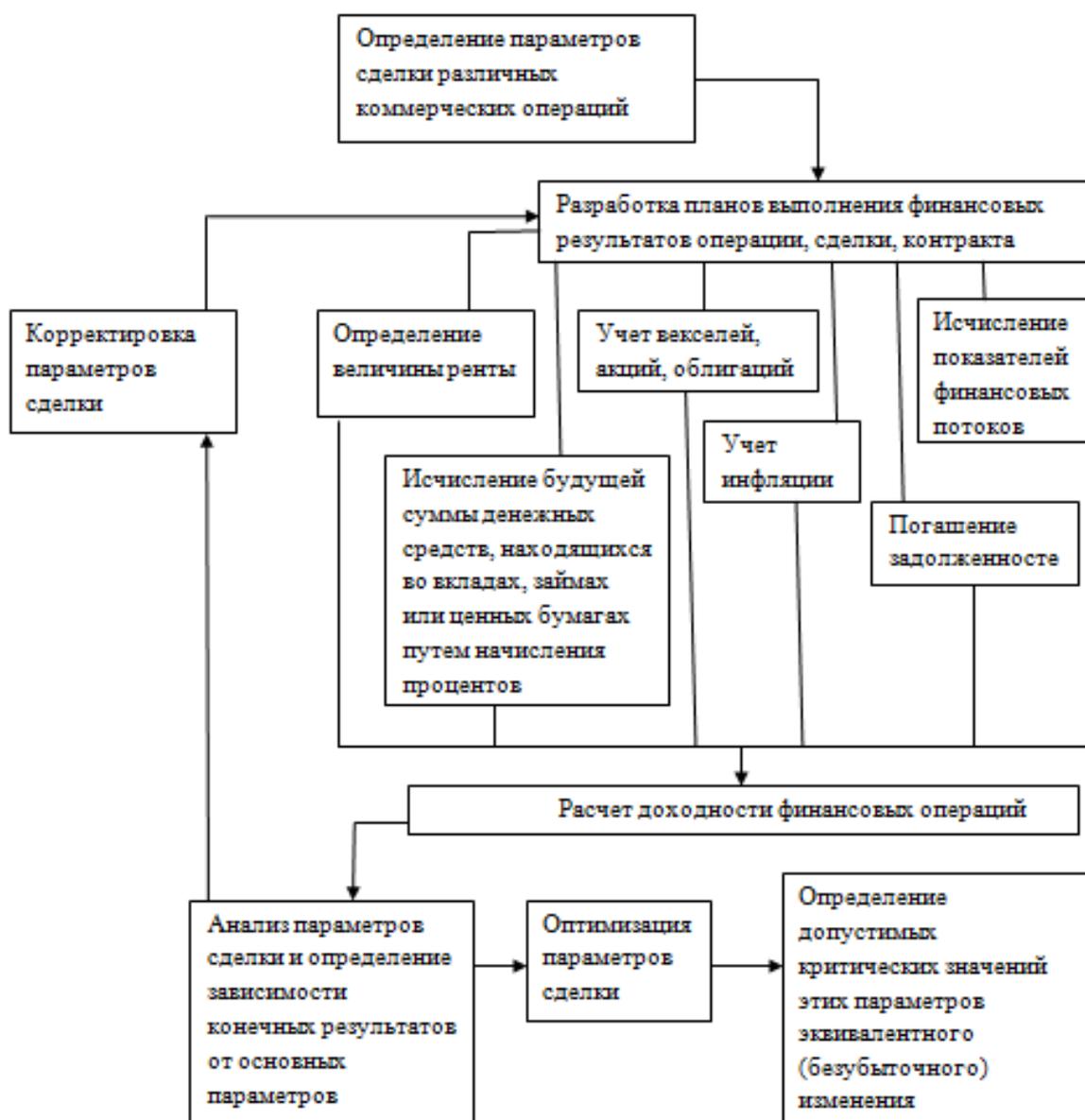


Рисунок 1 – Укрупненная структура системы задач финансовой математики

В дальнейшем в данной работе предполагается разработка программного комплекса финансово – коммерческих операций, в том числе моделей развития операций по схеме простых и сложных процентов, операций дисконтирования, финансовых и товарных потоков, инфляции в коммерческих операциях, сравнения финансово – коммерческих операций, операций с облигациями и операций с акциями, а также анализ этих моделей для выявления более выгодного распределения ресурсов или денежных средств.

Список литературы

1. Фомин Г.П. Математические модели и модели в коммерческой деятельности. – М:Финансы и статистика, 2005. – 616 с.
2. Четыркин Е.М. Финансовая математика. – М:Дело, 2005. – 400 с.

РЕАЛИЗАЦИЯ СИСТЕМЫ ПРОЕКТИРОВАНИЯ И ТЕСТИРОВАНИЯ КАРКАСОВ ПРОГРАММНЫХ СИСТЕМ

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

Современные методы и практики быстрой разработки программного обеспечения нацелены на минимизацию рисков, путём сведения процесса к серии коротких циклов, каждый из которых представляет программный проект в миниатюре и включает все задачи, необходимые для выдачи минимального прироста по функциональности. При таком подходе особенно важно представлять всю систему целиком на ранних этапах разработки. Для решения подобных задач применяются прототипы программных систем, которые позволяют вести разработку проекта методом непрерывной интеграции [1].

Прототип архитектуры программной системы – это упрощённое представление желаемой архитектуры, без учёта деталей и аспектов реализации. Прототипы архитектуры создаются, чтобы смоделировать будущую систему в целом. Ни один из отдельных модулей в прототипе не должен быть особенно функциональным. Основная цель создания прототипа архитектуры в том, чтобы смоделировать разрабатываемую систему и понять, как она будет выглядеть в собранном виде, опуская детали [2].

Поскольку большинство прототипов создается с целью моделирования рассматриваемой системы в целом, то полученный прототип есть не что иное, как одноразовая программа, необходимая для того, чтобы получить ответы на определенный ряд вопросов. Главная цель составления такого прототипа – это получить понимание того, как система будет выглядеть в собранном виде, не вдаваясь в детали и особенности. С этой точки зрения может показаться удобным создание прототипов посредством языков очень высокого уровня, а точнее языков более высокого уровня по сравнению с языком, используемым при написании системы. К таким языкам, например, можно отнести Perl и Python. Язык сценариев высокого уровня позволяет опустить многие детали (например, указание типов данных), но при этом создавать функциональный фрагмент программы. Такие языки также позволяют при необходимости соединить низкоуровневые фрагменты в новые сочетания.

Согласно приведенным выше рассуждениям, можно сделать вывод о том, что основополагающая проблема прототипирования архитектур программных проектов заключается в отсутствии на рынке целого класса специализированных систем, комплексно удовлетворяющих выдвинутым требованиям. Кроме того, современные тенденции развития методологий и практик в области быстрой интегральной разработки программного обеспечения, лишь подтверждают необходимость в появлении подобных инструментов прототипирования.

Основополагающая идея предлагаемого подхода заключается в использовании унифицированного языка в процессе описания прототипов, а также свойств конечных автоматов в процессе тестирования.

Описание прототипа, составленное с использованием унифицированного языка, является входными данными для системы. Затем на основе описания идёт построение его внутреннего представления. Внутренне представление прототипа должно позволять провести анализ исходного прототипа. С этой точки зрения в качестве внутреннего представления можно выбрать конечный автомат, представленный графом переходов. На основе полученного внутреннего представления можно провести его оптимизацию [3] (например, детерминизацию и минимизацию графа переходов) и его проверку с учётом исходных требований. Будем считать, что прототип является корректным, если в его внутреннем представлении возможен переход из каждого состояния конечного автомата в любое другое состояние. Иными словами, конечный автомат должен представлять собой полный граф (каждая пара различных вершин является смежной). После тестирования и

оптимизации прототипа можно переходить к этапу кодогенерации, на котором внутреннее представление транслируется в описание на целевом языке.

Исходя из вышесказанного, можно выделить следующие основные стадии в работе системы:

- лексический анализ;
- парсинг;
- оптимизация внутреннего представления;
- анализ внутреннего представления;
- тестирование.

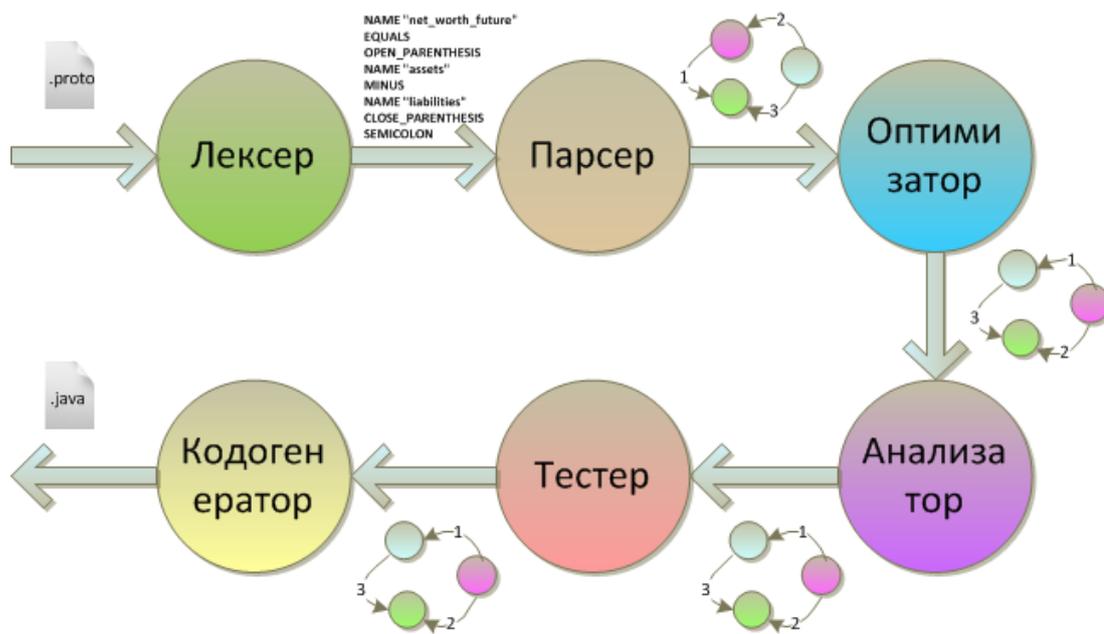


Рисунок 1 – Модель системы

В соответствии с выделенными стадиями работы системы можно разделить всю систему на несколько подсистем (см. Рисунок 1):

- Лексер – делает лексический анализ исходного описания прототипа;
- Парсер – строит внутреннее представление на основе потока токенов;
- Оптимизатор – выполняет ряд оптимизации над внутренним представлением, например, детерминизацию и минимизацию графа переходов для упрощения процесса тестирования;
- Анализатор – осуществляет поиск шаблонов проектирования, типичных ошибок в построении архитектуры, формирует отчет и рекомендации на основе полученных данных;
- Тестер – производит тестирование внутреннего представления на предмет работоспособности архитектуры и соответствия предъявляемым требованиям. В результате работы тестер выдает однозначный ответ о пригодности прототипа;
- Кодогенератор – позволяет получить исходный код внутреннего представления на целевом языке.

В некотором смысле, можно считать модель рассмотренной системы моделью оптимизирующего компилятора, включающего дополнительную стадию тестирования и осуществляющего трансляцию описания прототипов с унифицированного исходного языка в целевой для рассматриваемой платформы.

Автором было проведено исследование современных решений в области прототипирования программного обеспечения, оценка их эффективности и применимости согласно выдвинутой модели требований, а также выводы о необходимости появления нового класса инструментов прототипирования, в виду неготовности существующих решений удовлетворять ранее выдвинутым требованиям. Кроме того, в работе детально представлена предлагаемая автором модель системы описания и тестирования прототипов архитектуры программного обеспечения и рассмотрена ее реализация с точки зрения современных технологий программирования.

Список литературы

1. Макконелл С. Совершенный код. Мастер-класс. Практическое руководство по разработке программного обеспечения / С. Макконелл. – Издательство «Русская редакция», 2011. – 896 с.
2. Хант Э. Программист-прагматик. Путь от подмастерья к мастеру / Э. Хант, Д. Томас - СПб.: Питер, 2010. – 830 с.
3. Ахо А. Компиляторы: принципы, технологии и инструментарий, 2-е издание / А. Ахо, М. Лам, Р. Сети, Д. Уильямс – ООО «И. Д. Вильямс», 2011. – 1184 с.

СТОХАСТИЧЕСКАЯ ДИНАМИКА ЛИНЕЙНОЙ МАКРОМОЛЕКУЛЫ

Трегубова Ю.Б. – аспирант, Пышноград Г.В. – д.ф.-м.н., профессор*
 Алтайский государственный технический университет (г. Барнаул)
 Алтайская государственная педагогическая академия* (г. Барнаул)

Структура полимерных жидкостей более сложна по сравнению со структурой твердых тел и низкомолекулярных жидкостей. При этом они совмещают в себе черты тех и других. Заметим также, что механическое поведение полимерных систем также сочетает упругость твердого тела и текучесть жидкости, его определяют как вязкоупругое [1]. Это свойство является одним из проявлений медленных релаксационных процессов, которые связывают, прежде всего, с релаксацией отдельной макромолекулы в системе [2, 3].

Работа посвящена формулировке и решению уравнений динамики макромолекулы осуществляется в рамках микроструктурного (статистического) подхода. Такой подход позволяет учитывать как молекулярное строение вещества, так и процессы межмолекулярного взаимодействия.

Всякая макромолекула может быть эффективно представлена как цепочка связанных броуновских частиц (так называемая модель гауссовых субцепей или шариков и пружин [4]). При этом макромолекула разбивается на N субцепей длиной M/N каждая, а поведение макромолекулы описывается движением линейной цепочки из $N+1$ броуновских частиц, связанных между собой последовательно упругими силами.

Рассмотрим движение макромолекулы подробнее. Согласно второму закону динамики, ускорение частицы, прямо пропорционально силе, действующей на нее, и обратно пропорционально ее массе.

$$m \frac{d^2 r_i^\alpha}{dt^2} = F_i^\alpha, \quad \alpha = 0, 1, 2, \dots, N, \quad (1)$$

Динамика рассматриваемой макромолекулы может быть упрощена допущением, что соседние макромолекулы описываются как единообразная бесструктурная среда и все

важные взаимодействия могут быть превращены во внутримолекулярные взаимодействия, так что крупномасштабная стохастическая динамика единичной макромолекулы в такой системе может быть рассмотрена как динамика эффективной броуновской частицы.

Сила в правой части уравнения (1) представляет собой результат действия на частицу сил различной природы: диссипативные силы, силы упругости «пружин» между соседними частицами, случайная сила.

Пренебрегая взаимным гидродинамическим взаимодействием частиц в линейном по скоростям приближении, динамика единичной цепочки может быть описана набором стохастических уравнений [5]

$$m \frac{d^2 r_i^\alpha}{dt^2} = -\zeta u_i^\alpha + F_i^\alpha + G_i^\alpha - 2\mu T A_{\alpha\gamma} r_i^\gamma + \phi_i^\alpha(t), \quad \alpha = 0, 1, 2, \dots, N, \quad (2)$$

где m – масса броуновской частицы, которая моделирует участок макромолекулы длины M/N , r^α и u^α – координаты и скорость броуновской частицы, ζ – коэффициент трения в «мономерной» жидкости, F_i^α и G_i^α представляют эффективные силы соседних макромолекул: F_i^α – сила гидродинамического увлечения, G_i^α – сила внутренней вязкости; ϕ_i^α – случайная сила, $2T\mu$ – коэффициент упругости пружины между соседними частицами, T – температура в энергетических единицах. Матрица $A_{\alpha\gamma}$ описывает тот факт, что броуновские частицы связаны в единую линейную цепочку и имеет вид

$$\begin{pmatrix} 1 & -1 & 0 & \dots & 0 \\ -1 & 2 & -1 & \dots & 0 \\ 0 & -1 & 2 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & 1 \end{pmatrix}. \quad (3)$$

Сопротивление «мономерной» жидкости, а также, силы F_i^α и G_i^α представляют диссипативные силы в уравнениях (2). Причем эффективные силы соседних макромолекул (F_i^α и G_i^α) удовлетворяют уравнениям

$$\tau \frac{d(F_i^\alpha + G_i^\alpha)}{dt} + F_i^\alpha + G_i^\alpha = -\zeta B H_{ij}^{\alpha\gamma} u_j^\gamma - \zeta E G_{ij}^{\alpha\gamma} u_j^\gamma, \quad (4)$$

где τ – время релаксации. Сила F_i^α – это сила гидродинамического увлечения, в то время как G_i^α – сила внутренней вязкости, обладающая свойством

$$\sum_{\alpha=0}^N G_i^\alpha = 0, \quad i = 1, 2, 3. \quad (5)$$

Коэффициенты B и E введены как меры интенсивности внешних и внутренних добавочных диссипативных сил. Они определяются окружением рассматриваемой макромолекулы. В исключительных случаях зависят от длины соседних макромолекул.

Случайная сила в уравнениях (2) может быть представлена как сумма двух независимых процессов

$$\phi_i^\alpha(t) = \bar{\phi}_i^\alpha(t) + \tilde{\phi}_i^\alpha(t), \quad (6)$$

причем первое слагаемое – это гауссовский процесс с корреляцией

$$\langle \bar{\phi}_i^\gamma(t) \bar{\phi}_j^\mu(t') \rangle = 2T\zeta \delta_{\gamma\mu} \delta_{ij} \delta(t-t') \quad (7)$$

и второе – также гауссовский, но не дельта-коррелированный процесс.

Решение представленной системы дифференциальных уравнений проводилось методом Рунге-Кутты. В результате решения получали траектории частиц. Для того, чтобы уменьшить влияние случайных сил и проанализировать релаксационные свойства полученной физической системы проводилось достаточно большое количество вычислений, а затем усреднение полученных траекторий [6].

В ходе исследования было обнаружено, что результаты не зависят от числа N ($N > 2$) субцепей в моделируемой макромолекуле. Это позволяет говорить о корректности выбранной модели макромолекулы и схемы описания динамики ее поведения.

Также было проведено сравнение численного и аналитического решений. Согласно [7] выражение для вычисления среднего смещения центра масс макромолекулы для модели (2) – (7) в линейном приближении имеет вид

$$\langle R \rangle = 6D_0 \frac{\tau}{B} \left(\frac{t}{\tau} + 1 - e^{-tB/\tau} \right), \quad (8)$$

где D_0 – коэффициент диффузии, B – мера увеличения коэффициента трения частицы, τ – время релаксации среды.

Графики аналитического и численного решений очень близки. Качественное поведение численного и аналитического решений идентичны. В обоих случаях имеется ярко выраженный горизонтальный участок (плато). Это плато демонстрирует задержку в диффузии макромолекулы, связанную с наличием упругого взаимодействия между частями макромолекулы. Положение плато характеризуется временем его возникновения или величиной смещения, при которой наблюдается отмеченная задержка. То есть, наличие такого плато показывает существование в полимерной системе внутреннего масштаба, который может быть как временным (время релаксации), так и пространственным (диаметр «трубки»). До и после плато наблюдаются участки, когда смещение центра масс макромолекулы пропорционально времени наблюдения [6].

При этом численное решение учитывает влияние анизотропии подвижности на характер движения линейной макромолекулы в вязкоупругой среде, чего не удается достичь, решая систему аналитически. Влияние анизотропии подвижности на величину смещения центра масс макромолекулы становится заметным с момента выхода зависимости на плато и продолжает ощущаться с увеличением времени наблюдения. Решение демонстрирует наличие в системе различных пространственных масштабов и соответствующих времен релаксации, которые вводились Де Женем и Эдвардсом в своих теориях [2, 3].

Список литературы

1. Ferry J.D. Viscoelastic Properties of Polymers [Текст] / J.D. Ferry. – 3rd ed. – London: Wiley, 1980.
2. De Gennes P.G. Scaling Concepts in Polymer Physics [Текст] / P.G. De Gennes. – Ithaca, NY: Cornell Univ. Press, 1979.
3. Doi M., Edwards S.F. The Theory of Polymer Dynamics [Текст] / M. Doi, S.F. Edwards. – Oxford Univ. Press, Oxford, 1986.
4. Покровский В.Н. Динамика слабо связанных линейных макромолекул [Текст] / В.Н. Покровский // Успехи физических наук. – 1992. – № 5. – Т. 162. – С. 87-121.
5. Pokrovskii V.N. A justification of the reptation-tube dynamics of a linear macromolecule in the mesoscopic approach [Текст] / V.N. Pokrovskii // Physica. – 2006. – v.A366. – P. 88-106.
6. Трегубова Ю.Б. К обоснованию рептакционного механизма диффузии линейной макромолекулы в теории микровязкоупругости [Текст] / Ю.Б. Трегубова, Ю.А. Алтухов, И.В. Третьяков // ФПСМ. – 2011. – №4. – С. 27-31.
7. Кокорин Ю.К. Механизм сверхмедленных релаксационных процессов в неразбавленных линейных полимерах [Текст] / Ю.К. Кокорин, В.Н. Покровский // Высокомолекулярные соединения. – 1990. – Т. А32. – № 12. – С. 2409-2417.

ПРИМЕНЕНИЕ НЕЙРОННЫХ СЕТЕЙ ДЛЯ РУБРИКАЦИИ ТЕКСТОВ НА ЕСТЕСТВЕННОМ ЯЗЫКЕ

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

На сегодняшний день, в условиях стремительно растущего объема информации, актуальна задача автоматической классификации и упорядочивания информации, представленной на естественном языке (ЕЯ). Одной из подзадач этой области является рубрикация (распределение информации согласно ее тематики). Различные решения данной задачи находят свое применение во многих областях, таких как: обработка новостей, фильтрация спама, классификация библиотечных материалов и т.д.

Среди существующих методов обработки текстов на ЕЯ можно выделить два класса - методы машинного обучения и методы, основанные на знаниях. Методы машинного обучения используют представления текста в виде векторов признаков и анализ этих векторов. К таким методам можно отнести: метод Байеса, метод Роше, метод опорных векторов, метод k -ближайших соседей. Методы, основанные на знаниях (экспертные системы), используют знания эксперта. Хорошим примером может служить система, основанная на правилах «если ... - то ... », в которой посылка (часть если) соответствует условию, а заключение (часть то) — действию.

В данной работе представлен метод машинного обучения для классификации текстов на русском языке, который использует искусственную нейронную сеть для анализа характеристик текста. В свою очередь характеристики текста основываются не только на ключевых словах, но и на словах, связанных с ключевыми по смыслу.

Исходными данными является словарь, основанный на толковом словаре Ожегова, в котором представлен перечень термов в нормальной форме. Каждому терму сопоставлен набор термов, связанных с ним ассоциативной, синонимической и т. д. связью. Таким образом, словарь представляет собой граф, где вершины — слова, а ребра имеют весовые коэффициенты от 0 до 1. Модифицированным методом Дейкстры находим все вершины графа (термы), удаленные от ключевых термов рубрики на расстоянии не превышающем заданное значение:

$$R(k) = \sum_{i=0..n} \prod W_i, \quad (1)$$

где k – вершина графа, W_i – вес ребра с индексом i , n – количество ребер, связующих вершину k с вершиной ключевого терма. Размер входного слоя нейронной сети равен количеству ключевых термов рубрики. Выходной слой имеет два выхода по значению которых можно сделать вывод — принадлежит ли текст данной рубрике или нет. Обрабатываемый текст (будь то текст из выборки обучения или текст, рубрику которого необходимо определить) подвергается анализу. Термы текста приводятся к нормальной форме. Для этого задействована библиотека lucene — кроссплатформенная, свободно распространяемая Java библиотека, разработанная компаний Apache (www.apache.org) и библиотека russianmorphology (<http://code.google.com/p/russianmorphology/>). Считается удаленность (1) терма от каждого ключевого терма рубрики. Полученные значения суммируются с соответствующими значениями других термов текста:

$$S[k] = \sum_{i=0..n} W_i, \quad (2)$$

где W_i – расстояние от терма i до ключевого терма k , n – количество термов текста. Полученные значения нормируются. Таким образом, на выходе получаем значения для входного слоя нейронной сети. Далее происходит обучение нейронной сети с помощью

обучающей выборки (тексты с заведомо известной рубрикой). И после обучения искусственная нейронная сеть готова к анализу текста.

Полученные результаты тестов дали относительно хороший результат — от 78% до 85 % текстов были успешно распознаны.

Список литературы

1. Automatic Classification Research Group - Методы машинного обучения / Режим доступа: <http://acrg.ru/algorithms.html>
2. Агеев А.М. Методы автоматической рубрикации текстов, основанные на машинном обучении и знаниях экспертов / Режим доступа: www.cir.ru/docs/ips/publications/2005_diss_ageev.pdf
3. Дунаев Е.В. Автоматическая рубрикация web-страниц в интернет-каталоге с иерархической структурой / Режим доступа: www.ict.edu.ru/vconf/files/11556.pdf

РАЗРАБОТКА ИНСТРУМЕНТАЛЬНОЙ СИСТЕМЫ ДЛЯ ИССЛЕДОВАНИЯ НЕЧЕТКИХ ВРЕМЕННЫХ РЯДОВ С ПРИМЕНЕНИЕМ ПРАВИЛ СВОБОДНОЙ СЕМАНТИКИ

Мезенцев В.Ю. – студент, Сучкова Л.И. – к.т.н., профессор
Алтайский государственный технический университет (г. Барнаул)

Концепция нечетких множеств была впервые изложена Лотфи Заде (Lotfi A. Zadeh) в 1965 году. Практически с этого же момента начинается развитие аппарата нечетких множеств (fuzzy sets), позволяющего описывать понятия естественного языка (лингвистические конструкции) и присущую им неопределенность. Фактически, нечеткие множества образуют важное связующее звено между символическими и численными вычислениями, являясь ключевым инструментом вычислений со словами (computing with words), ориентированных на использование слов и предложений языка человеческого общения. Перспективным направлением применения концепции нечетких вычислений является Fuzzy Time Series Data Mining, предусматривающий накопление реальной истории изменения некоторых числовых и лингвистических характеристик объектов и построение заключений о вероятном будущем тренде. Целью анализа любого временного ряда (ВР) является достижение понимания причинных механизмов, обусловивших поведение изучаемого процесса, построение моделей ВР, которые не только объясняют поведение процесса, но и могут быть использованы для оценки прогноза развития изучаемого процесса.

Применение аппарата нечеткого множества дает возможность для конечного пользователя оперировать естественными предметно-ориентированными лингвистическими терминами, представляемыми на уровне компьютерных вычислений в виде чисел. Под лингвистической переменной будем понимать переменную, значения которой представляют собой нечеткие термы. Терм - лингвистическая фраза, используемая в качестве семантического обозначения соответствующей функции принадлежности. Примером термов лингвистической переменной «скорость ветра» могут быть «слабый», «сильный», «шторм» и т.д., являющиеся экспертной оценкой измеряемой величины. Экспертная оценка величины, описывающей отдельное состояние процесса, системы, объекта, полученная в некоторый момент времени, является по существу качественной интервальной оценкой локального состояния. Упорядоченная во времени последовательность таких оценок состояний системы и объектов представляет собой временной ряд экспертных оценок [1]. Характерной особенностью такого ВР является нечеткость его значений, вытекающая из природы

экспертных оценок, поэтому такой временной ряд относится к классу нечетких временных рядов (НВР).

Существующие качественные и количественные подходы к прогнозированию ВР ставят своей целью увеличение точности предсказания в максимальной степени, однако, традиционные методы не могут применяться к задачам, в которых исторические данные представлены не в числовом виде, а в словесной форме. Нечеткие временные ряды позволяют преодолеть эту проблему, открывая реальную перспективу возможности обработки как лингвистических, так и числовых данных. Согласно теореме FAT (Fuzzy Approximation Theorem) любая математическая система может быть аппроксимирована системой, основанной на нечеткой логике, что позволяет с помощью естественно-языковых высказываний «ЕСЛИ-ТО» сколько угодно точно описать произвольную взаимосвязь «входы-выходы» без использования сложного математического аппарата. Однако использование только высказываний «ЕСЛИ-ТО» не позволяет вводить операции над обрабатываемыми лингвистическими термами, осуществлять анализ истории тренда НВР, работать с группой НВР, оперировать с временными аспектами НВР, например, с продолжительностью наблюдения лингвистического термина. Кроме того, для информационно-измерительных систем особое значение имеют функции прогнозирования нештатных ситуаций, в которых важны не только текущие значения группы НВР, но и их динамика.

Для исследования НВР будем использовать метод свободной семантики, опирающийся на описание зависимостей и прогнозируемых термов в НВР с применением высказываний естественного языка [2]. Разработан КС-язык, правила которого позволяют оперировать разнородными данными – лингвистическими термами, как текущими, так и взятыми из истории, четкими значениями, нечеткими значениями изменений отсчетов ВР, влияющими на прогнозируемое значение ВР. Созданная инструментальная система позволяет создавать лингвистические переменные, задавать с помощью функций принадлежности их термы, предоставляя необходимые возможности для редактирования и визуального представления.

Инструментальная система также дает возможность задания пользователем четких временных рядов для дальнейшей фаззификации. Фаззификация производится на основе уже заданной лингвистической переменной, а также позволяет выбрать переменную для создания дополнительного нечеткого ряда, состоящего из приращений четкой величины. Предусмотрена возможность выполнения формирования нечеткого ряда из четких значений с требуемой периодичностью. Отметим, что в инструментальной системе время является условной величиной, отсчеты НВР начинаются с 0 и далее возрастают, а для четких рядов – данных измерений время хранится в формате UTC.

Для контроля и прогнозирования нештатных ситуаций для каждого ряда предусмотрены граничные функции, описывающие допустимые пределы изменения контролируемого параметра, в рамках которых поведение ВР и НВР является инвариантным. Граничные функции могут быть описаны с учетом сезонности, периодичности, характерных особенностей поведения процесса во времени.

В инструментальной системе для прогнозирования НВР предусмотрен интеллектуальный анализ на базе паттернов поведения системы. Паттерн содержит в себе информацию об изменении во времени значений по некоторому множеству рядов. Пользователь всегда может добавить значения, которые наиболее точно передают поведение системы во времени, и на основании которых можно составить некоторый возможный прогноз, если во время исследования состояние системы наиболее полно соответствует конкретному паттерну поведения.

Язык, предоставляемый пользователю для прогнозирования, позволяет обращаться независимо к различным нечетким рядам, а также рядам изменения состояния процессов, дает возможность проверять сопоставляемые границы для некоторых значений времени. Возможно введение новых переменных, обращение к заданным лингвистическим

переменным и переменным, характеризующим время, построение условных переходов и циклов. В выражениях, кроме базовых нечетких арифметических операций и операций нечеткого сравнения, введены логические операции и функции фаззификации и дефаззификации значений. Кроме того разработанный язык для удобства выявления зависимостей между различными нечеткими рядами содержит функции, опирающиеся на логику Аллена, рассматривающую базовые отношения между интервалами, такие как предшествование, завершение, пересечение и т.д.

Кроме прогнозирования на основе задаваемых правил свободной семантики сохраняется возможность прогнозирования классическим методом (модифицированный стационарный метод, предложенный Дегтяревым [3]).

Разработанная инструментальная система для исследования НВР на базе правил свободной семантики позволит проверять догадки эксперта о поведении НВР на существующих данных, а также о прогнозируемых значениях и возможных приближениях системы к нештатным ситуациям. Предложенный подход дает качественные способы описания поведения сложных организационно-технических систем.

Список литературы

1. Ярушкина Н. Г. Интеллектуальный анализ временных рядов : учебное пособие [текст] / Н. Г. Ярушкина, Т. В. Афанасьева, И. Г. Перфильева. – Ульяновск : УлГТУ, 2010. – 320 с.
2. Модели представления и поиска нечетких темпоральных знаний в базах данных временных рядов. [Электронный ресурс]. – Режим доступа: http://nsmv2008.ulstu.ru/docs/modeli_predstavleniya.pdf
3. Прогнозирование валютных котировок с использованием модифицированного стационарного метода, основанного на нечетких временных рядах. [Электронный ресурс]. – Режим доступа: <http://www.exponenta.ru/educat/news/degtyarev/paper2.pdf>

РАЗРАБОТКА МОДУЛЯ ВИЗУАЛИЗАЦИИ ДЛЯ УЧЕТА ПОСЕЩЕНИЯ МАССОВЫХ МЕРОПРИЯТИЙ С ПРОИЗВОЛЬНОЙ СХЕМОЙ РАЗМЕЩЕНИЯ ЗРИТЕЛЕЙ

Козлова И.И. – студент, Сучкова Л.И. – к.т.н., профессор
Алтайский государственный технический университет

В последние годы в России внедряется система сканирования билетов. В данной системе каждый билет обладает уникальным штрих-кодом.

На входе в зал расположенные специальные устройства, которые сканируют билет. При сканировании происходит мгновенная онлайн-идентификация подлинности предъявленной распечатки, что исключает возможность подделки билетов. Предъявить на входе копию билета также невозможно, поскольку в системе мгновенно отмечается, что посетитель уже находится в зале. Штрих-код имеет миллионы комбинаций, поэтому подделать его невозможно.

Однако у данной системы есть ощутимый недостаток: отсутствует визуальное оформление присутствия посетителя в зале. Поэтому возникла задача разработки визуального интерфейса для данной системы, который мог бы поставляться как отдельное приложение, обеспечивающее достоверные данные по заполненности зала.

В последние годы стала популярной продажа билетов по сети Internet. Существует большое количество сайтов, которые предоставляют такие услуги.

Рассмотрим сайт kontramarka.ru [1]. Выбор билетов на продажу на этом сайте осуществляется по схеме зала. Приложение для выбора места по схеме зала представляет собой Java

Applet и является частью большого модуля по продаже и бронированию билетов на мероприятия.

На рисунке 1 представлено окно заказа билетов по схеме зала.

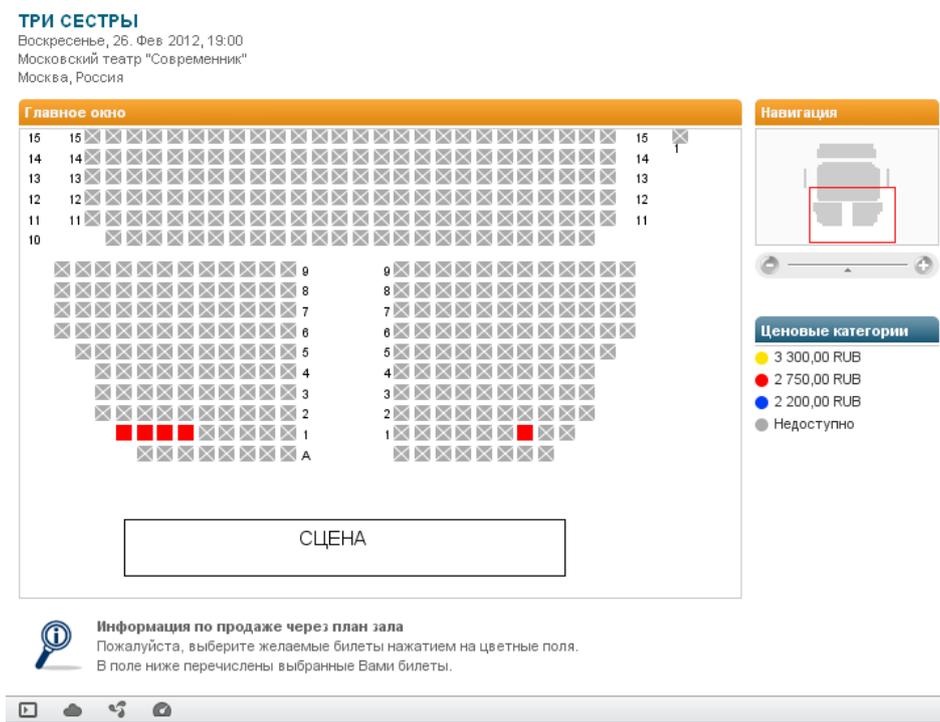


Рисунок 1 – Заказ билетов по схеме зала

В главном окне изображена схема зала. Квадратами обозначены места. Цвет места зависит от ценовой категории. Возможно изменение масштаба с различной степенью детализации, что хорошо при больших залах. Однако так как данное приложение является частью модуля по продаже билетов, то не представляется возможности использовать его отдельно от этого модуля.

К сожалению, графический редактор залов в настоящее время не существует, как самостоятельный проект. Вследствие этого возникла необходимость разработки приложения, позволяющего создавать схемы размещения мест, категории мест, ярусы, которые затем могли бы использоваться для учета посещения массовых мероприятий.

Разработанный модуль состоит из двух приложений:

1. Представляет собой интерфейс для рисования зала и сохранение его в xml файле;
2. Интерпретирует этот файл в объект и отображает его для работы пользователя.

Первое приложение предоставляет следующий функционал:

- Создание секторов с заданным количеством рядов и мест в каждом ряду;
- Изменение количества рядов в секторе;
- Изменение количества мест в ряду;
- Изменение местоположения секторов, рядов;
- Просмотр зала с произвольным масштабом;
- Нумерация рядов, мест, задание наименования сектора и вывод соответствующих подписей на схеме;
- Создание сцены, как из одного элемента, так и из нескольких;
- Сохранение схемы как xml документ;
- Открытие ранее созданной схемы.

Второе приложение, интерпретирует xml файл и отображает схему зала для работы пользователя, а также отмечает на схеме занятые места. Приложение имеет возможность функционировать в комплексе с системой сканирования билетов.

Список литературы

1. Официальный сайт [kontramarka.ru](http://www.kontramarka.ru) [Электронный ресурс]. – Режим доступа: <http://www.kontramarka.ru/ru/>

РАЗРАБОТКА ГИС НА БАЗЕ КАРТОГРАФИЧЕСКОГО СЕРВЕРА С ОТКРЫТЫМ КОДОМ

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

Геоинформационные системы (ГИС) являются классом информационных систем, имеющим свои особенности. Они построены с учетом закономерностей геоинформатики и методов, применяемых в этой науке. ГИС как интегрированные информационные системы предназначены для решения различных задач науки и производства на основе использования пространственно - локализованных данных об объектах и явлениях природы и общества. Неразрывно с ГИС связаны геоинформационные технологии. Геоинформационные технологии можно определить как совокупность программно-технологических средств получения новых видов информации об окружающем мире. Геоинформационные технологии предназначены для повышения эффективности: процессов управления, хранения и представления информации, обработки и поддержки принятия решений. ГИС имеет ряд особенностей, которые необходимо учитывать при изучении этих систем. Одна из особенностей ГИС и геоинформационных технологий состоит в том, что они являются элементами информатизации общества. Это заключается во внедрении ГИС и геоинформационных технологий в науку, производство, образование и применение в практической деятельности получаемой информации об окружающей реальности.

Геоинформационные технологии в последнее время успешно внедряются во многие прикладные задачи. Сама идея отображения и хранения картографической информации на цифровом носителе не нова, но из-за относительно больших требований к аппаратным ресурсам активное использование началось недавно в связи с резким увеличением производительности компьютеров и снижением их стоимости.

Любая геоинформационная система должна состоять из нескольких частей. Первой основополагающей частью является источник картографической информации. Эта информация привязывается к одной из существующих систем координат, и отображается либо в виде набора растровых изображений, либо в виде набора векторных слоев. Первый вариант чаще всего используется для отображения реальных снимков поверхности земли, сделанных со спутников. Второй представляет собой множество векторных примитивов, из которых в совокупности получаются карты.

Предоставлением картографической информации обычно занимается картографический сервер. На сегодняшний день на рынке существует большое количество таких сервисов. У них есть как локальные версии, так и версии, работающие через Интернет. Основными критериями при выборе того или иного картографического сервера являются его рентабельность, поддержка им существующих форматов отображения и передачи картографических данных, производительность, стоимость. Последний пункт имеет

огромное значение, т.к. на рынке есть отличный продукт компании ESRI, ArcGIS, но его могут позволить себе только крупные предприятия.

Вторая часть геоинформационной системы — это собственно отображаемая на картах информация. Она может иметь различный характер — от простого справочника объектов до информации на местности, получаемой с датчиков в реальном времени. Всю эту информацию удобно хранить в СУБД, привязывая ее к картографическим объектам.

В ИВЭП СО РАН идет разработка геоинформационной системы «Реестр водных объектов Обь-Иртышского бассейна». Картографическим сервером, используемым для этой ГИС, выбран проект с открытым исходным кодом на Java – GeoServer. Причины такого выбора просты. GeoServer бесплатен, поддерживает множество стандартов картографической информации, имеет достаточно простой веб-интерфейс, поддерживает плагины. Также, благодаря использованию Java, GeoServer является кроссплатформенным приложением, которое должно будет работать на основных распространенных операционных системах.

На данный момент в тестовом режиме работает сервис, предоставляющий и топографическую информацию о территории Обь-Иртышского бассейна. Подготовлены для отображения на карте данные мониторинга, в том числе, данные о качестве воды на различных участках речной сети, об ее количестве и использовании.

Для использования разрабатываемой ГИС в системах поддержки принятия решений необходимо расширение ее функциональности, а также обеспечение разработчиков ГИС удобными инструментальными средствами.

Авторами статьи ведутся работы по отображению имеющихся данных на картах в виде деловых диаграмм. С их помощью можно будет анализировать состояние объектов, а также составлять отчеты.

Также необходимо разработать приложение, создающее удобную работу со способами отображения слоев карты. Это приложение должно будет генерировать файлы специального формата, которые потом будут использоваться в GeoServer. Потребность в такой утилите обусловлена тем, что спецификация OpenGIS Styled Layer Descriptor, используемая в GeoServer, подразумевает хранение информации, формирующей вид слоя, в текстовом xml-подобном формате. Правка такого файла непосредственно трудоемка и имеет слабую обратную связь.

Модуль для отображения деловой графики реализуется модулем, выполняющимся в среде OSGI. Приложение для генерации файла отображения слоев реализуется Java-приложением.

Вся разработка ведется в интегрированной среде разработки Eclipse 3.x.

Методы и технологии, разработанные и опробованные в процессе создания ГИС «Реестр водных объектов Обь-Иртышского бассейна», позволят создавать другие системы, имеющие прикладное значение для различных организаций и предприятий.

РАЗРАБОТКА МОДУЛЯ ПРЕПАРИРОВАНИЯ ИЗОБРАЖЕНИЙ ДЛЯ СИСТЕМЫ РАСПОЗНАВАНИЯ НА БАЗЕ WEB-ИНТЕРФЕЙСА

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

В настоящее время технология оптического распознавания текста стала очень востребованной. Этому способствует переход большинства компаний к безбумажной технологии – подготовке чертежей, схем, карт, документов с помощью компьютера. При

этом уже существующие в бумажном виде документы переводятся в электронный вид. Этот процесс осуществляется с помощью OCR (Optical Character Recognition) систем.

В данное время на рынке присутствуют множество продуктов данного направления, перечислим некоторые из них:

- ABBY FineReader – Продукт от российской компании АБВУ, занимает лидирующие позиции на рынке оптического распознавания текста. Разработчиком представлены версии для домашнего, индивидуального и корпоративного использования. Так же недавно компанией запущен облачный OCR сервис предоставляющий возможность распознавания документов независимо от используемой операционной платформы. Данный продукт распространяется по коммерческой лицензии.
- CuneiForm, Cognitive OpenOCR – Программный продукт разработанный компанией Cognitive Technologies как коммерческий продукт. В настоящее время распространяется как система с открытым исходным кодом. Интерфейсом взаимодействия с пользователями является командная строка.
- Tesseract – программный продукт, изначально разработанный компанией Hewlett-Packard, а затем купленный Google. Имеет открытый исходный код. Как и в случае с OpenOCR, для взаимодействия с пользователем используется командная строка. Однако сторонними разработчиками выпущено большое количество графических оболочек для данного продукта: YAGF, OCRFeeder, gImageReader.

В последнее время стали популярны онлайн сервисы, предоставляющие услуги распознавания текста. Основным фактором появления таких систем стало открытие исходных кодов CuneiForm и Tesseract. Данные сервисы в большинстве случаев базируются на этих двух системах, предоставляя удобный веб-интерфейс для взаимодействия с пользователем, а так же в силу своей веб-ориентированности, снимают ограничения на используемые операционные платформы.

Точное распознавание символов в печатном тексте в настоящее время возможно, только если доступны четкие, не размытые и не зашумленные изображения. Точность при такой постановке задачи превышает 99%, абсолютная точность может быть достигнута только путем последующего редактирования человеком. Для изображений худшего качества применяют предварительную обработку изображений и обучение системы. При чем практика показывает, что предварительная обработка дает больший эффект чем обучение.

Рассмотренные онлайн системы распознавания текста в большинстве своем не имеют отдельных модулей обработки изображений, и не всегда предоставляют необходимый уровень точности распознавания, поэтому было решено разработать модуль препарирования изображений для повышения точности распознавания документов.

В рассматриваемой работе предлагается создать онлайн сервис позволяющий загружать изображения сканированных документов, а так же документы в формате *.pdf; выполнять препарирование для их дальнейшего распознавания.

Препарирование изображения включает:

1. Бинаризацию;
2. Выделение границ;
3. Преобразование к заданному количеству оттенков серого.

Для подобной обработки изображения будем использовать следующие методы:

1. Бинаризация изображения с адаптивным выбором порога;
2. Использование преобразования Собеля и Робертса;
3. Использование метода k-средних и медианного метода для преобразования палитры.

В перспективе ожидается создание веб-ориентированной OCR-системы на основе свободно-распространяемых продуктов Tesseract и CuneiForm с использованием разработанного модуля препарирования изображений.

Список литературы

1. Федоров А. Бинаризация черно-белых изображений: состояние и перспективы развития / А. Федоров – 2000. – Режим доступа: <http://it-claim.ru/Library/Books/ITS/wwwbook/ist4b/its4/fyodorov.htm>
2. Saulova J. Adaptive document image binarization / J.Saulova, M. Pietikäinen – 1999. – 12 с.
3. Андреева А.Ю. Компьютерная графика: учебное пособие./ Андреева А.Ю., Бубнова Н.Д. - Барнаул: Изд-во АлтГТУ, 2008. – 72 с.

РАЗРАБОТКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ДЛЯ ИССЛЕДОВАНИЯ МЕТОДОВ ПРЕДВАРИТЕЛЬНОЙ СЕГМЕНТАЦИИ СИМВОЛОВ ТЕКСТА

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

На сегодняшний день очень востребована технология оптического распознавания текста. Появление сканеров позволяет быстро получать изображение рисунков и текста. Однако при сколь угодно точной передаче изображения букв сканер не позволяет передать полученное изображение в текстовый процессор для того, чтобы его редактировать. Для этого текст необходимо распознать, то есть сравнить каждый отсканированный символ с шаблоном и таким образом «прочитать» текст.

Одна из наиболее существенных проблем в OCR - сегментация отдельных символов. Сложность сегментации символов происходит из-за разнообразных шрифтов, быстро расширяемых стилей текста, изображений и характеристик, таких как плохое качество печати или не качественное изображение. Размытые, нечеткие, слипшиеся и разбитые буквы являются основными причинами ошибок сегментации. Существуют шрифты, у которых, несмотря на общее хорошее качество печати, встречаются склейки как, например, в шрифте Tense. Более того, когда документ состоит из нескольких шрифтов, то более трудно выделять символы из-за различий в размерах и характерных особенностях типов каждого шрифта. Анализ изменения интенсивности изображения позволяет определить границы символа. Такие особенности, полученные из изображения, помогают эффективно провести сегментацию и распознавание текста. В данной работе будет реализована сегментация текста.

Методы сегментации можно условно разделить на следующие категории: прямой метод сегментации и распознавание, основанное на сегментации.

В первой категории, каждое слово сегментируется и далее к каждому символу слова применяется метод распознавания. Несмотря на простоту в реализации этого метода, его точность зависит от высокой точности точек сегментации. Тем не менее, такие точные способы сегментации еще не доступны. Следовательно, сегментацию слов и распознавание символов необходимо совмещать.

Во второй категории, потенциальные точки сегментации находятся на границах символов. Предполагаемые точки сегментации подтверждаются с помощью результатов распознавания. Этот метод является более разумным, чем первый, но зависит от выполнения распознавателя.

В рассматриваемом методе все действия проводятся с черно-белым изображением. Для указанного изображения проекция контуров может быть получена путем подсчета числа черных пикселей в столбце или строке. Но в таком изображении искомая проекция не может быть получена путем простого подсчета черных пикселей. Поэтому проекция определяется следующим образом:

$g(x,y)$ имеет значение из следующего диапазона:

$$0 \leq g(x,y) \leq L - 1$$

где $g(x,y)$ интенсивность пикселя (x,y) в черно-белом изображении, L – это уровень интенсивности.

Пусть $H_x(g)$ и $H_y(g)$ это гистограммы столбца x и строки y с интенсивностью g соответственно. Вертикальную проекцию контура, $P(x)$ можно определить следующим образом:

$$P(x) = \sum_{g=0}^{L-1} H_x(g) \cdot c(g), \quad 0 \leq c(g) \leq 1$$

где $c(g)$ коэффициент, зависящий от интенсивности и высоты изображения.

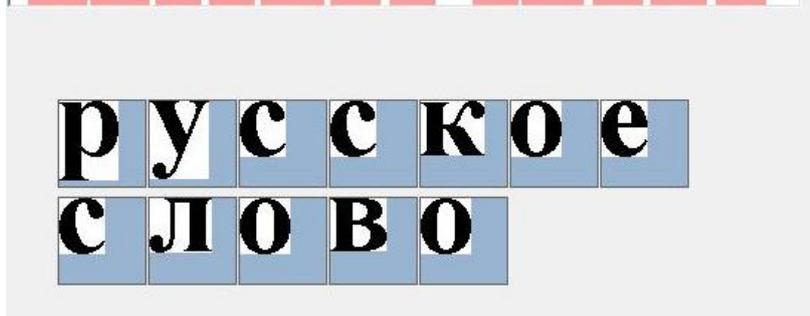
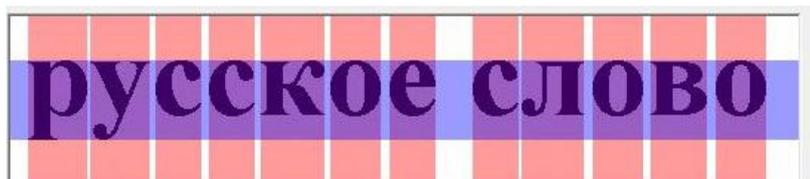
Подобным образом, определяется горизонтальный контур проекции $P(y)$:

$$P(y) = \sum_{g=0}^{L-1} H_y(g) \cdot c(g), \quad 0 \leq c(g) \leq 1.$$

Далее, с помощью посчитанной вертикальной проекции контуров $P(x)$ по определенным критериям выполняется поиск границ каждого символа. А затем для каждого отделенного символа, с помощью горизонтальной проекции $P(y)$ выполняется поиск верхней и нижней горизонтальной границы. Пример работы программы приведен на рисунке 1.

русское слово

(а)



(б)

Рисунок 1 – (а) Входящее изображение (строка слов), (б) Вверху выделенные вертикальные и горизонтальные границы символов, внизу конечный результат – отделена каждая буква.

Таким образом, данный метод показывает хорошие результаты. В дальнейшем предполагается применять его на реальных текстах и исследовать проблему разделения смежных символов.

Список литературы

1. Lee S. A new methodology for gray-scale character segmentation and recognition / S. Lee, D. Lee, H. Park – 1996. – 6 с.
2. Андреева А.Ю. Компьютерная графика: учебное пособие./ Андреева А.Ю., Бубнова Н.Д. - Барнаул: Изд-во АлтГТУ, 2008. – 72 с.

ТЕХНОРАБОЧИЙ ПРОЕКТ КОМПЛЕКСА ПРИКЛАДНЫХ ЗАДАЧ «ОПЕРАТИВНОЕ УПРАВЛЕНИЕ ОСНОВНЫМ ПРОИЗВОДСТВОМ»(НА МАТЕРИАЛАХ КОМПАНИИ «ЗАПАДНО-СИБИРСКИЕ ЖЕЛЕЗНЫЕ ДОРОГИ МОТОРВАГОННОЕ ДЕПО СТ. АЛТАЙСКАЯ» ОАО РЖД)

Смирнова Е.Ю. – студент, Бубнова Н.Д. – ст. преподаватель
Алтайский государственный технический университет (г. Барнаул)

Результаты работы каждого конкретного производственного объединения (предприятия) в конечном итоге определяются работой цехов основного производства. Поэтому необходимым условием успешного функционирования данного производственного объединения (предприятия) является построение рациональной системы управления основным производством.

Технорабочий проект – это документация на автоматизированную систему по ГОСТ 34.003-90. Настоящий стандарт устанавливает термины и определения основных понятий в области автоматизированных систем, содержанием которых является переработка информации [4].

Для достижения наибольшей эффективности производства и повышения качества работы используют комплексную автоматизацию. Автоматизация позволяет повысить производительность труда, улучшить качество продукции, оптимизировать процессы управления, отстранить человека от большой нагрузки в производстве и многое другое.

Автоматизация, за исключением простейших случаев, требует комплексного, системного подхода к решению задачи. Процесс автоматизации представляет собой совокупность методических, языковых, технических и программных средств, позволяющих организовать работу пользователей в предметной области [2].

Процесс автоматизации деятельности предприятия включает такие этапы как: разработка технического задания, описание и проектирование бизнес-процессов, создание базы данных, кодирование пользовательского интерфейса, тестирование и отладка системы автоматизации. Дальнейшее развитие системы автоматизации предусматривает анализ и сопровождение системы. На этапе эксплуатации системы также может потребоваться произвести перепроектирование системы автоматизации в соответствии с изменениями бизнес - структуры предприятия.

Основная тенденция развития систем автоматизации идет в направлении создания автоматических систем, которые способны выполнять заданные функции или процедуры без участия человека. Роль человека заключается в подготовке исходных данных, выборе алгоритма(метода решения) и анализе полученных результатов.

Анализ деятельности предприятий позволяет классифицировать задачи, решаемые предприятием.

Автоматизированное рабочее место (АРМ) - это совокупность программных и технических средств, установленных на рабочем месте пользователя для автоматизации его

работы. Современные АРМ строятся в основном на базе ПЭВМ, установленных на рабочих местах служащих и имеющих широкий набор периферийных средств.[1]

К достоинствам автоматизации можно отнести:

- Сравнительно низкую стоимость;
- Невысокую потребляемую мощность;
- Относительную простоту обслуживания;
- Использование простых языков общения с ПК;
- Возможность подключения к ЛВС и к широкомасштабной вычислительной сети;
- Возможность пользования информационными ресурсами из БД;
- Повышение оперативности и качества информации;
- Освобождение персонала от рутинных работ;
- Сокращение численности служащих и т.д.

Главной задачей в дипломной работе является разработка АРМ мастера(начальника) электромашинного цеха в моторвагонном депо.

Функция цеха – ремонт ТР-3 тяговых электродвигателей ТЛ2К и НБ-418К электровозов постоянного и переменного тока.

Были рассмотрены должностные обязанности мастера, подлежащие автоматизации:

- Разработка перспективных и текущих планов ремонта основных производственных средств предприятия;
- Организация разработки и доведения до исполнителей плановых заданий и графиков ремонта, контроль за проведением ремонта и испытаний оборудования;
- Контроль соблюдения правил эксплуатации и технического обслуживания;
- Руководство работой по аттестации и рационализации рабочих мест, улучшению нормирования труда и материального стимулирования;
- Организация ведения учета и составление отчетности о ремонтно-производственной деятельности;
- Обеспечение технически правильного выполнения эксплуатации оборудования;
- Организация работы по повышению квалификации рабочих и служащих [3].

При организации работы электромашинного цеха, возникает необходимость в обработке большого количества информации: составление отчетов различного вида, обработка поступающих актов и т.п. Чтобы обеспечить быстрый и удобный способ обработки нужных данных, необходимо их систематизировать и автоматизировать доступ к ним. Для автоматизации этой функции было рассмотрено несколько алгоритмов и выбран более удобный, а именно «электронный архив».

Электронный архив, система электронного архива - система структурированного хранения электронных документов, обеспечивающая надежность хранения, конфиденциальность и разграничение прав доступа, отслеживание истории использования документа, быстрый и удобный поиск. Так как нашей задачей является разработка АРМ мастера цеха, а не только функции документооборота, были выбраны некоторые функциональные части «электронного архива»:

- Структурированное хранение электронных документов;
- Быстрый и удобный поиск.

Программный продукт реализован в среде визуального программирования Visual Studio 2008 на языке С# с использованием СУБД и поэтому все новые изменения, исправления, добавление новых функций можно произвести за короткий период времени. Программный продукт обладает графическим интерфейсом под Windows, который обеспечивает удобную работу с программой. Ввод информации реализован с помощью меню. Для каждого пункта меню в программе имеется вспомогательная информация. Результаты, полученные в ходе

работы программы, можно увидеть на экране, сохранить на магнитном диске или вывести на печать.

Автоматизация рабочего места необходима для успешного развития производства.

Список литературы

1. Автоматизированное рабочее место [Электронный ресурс]. – Режим доступа: <http://ru.wikipedia.org/wiki/>
2. Автоматизация [Электронный ресурс]. – Режим доступа: <http://ru.wikipedia.org/wiki/>
3. Должностная инструкция начальника (мастера) ремонтного цеха [Текст]. – Метод. указание.
4. Межгосударственный стандарт «Информационная технология Комплекс стандартов на автоматизированные системы АВТОМАТИЗИРОВАННЫЕ СИСТЕМЫ термины и определения» ГОСТ 34.003-09 [Текст]. – Введен 01.01.92 г.

РАЗРАБОТКА ТЕХНИЧЕСКОГО ПРОЕКТА «АВТОМАТИЗАЦИЯ УЧЕТА И АНАЛИЗА ЭЛЕКТРИЧЕСКОЙ ЭНЕРГИИ» (НА МАТЕРИАЛАХ ОАО «ВЭС»)

Юношев Я.Ю. – студент, Бубнова Н.Д. – ст. преподаватель
Алтайский государственный технический университет (г. Барнаул)

В настоящее время большинство сфер деятельности человека автоматизировано. Не является исключением и учет электроэнергии.

Самое трудное – построить единую систему, которая будет обслуживать все запросы сотрудников. Однако при проектировании таких систем очень часто встречаются однотипные элементы, при работе с которыми разработчикам систем приходится совершать большой объем повторяющейся рутинной работы, на которую уходит много времени. Сегодня существует множество решений, которые значительно упрощают разработку и реализацию бизнес-приложений, такие как:

- 1С:Предприятие 8;
- MicrosoftDynamics NAV;
- MicrosoftDynamics AX;
- SAP NetWeaver;
- Платформа БАРС;
- Ананас.

Однако не всем нужны такие мощные системы, как 1С:Предприятие 8, или MicrosoftDynamics NAV, а нужна более простая и «легкая» система для создания небольших бизнес-приложений. В нынешних условиях лицензионной чистоты все более актуальным становится создание несложной и недорогой системы.

При организации работы ОАО «ВЭС», возникла необходимость в обработке большого количества информации для составления отчетов различного вида, обработки данных о количестве потребленной и потерянной электроэнергии и т.п. Чтобы обеспечить быстрый и удобный способ обработки данных, необходимо их систематизировать и автоматизировать доступ к ним.

Входной информацией для системы являются показания счетчиков, адреса, паспортные данные, потери, напряжение, мощность, имя пользователя, пароль, тип подстанций, дата, контактная информация. На выходе надо получить перечень клиентов, показания счетчиков,

информацию о потерях в сетях с полным описанием их характеристик из базы данных, удовлетворяющих входным данным пользователя.

Для названной предметной области была построена и согласована со специалистами ОАО «ВЭС» концептуальная модель. Был разработан технический проект, в состав которого входит АРМ инженера-инспектора и АРМ инженера энергетика. Их использование позволит повысить производительность труда инженеров, а так же уменьшить время расчетов потерь электроэнергии за счёт выявления должников, неэффективности отдельных узлов подстанций и трансформаторов с помощью дополнительного анализа.

АРМ инженера энергетика дает возможность непосредственной работы с базой данных - первоначальный ввод информации, добавление, корректировка, удаление, поиск информации, формирование отчетов.

Учетом и анализом электроэнергии занимается отдел контроля, в основные задачи которого входит:

- Выполнение планов по потреблению электроэнергии.
- Выявление причин потерь.
- Выявление должников и потребителей, не соблюдающих правил договора потребления.
- Своевременная передача энергии по внутренним сетям в другие регионы.

Поставленные задачи должен выполнять инженер-инспектор и он же является основным пользователем разрабатываемого программного продукта.

Инженер-инспектор ОАО «ВЭС» в соответствии с планом работ, выполняет проверку счетчиков, установленных в домах, предприятиях, питающих трансформаторах и т.д. Т.е. занимается сбором информации и внесением полученной информации в реестр. Далее он проверяет соответствие потребленной потребителем энергии к выданной ему энергии, тем самым выявляет, имеют ли место потери энергии на закрепленном за инспектором участке. Если потери имеются, то их сравнивают с разрешенными потерями на данный месяц. При выявлении потерь на участке необходимо определить, являются ли они техническими (в связи с неисправностью оборудования), либо связаны с нарушением договоров с потребителями (незаконное потребление энергии, выведения из строя учетного прибора (счетчика)). После обнаружения причины потерь следует ее устранить разрешенным для этого способом.

Актуальность комплекса прикладных задач автоматизации учета и анализа электроэнергии состоит в сокращении времени на выявление причин потерь, повышении качества анализа потребления энергии, уменьшении числа ошибок при внесении данных. Также исчезает потребность в использовании ранее внедрённых, разрозненных программ для составления отчетов, ведении базы данных, ее хранения и т.д.

Программный продукт реализован в среде визуального программирования С# с использованием СУБД и предусматривает возможность дальнейшего развития системы.

Список литературы

1. Ананьев П.И. Базы данных. Часть 1: Учебное пособие/ П.И. Ананьев, М.А. Кайгородова - Барнаул: Изд-во АлтГТУ им. И.И. Ползунова, 2006. – 168 с.
2. Инструкция по расчету и анализу технологического расхода электрической энергии на передачу по электрическим сетям энергосистем и энергообъединений. И 34-70-030-87. – М.: СПО "Союзтехэнерго", 1987.
3. Воротницкий В.Э., Калинин М.А. Расчет, нормирование и снижение потерь электроэнергии в электрических сетях. / Учебно-методическое пособие. – М.: ИПК госслужбы, 2000.

РЕАЛИЗАЦИЯ ИНТЕРАКТИВНОГО ОБЩЕНИЯ МЕЖДУ ПОЛЬЗОВАТЕЛЯМИ СОЦИАЛЬНОЙ СЕТИ ВУЗА ВКЛЮЧАЯ ВИДЕО-СЕМИНАРЫ

Устинов А.В. – студент, Старолетов С.М. – к.ф.-м.н., ст. преподаватель
Алтайский государственный технический университет (г. Барнаул)

1. Актуальность

Дистанционное образование с каждым годом набирает все большие обороты. Крупные учебные заведения, такие как Stanford и MIT, выкладывают бесплатно курсы по многим дисциплинам, адаптируя их под использование современных технологий. При этом они построены на объяснении преподавателем материала и обратной связью от пользователей. Миллионы людей принимают участие в их изучении. Люди заинтересованы в возможности дистанционного получения образования. А преподаватели в доступной возможности преподнести информацию.

В рамках данной работы реализовывалось приложения для проведения интерактивных консультаций с использованием вспомогательных инструментов и возможностью расширения функционала.

2. Новизна

В настоящий момент на рынке присутствует некоторое количество систем для проведения интерактивных консультаций. Как пример: OpenMeetings, BigBlueButton, сервис OnWebinar и другие. Минусом этих систем является либо невозможность использования без установки специального программного обеспечения, либо скудные возможности для интеграции в собственные разрабатываемые системы. Одной из основных целей является создание системы на основе открытых технологий, доступным на большом количестве устройств. Из-за этого выбран стек технологий HTML5, что позволит запускать приложение даже на мобильных устройствах.

Также разрабатываемая система для проведения интерактивных консультаций обладает возможностью расширения своего функционала за счет добавления новых приложений, встраиваемых через стандартный, предоставляемый системой API.

В последние годы бурным темпом идет развитие интернет-технологий, появляются новые возможности представления информации в лице HTML5, передачи информации при помощи WebSockets. Помимо статичных веб-страниц, генерируемых на стороне сервера, появилась возможность использования веб-приложений, которые представляют из себя скорее настоящие приложения, чем веб-страницы. Это достигается за счет использования событийно-ориентированных фреймворков, таких как Node.js или Twisted.

3. Постановка задачи

Вебинар — разновидность веб-конференции, проведение онлайн-встреч или презентаций через Интернет в режиме реального времени. Во время веб-конференции каждый из участников находится у своего компьютера, а связь между ними поддерживается через Интернет посредством загружаемого приложения, установленного на компьютере каждого участника, или через веб-приложение. В последнем случае, чтобы присоединиться к конференции, нужно просто ввести URL (адрес сайта) [1].

Вебинары могут проводиться с использованием различного рода инструментов. Это может быть не только видео-аудио, но и текстовый чат, совместное использование каких-либо графических материалов, совместное редактирование документов.

Вебинар представляет собой набор приложений, необходимых для его проведения. Приложения имеют доступ к данным о пользователях, проводящих и просматривающих вебинар, могут получить подробную информацию о них с помощью API социальной сети. Приложения могут быть различного характера. Начиная от основных: окно для видеоконференции, текстовый чат, заканчивая вспомогательными и даже специфичными для тематики вебинара: доска для рисования, приложение для показа слайдов, дополнительное окно для видеоконференции (например, для другой камеры), приложение для построения графиков и т.д. Основная идея в том, чтобы предоставить API для приложений, которое они могли бы использовать. И позволить загружать новые приложения. Таким образом, не меняя самой системы, расширять ее функционал.

Приложения имеют механизм сохранения своего состояния и отображения загруженного состояния. Этот механизм необходим для сохранения вебинаров и дальнейшего их просмотра. Сохраненные состояния приложения могут привязываться к времени. Таким образом, при воспроизведении вебинара присутствует объект таймера – отсчета времени, который уведомляет приложения о текущем времени воспроизведения, текущем состоянии (пауза, остановка, воспроизведение, переход на конкретное время), а приложения вебинара в зависимости от этих показателей извлекают из базы данных требуемые значения и воспроизводят свое состояние на экране.

Вебинар может подразумевать ведение его сразу несколькими людьми. Таким образом, поддерживается возможность поддержки нескольких камер, вещания нескольких потоков и объединения их.

Также, даже во время проведения вебинара, человек, проводящий его, должен иметь возможность вернуться к какому-то состоянию системы, в которой она находилась некоторое время назад. Это могут быть какие-то записи на доске для рисования, сделанные полчаса назад.

Человек, который ведет вебинар, управляет текущим состоянием приложений, используемых в текущем вебинаре. Когда он выполняет какое-то действие в одном из них, это действие выполняется для всех участников. Однако помимо общедоступных действий, также можно выполнять и действия для определенного круга лиц. Например, личные приватные сообщения.

4. Результаты

Результатом работы является система, состоящая из составных частей, написанных на:

- ActionScript 3.0 – приложения вебинара
- JavaScript – приложения вебинара
- Серверная часть в лице сервера Erlyvideo, используемого для передачи и записи видео и аудио
- PHP-фреймворк Symfony – интеграция с социальной сетью вуза
- Node.js – поддержка функционирования приложения вебинаров и обеспечения сохранения текущего состояния и доступа к каким-либо данным текущего сеанса

Система предоставляет все необходимые средства для интерактивного взаимодействия между пользователями социальной учебной сети.

Список литературы

1. Онлайн-семинар [Электронный ресурс]: материалы сайта – Режим доступа: <http://ru.wikipedia.org>.

РАЗРАБОТКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ДЛЯ РЕАЛИЗАЦИИ ИНТЕРВАЛЬНЫХ ВЫЧИСЛЕНИЙ

Поповцев А.В. – студент, Сорокин А.В. – к.т.н., доцент
Алтайский государственный технический университет (г. Барнаул)

В настоящее время одним из современных подходов в изучении математических моделей, параметры которых не являются обычными числами, а представляют собой некоторые интервалы, является подход, основанный на интервальных вычислениях. Такого рода подход позволяет расширять область исследуемых моделей, приводя к ним, например, нестационарные модели, параметры которых изменяются во времени, но остаются в рамках определенных интервалов. Такое преобразование с одной стороны удобно, т.к. расширяет возможности данного подхода, но с другой может представлять математические сложности, поскольку оперировать приходится не с числами, а с множествами (интервалами). Вследствие интереса со стороны математиков к этой проблеме подход, основанный на представлении параметров как интервалов, гармонично развился в одну из достаточно известных математических дисциплин, называемых «интервальным анализом». Главной идеей интервального анализа является представление значений числовых величин в виде интервала, внутри которого может изменяться истинное значение рассматриваемой величины. При этом сами интервалы рассматриваются как самостоятельные целостные объекты, над которыми определяются арифметические, аналитические и т. п. операции и отношения [1,2].

Обобщение вещественных чисел в виде интервалов приводит ко многим трудностям при применении традиционных численных методов для случая интервальных чисел. Из-за этого в интервальном анализе многие численные методы были пересмотрены и разработаны новые специализированные алгоритмы.

Реализаций алгоритмов интервального анализа не так много, как для традиционных численных методов, более того, чаще всего встречаются реализации для конкретных специфичных задач. Возникает потребность в инструменте, объединяющем реализации различных алгоритмов интервального анализа в рамках одной удобной оболочки.

Целью данной работы является разработка программного продукта, объединяющего в себе средства, позволяющие вести работу с интервальными числами, интервальными матрицами, решать некоторые задачи, связанные с различными приложениями интервального анализа, в первую очередь к теории автоматического управления. Программа должна предоставлять удобные средства для ввода значений, являющихся в общем случае интервальными матрицами, и последующей работы с ними.

Перечислим далее основные функциональные возможности, которые планируется реализовать в разрабатываемом программном продукте:

1. Считывание данных из файла и запись данных в файл. Пользователь должен иметь возможность сохранять результаты своих действий в рамках одного сеанса работы с программой для продолжения работы в рамках следующего сеанса.
2. Возможность выполнять некоторые команды с использованием арифметических операций и вызовов встроенных функций. Должен быть реализован интерпретатор команд, основными из которых будут арифметические выражения. Также выполнение команд должно быть одним из способов ввода данных пользователем.
3. Реализация некоторых основных операций над интервалами и интервальными матрицами. Например, вычисление метрик, нахождение определителя матрицы.
4. Решение интервальных систем линейных алгебраических уравнений. Должно быть реализовано несколько методов решения таких систем, выбор метода осуществляется пользователем.

5. Реализация метода Лемурье и метода главных миноров для построения интервального характеристического полинома интервальной матрицы.
6. Исследование устойчивости интервального характеристического полинома интервальной матрицы с использованием критерия Харитоновна.
7. Решение матричных уравнений Сильвестра и Ляпунова в интервальном случае.
8. Решение уравнения межотраслевого баланса (модель Леонтьева).
9. Решение задачи модального синтеза системы автоматического управления.

Список литературы

1. Алефельд Г. Введение в интервальные вычисления [Текст] / Г. Алефельд, Ю. Херцбергер. – М.: Мир, 1987. – 360 с.
2. Шарый С.П. Конечномерный интервальный анализ [Текст] / С. П. Шарый. – XYZ, 2010. – 591 с.

РАЗРАБОТКА СИСТЕМЫ АВТОМАТИЗИРОВАННОЙ ПОДГОТОВКИ И РАЗМЕЩЕНИЯ СБОРНИКОВ НАУЧНЫХ КОНФЕРЕНЦИЙ НА WEB-САЙТЕ АЛТГТУ

Горяев С.Л. – студент, Егорова Е.В. – к.т.н., доцент
Алтайский государственный технический университет (г. Барнаул)

Всё большее количество российских вузов проникается идеями проекта Webometrics, из которых наиболее привлекательной для России является обмен научной информацией за счет публикаций он-лайн.

Рейтинг Webometrics создает исследовательская группа Cybermetrics Lab при крупнейшем в Испании исследовательском центре Consejo Superior de Investigaciones Cientificas (Высший совет по научным исследованиям). Рейтинг построен на основе таких показателей, как веб-размер сайта университета (количество страниц, обнаруживаемых поисковыми машинами), его заметность (количество внешних ссылок на сайт), количество загруженных файлов (pdf, doc, ppt и ps), количество публикаций и индекс цитируемости по оценке Google Scholar.

Одна из основных идей рейтинга Webometrics (Webometrics Ranking of World Universities) – стимулирование обмена информацией между учеными мира за счет публикации результатов исследований вузов на вузовских сайтах. Анализ результатов по отдельным индикаторам и прежде всего по индикатору Sc (Scholar), характеризующему объем научных исследований, позволяет получить важную составляющую в общей картине интеллектуального развития регионов. Это особенно важно в современных условиях, когда обмен научной информацией между регионами приобретает первостепенное значение.

В связи с этим, на сайте вуза должно размещаться много сборников научных трудов. Сборники научных конференций состоят из секций, включающих в себя несколько разделов. Для удобства размещения и использования сборника его необходимо разбить на количество файлов равное количеству разделов с соответствующим содержанием в каждом файле. Ручная обработка такого документа требует много рабочего времени сотрудника, не говоря уже о нескольких сборниках сразу. Следовательно, необходима автоматизация этого процесса. Также при таком размещении сборника, в виде нескольких файлов, пользователь сможет загружать лишь необходимый раздел из интересующей его секции, а не целиком весь сборник, что позволит значительно снизить нагрузку на сервер.

Разрабатываемый программный продукт должен объединять в себе следующие функциональные возможности:

1. Принимать на вход pdf-файл(ы), на выходе выдавать множество pdf-документов соответствующих содержанием главам каждого файла;
2. Извлечение содержимого из pdf-документа, используя один из шаблонов, учитывающих сложную структуру документа;
3. Поиск в полученных данных оглавления документа;
4. Разбиение исходного документа на части в соответствии с результатом обработки оглавления документа.

Основной функционал программы, извлечение содержимого и разбиение документа на части, реализуется с помощью предназначенных для работы с pdf java-библиотек PDFBox и iText, а также php-библиотеки Zend_Pdf

Список литературы

1. Java PDF Library [Электронный ресурс]. – Режим доступа: <http://pdfbox.apache.org/>.
2. PHP Cross Reference [Электронный ресурс]. – Режим доступа: <http://phpcrossref.com/zendframework/library/Zend/Pdf/Parser.php.html/>.
3. iText Extract Page Content [Электронный ресурс]. – Режим доступа: <http://itextpdf.com/examples/iaa.php?id=277>.

ПОДХОДЫ К РАЗРАБОТКЕ ИЗОМЕТРИЧЕСКОГО ДВИЖКА ДЛЯ СОЗДАНИЯ ИНТЕРАКТИВНЫХ ПРИЛОЖЕНИЙ

Лелетко В.В. – студент

Алтайский государственный технический университет (г. Барнаул)

Игровой движок – это программа, которая является движущим звеном любой компьютерной игры. Это именно тот механизм, благодаря которому осуществляется весь необходимый функционал интерактивного компьютерного медиа контента.

Для первых игр писался собственный движок, направленный под конкретную игру. С ростом мощностей компьютеров, появилась возможность делать движки, незначительные изменения в которых могли обеспечивать необходимый функционал для другой игры, что значительно ускорило разработку и позволило делать больший акцент на медиа контенте, а не на программной части.

Некоторые современные движки позволяют создавать интерактивные приложения даже без знания языков программирования. Тем не менее, как правило, даже если изменения в программной части не нужны, при создании игры используются скриптовые языки для реализации игровой логики. Через них, например, регулируется поведение неигровых персонажей (NPC), игровых событий, реализуется интерфейс приложения и многие другие аспекты, динамически регулируемые по ходу разработки.

Также, существуют, как и универсальные движки, на которых можно создать игру практически любого жанра, так и привязанные к конкретному жанру или даже сеттингу. Например, Infinity Engine (BioWare) использовался лишь для создания CRPG, использующих ролевою систему D&D (Dungeon & Dragons), а на движке Source (Valve) создаются игры различных жанров.

Из ближайших аналогов будущего проекта стоит отметить движки Arcanum (Troika Games) и Fallout (Interplay). Их минусами является то, что они создавались под конкретную

игру, что само собой было не универсальным решением. Отчасти универсальным движком можно назвать уже упомянутый Infinity Engine (BioWare), но, как я сказал, он был ограничен ролевой системой D&D, к тому же имел значительный недостаток – любое здание в игре являлось отдельной локацией, что приводило к большому количеству лишних загрузок.

Часть движков создаются для того, чтобы программист мог взять его за основу и адаптировать под игру, часть же предназначены в первую очередь для того, чтобы лишь замена файлов ресурсов (скрипты, графика, звуки) привела к созданию новой игры. Моя будущая разработка предназначена в первую очередь для второго варианта, но, возможно, для специфических задач будет необходимо сделать изменения в модуле игровой механики.

Как и многие другие приложения, игровой движок можно разбить на несколько частей, особенно, если он обладает некоторой универсальностью:

- обработчик скриптов;
- отрисовщик графики;
- обработчик событий;
- движок интерфейса;
- звуковой движок;
- модуль, соединяющий все компоненты вместе;
- модуль игровой механики (модель);
- физический движок.

Здесь описаны лишь типичные части игровых движков. Одни же игры могут иметь дополнительные части, в некоторых же движках несколько частей объединены в одну и т.д.

В данной работе не будет использоваться физический движок (при желании, необходимая функциональность может быть описана через скрипты), обработчик скриптов и соединяющий модуль будут объединены (фактически, все остальные модули будут надстройкой для скриптов), обработчик событий и интерфейс также будут соединены, так как их отдельная реализация не эффективна.

Таким образом, необходимо разработать следующие компоненты:

Обработчик скриптов. Позволяет реализовать игровую логику, задавать «правила игры», не учтенные в модели игры, значительно облегчает реализацию и настройку многих аспектов игры. Также, весь интерфейс создается с помощью скриптов.

Отрисовщик графики. Позволяет производить отрисовку игровой сцены по заданным параметрам, скрывает все низкоуровневые вызовы, что повышает стабильность системы и делает работу над остальными компонентами проще. Основное средство вывода в приложении.

Движок событий и интерфейса. Средство ввода в приложении. Позволяет создавать такие элементы интерфейса, как окна и кнопки на них. Обрабатывает события, происходящие с мыши и клавиатуры.

Звуковой движок. Воспроизводит звук, поддерживает позиционирование звука относительно пользователя.

Модель игры. Задаёт простейшие правила (например, правило передвижения объектов по игровому полю), а также обеспечивает поддержку многих функций, которые в дальнейшем могут пригодиться скриптовому (например, вычисление расстояния между двумя объектами).

Основным функционалом разработанного комплекса являются:

- спрайтовая графика (с возможностью дальнейшей замены на 3d);
- работа с игровым полем с произвольным размером ячейки;
- масштабирование игрового поля;
- функционал для удобной работы с объектами на поле;
- управление интерфейсом через скрипты;
- динамическое освещение;

- пространственный звук;
- кросс-платформенность.

Фактически, разрабатываемый продукт является средством, благодаря которому будут осуществлены другие задачи. Сам по себе движок лишь создает возможности. То есть, движок – это лишь промежуточное звено в исследованиях. Конечные цели:

- создание гибкого и минималистичного языка программирования для определенных задач;
- разработка и тестирование интерфейсов;
- быстрое прототипирование медийных приложений.

Архитектура движка, изображенная на рис.1, очень похожа на концепцию MVC (Model-View-Controller). Центральным звеном всей системы является скрипт-движок. Средства ввода, средства вывода и механика четко разделены.

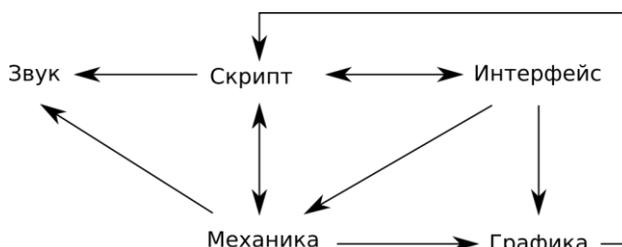


Рисунок 1 – Архитектура движка

Взаимодействие между скриптами и другими частями движка осуществляется через привязку скриптов к функциям модулей.

Основным правилами при разработке приложения являются:

Простая архитектура. Для повышения дальнейшей расширяемости программы.

Вытесняющая модульность. Абсолютно все компоненты движка разрабатываются независимо, по заданной спецификации, а затем, в соответствии с данной спецификацией соединяются вместе. Такой подход позволяет максимально просто производить замену модулей (если, например, предыдущая версия модуля оказалась нестабильной или неэффективной).

Гибкость хороша до тех пор, пока это не усложняет архитектуру. Если какая-либо функция позволяет сделать движок гибче, но реализация его основного предназначения становится сложнее, то эту функцию добавлять не надо.

Метод оболочек для модулей. Ни один модуль не взаимодействует с другим напрямую, лишь через оболочку, которая скрывает внутренние вызовы, напрямую вытекает из вытесняющей модульности.

Предполагается использование следующих технологий:

Язык: C++ (без STL). C++ обеспечивает хорошую скорость работы приложения, что очень важно для графики реального времени. Использование ООП подхода позволяет ускорить разработку и обеспечить модульность.

Среда разработки: CodeBlocks. Достоинствами для данного проекта являются минималистичность среды, ее кросс-платформенность, поддержка языка C++ и компилятора gcc.

Компилятор: gcc. Имеет большое количество параметров компиляции, что позволяет сделать приложение еще быстрее. Также, важным достоинством gcc является кросс-платформенность и эффективный механизм нахождения ошибок.

Библиотека SDL является удобной прослойкой между ОС и движком, что позволяет создавать кросс-платформенные медийные приложения без значительных затрат. Также

важным достоинством является большое количество необходимых расширений, которые будут использованы при разработке (SDL_image, SDL_ttf, SDL_mixer, SDL_net).

Библиотека **OpenGL** обеспечивает большие возможности по созданию трехмерной и двумерной графики при этом является кросс-платформенной.

Библиотека **Audiere** (либо **SDL_mixer**). Используется для звукового движка. К достоинствам Audiere можно отнести простоту библиотеки, к достоинствам SDL_mixer – встроенную поддержку пространственного звука.

РАЗРАБОТКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ДЛЯ РЕАЛИЗАЦИИ СИСТЕМЫ ТЕСТИРОВАНИЯ СТУДЕНТОВ

Селезнев К.К. – студент, Сорокин А.В. – к.т.н., доцент
Алтайский государственный технический университет (г. Барнаул)

Введение

На сегодняшний день современные информационные технологии предоставляют широкие возможности для качественного развития процесса образования. Прогресс технологий дает возможность реализовывать мировые тенденции в образовании, таким образом позволяя выйти в единое мировое информационное пространство.

Результатами внедрения информационных технологий в образовательный процесс могут послужить возможность реализации принципов дифференциального и индивидуального подхода к обучению, которые можно использовать как для очного, так и для дистанционного видов образования, а так же оптимизация процесса обучения за счет повышения эффективности самостоятельной работы студентов и совершенствования системы контроля качества знаний.

Контроль и учет знаний студентов является неотъемлемой частью образовательного процесса. Контроль в педагогическом труде – это система наблюдения и проверки соответствия педагогического труда, выявление достигнутых успехов студентов, допущенных ошибок и отклонений. Основными способами проверки и оценки знаний являются текущая проверка и оценка знаний, проводимая в ходе повседневных учебных занятий или периодическая оценка знаний в виде контрольных работ, тестов или экзаменов.

Тестирование студентов с целью контроля и учета знаний играет важную роль в учебном процессе. Тест – это совокупность заданий, ориентированных на определение уровня усвоения определенных аспектов содержания обучения, которые предполагают предоставление студенту или преподавателю возможности анализа, и, возможно, исправление своих ошибок. Для теста важнейшими критериями качества будут являться:

- Действенность теста, выражающаяся полнотой, всесторонностью проверки, пропорциональностью представления всех элементов изучаемых знаний и умений. Неотъемлемым условием действенности теста является четкая и ясная постановка вопроса в пределах освоенных знаний.
- Надежность, характеризующаяся стабильностью, устойчивостью показателей при повторных измерениях с помощью одного и того же теста или его равноценного заменителя. Количественно этот критерий характеризуется вероятностью достижения запрогнозированных результатов. Надежность теста повышается при увеличении количества тестовых заданий.
- Дифференцированность – способность теста отделить тех, кто усвоил материал на необходимом уровне, от тех, кто заданного уровня не достиг.

Помимо функции проверки и оценки знаний тестирование так же можно рассматривать как элемент обучения. Если после очередной попытки прохождения теста студенту выводить не только итоговую оценку, но и все варианты ответов студента, разграниченные на правильные и неправильные, то у него появится возможность обдумать, где он ошибся, проанализировать причины своих ошибок и уяснить, почему тот или иной ответ был правильным.

Таким образом, тестирование в образовательном процессе может быть использовано не только как элемент контроля и оценки усвоенных знаний, но и как элемент обучения, позволяющий студентам и преподавателям более детально анализировать уровень полученных знаний и предпринимать попытки восполнения пробелов в знаниях, что только увеличивает важность использования тестирования.

Обзор существующих систем тестирования

Из всего существующего на сегодняшний день программного обеспечения, позволяющего проводить тестирование студентов, можно выделить два вида систем: специализирующиеся только на реализации тестирования системы и всеобъемлющие программные учебные комплексы.

Среди узкоспециализированных систем самыми примечательными являются системы тестирования SunRav TestOfficePro и VeralTest. Обе системы представляют собой пакет программ, предоставляющих возможность создания, редактирования тестов и администрирования и средства запуска тестирования. Системы значительно отличаются друг от друга лишь способами установки и настройки, а так же методами работы в локальной сети. Если SunRav TestOfficePro требует наличие компьютера-сервера, а так же установку клиентских программ на каждый компьютер, участвующий в проведении тестирования, то в VeralTest достаточно запуска специальной программы-сервера тестирования, после чего на каждом компьютере, входящих в одну сеть с сервером, становится доступным тестирование посредством web-интерфейса. Данные системы являются коммерческими, и требуют обязательную покупку лицензии на использование в образовательных учреждениях.

Не менее примечательной является система управления обучением Moodle (Modular Object-Oriented Dynamic Learning System), представляющая собой web-приложение, с помощью которого предоставляется возможность создания сайтов для обучения в режиме «онлайн». Moodle – это программный комплекс, включающий в себя огромное множество учебных модулей, в том числе и модуль тестирования. Moodle свободно распространяется под лицензией GNU GPL. Для установки данной системы управления обучением требуется компьютер-сервер с установленными на него HTTP-сервером с интерпретатором PHP-кода, а так же системой управления SQL-базами данных, необходимых для ее работоспособности.

Все рассмотренные системы имеют схожие принципы проведения тестирования: пять основных типов вопросов, возможность случайного перемешивания вопросов и вариантов ответов, схожие опции настроек теста и др. Лишь стоит отметить, что Moodle выгодно отличается от других систем расширенным набором настроек, что позволяет этой системе быть более гибкой, подходящей практически под любые требования.

Необходимость разработки

Основными характеристиками качественной системы тестирования являются:

- Простота установки, не требующей установки дополнительных программных и аппаратных средств.
- Удобный, интуитивно понятный интерфейс программ.

- Широкий набор опций, позволяющий настраивать систему под любые требования, в то же время не являющийся чересчур сложным для среднестатистического пользователя.
- Простота работы системы в пределах компьютерной сети. Например, использование специальной программы-сервера для проведения тестирования посредством web-интерфейса.
- Компактность программного пакета, обеспечивающая портативность системы.

Из приведенных в обзоре существующих систем тестирования практически всем данным характеристикам соответствует лишь пакет программ VeralTest, однако такой пакет распространяется платно, что является его существенным недостатком при использовании в образовательных учреждениях. Поэтому целесообразной будет разработка такой системы тестирования, которая будет свободно распространяться, соответствовать заданным характеристикам, и не иметь ничего лишнего из функционала, что может только усложнить как разработку, так и использование системы.

Постановка задачи

Данная разработка программного обеспечения для реализации тестирования студентов подразумевает разработку программного пакета, включающего в себя средства создания и редактирования тестов, запуска процесса тестирования и административной части, отвечающей за настройку системы, получение и обработку результатов тестирования. Основные функциональные возможности, планируемые к реализации в программном обеспечении:

- Наполнение базы вопросов – специального хранилища всех тестовых заданий, разделенных по категориям и тематикам.
- Создание и редактирование тестов в специальном редакторе как с нуля, так и путем импорта из различных популярных форматов.
- Запуск тестирования с помощью программы-сервера, позволяющей получить доступ к тесту с любого компьютера, входящего в одну с сервером компьютерную сеть, посредством web-интерфейса.
- Настройка вывода: последовательный или случайный порядок вопросов и вариантов ответов, временные ограничения в процессе тестирования, тип вывода результатов тестирования и другие параметры.
- Защита файлов системы шифрованием.
- Генерация общих и детальных отчетов о результатах тестирования по одному пользователю или группе пользователей.

Список литературы

1. Контроль и учет знаний в учебном процессе [Электронный ресурс]. – Режим доступа: http://pedsovet.org/component/option,com_mtree/task,viewlink/link_id,6087/itemid,118/.
2. SunRav TestOfficePro – компьютерное тестирование, аттестация, сертификация [Электронный ресурс]. – Режим доступа: <http://sunrav.ru/testofficepro/>.
3. VeralTest. Пакет программ для разработки тестов и проведения тестирования [Электронный ресурс]. – Режим доступа: <http://veralsoft.com/veraltest.shtml>.
4. Демонстрационный сайт Moodle [Электронный ресурс]. – Режим доступа: <http://moodle.net/>.

РАЗРАБОТКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ДЛЯ ФОРМИРОВАНИЯ СТАТИСТИЧЕСКОЙ ОТЧЕТНОСТИ ПО РЫНКУ ТРУДА КЫРГЫЗСКОЙ РЕСПУБЛИКИ

Норсеев А.В. – студент, Ананьев П.И. – доцент
Алтайский государственный технический университет (г. Барнаул)

Как показывает мировая практика, наличие качественных государственных информационных ресурсов является важнейшим элементом эффективного государственного управления. Особенно, это касается таких областей, как демография, трудовые ресурсы, миграционные процессы. Такие информационные ресурсы формируются путем постоянной регистрации населения и используются государством для решения различных социально-экономических проблем.

Создание автоматизированной информационной базы рынка труда в Кыргызской республике обусловлено необходимостью решения проблем, которые оказывают непосредственное влияние на социально-экономическое развитие республики.

Основанием для разработки автоматизированной информационной базы рынка труда являются закон Кыргызской республики «О содействии занятости населения» № 113 от 27.07.1998 года, закон Кыргызской республики «О внутренней миграции» № 133 от 30 июля 2002 года, трудовой кодекс Кыргызской республики № 106 от 4.08.2004 года, закон Кыргызской республики «О внешней миграции» № 61 от 17.07.2000 года, постановление правительства Кыргызской республики № 709 от 17.11. 2009 года «Положение министерства труда, занятости и миграции Кыргызской республики».

Автоматизация рынка труда Кыргызской республики должна осуществляться в интересах граждан и государства и строиться на основе данных регистрации субъектов рынка труда в государственных службах занятости.

На основе выше изложенного для автоматизации была разработана специальная компьютерная программа, которая обеспечивает автоматизацию следующих направлений деятельности:

- учет ищущих работу граждан и содействие их трудоустройству;
- оказание услуг по профессиональной ориентации;
- организация профессионального обучения безработных граждан;
- организация проведения оплачиваемых общественных работ;
- приказооборот;
- учет форм активной политики содействия занятости населению (временное трудоустройство граждан, испытывающих трудности в поиске работы, несовершеннолетних в возрасте от 14 до 18 лет, граждан в возрасте от 18 до 20 лет из числа выпускников учреждений начального и среднего профессионального образования, ищущих работу впервые);
- начисление и выплата пособий по безработице, стипендий и пр. выплат;
- взаимодействие с работодателями (ведение базы работодателей, учет посещений и консультаций работодателей, учет квотируемых рабочих мест, информация о заключенных договорах, вакансиях, учебных группах, учебно-производственной базе, сведениях о высвобождении работников и пр.);
- межтерриториальный обмен вакансиями;
- оказание информационно-справочных услуг (информационные киоски);
- формирование государственных статистических форм;
- анализ данных, формирование ведомственных, произвольных региональных статистических форм и др.

В ходе работы над автоматизированной системой рынка труда Кыргызской республики было реализовано следующее:

- Аутентификация работника службы занятости;
- Разграничение прав доступа: служит для того чтобы отдельное подразделение службы занятости имело доступ только к той информации с которой оно работает;
- Формирование статистических отчетов: является одной из главной функцией программного комплекса службы занятости;
- Синхронизация баз данных региональных центров служб занятости(на уровне сервера), все данные стекаются на главный сервер для формирования статистической отчетности.

Программный продукт имеет интерактивный графический интерфейс и защиту от ввода ошибок пользователя, что обеспечивает высокую отказоустойчивость системы.

Для разработки программы был использован язык программирования С#. Технология – .NETFramework 4.0, сервер базы данных – MSSQLServer 2008, сервер отчетов – MSSQLReporting Services 2008

Список литературы

1. Методические руководство по составлению статистической отчетности по рынку труда. – Бишкек. – 2010.
2. Официальный сайт Министерства труда, занятости и миграции Кыргызской республики [Электронный ресурс] – Режим доступа: www.mz.kg
3. Законодательство Кыргызской республики [Электронный ресурс] – Режим доступа: <http://www.kg.spinform.ru/>

НОВЫЙ ПОДХОД К ПРОГРАММИРОВАНИЮ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА

Вотяков Р.А. – студент, Старолетов С.М. – к.ф.-м.н., ст. преподаватель
Алтайский государственный технический университет (г. Барнаул)

Введение

На данный момент существует множество задач, с которыми человеческий мозг может справиться очень легко, но для вычислительной техники они представляют особую проблему. Данные задачи можно обозначить, как проблемы искусственного интеллекта (ИИ). Среди таких проблем можно выделить: распознавание образов, обработку естественного языка, разработку экспертных систем, разработку прогнозирующих систем и т.д. Проблемы ИИ представляют интерес во многих прикладных областях компьютерной науки.

Подходы к решению проблем искусственного интеллекта

На сегодняшний день существует множество подходов к решению проблем ИИ. Самым распространенным из данных подходов является — моделирование искусственной нейронной сети, которая в какой-то мере моделирует поведение человека. Возможно это не самый лучший подход. Ведь искусственные нейронные сети далеки от реального строения мозга человека. Они конечно имеют схожую структуру, то есть головной мозг тоже состоит из нейронов, но головной мозг имеет более сложное внутреннее строение. К тому же нейронные сети, как правило, пытаются моделировать поведение человека, а не работу головного мозга. Но ведь интеллект человека не ограничивается разумным поведением.

В данной работе предлагается рассмотреть подход, который основан на понимании строения головного мозга и его функций. Ведь чтобы моделировать человеческий разум, необходимо разобраться во внутренней структуре головного мозга.

Строение головного мозга

Головной мозг имеет множество отделов, но нас больше всего интересует кора головного мозга, так как именно она отвечает за осуществление интеллектуальной деятельности. Кора головного мозга состоит из нескольких полунезависимых зон, каждая из которых отвечает за свои функции. К примеру, кора состоит из различных зон, которые отвечают за зрительную информацию (V1, V2, V4, IT), зон, которые отвечают за слуховую информацию (A1), зон, которые отвечают за моторные функции (M1) и т. д. Но кора головного мозга очень однородна, следовательно можно предположить, что различные функциональные зоны коры работают по одному и тому же принципу (алгоритму).

Функциональная организация головного мозга имеет форму иерархии.

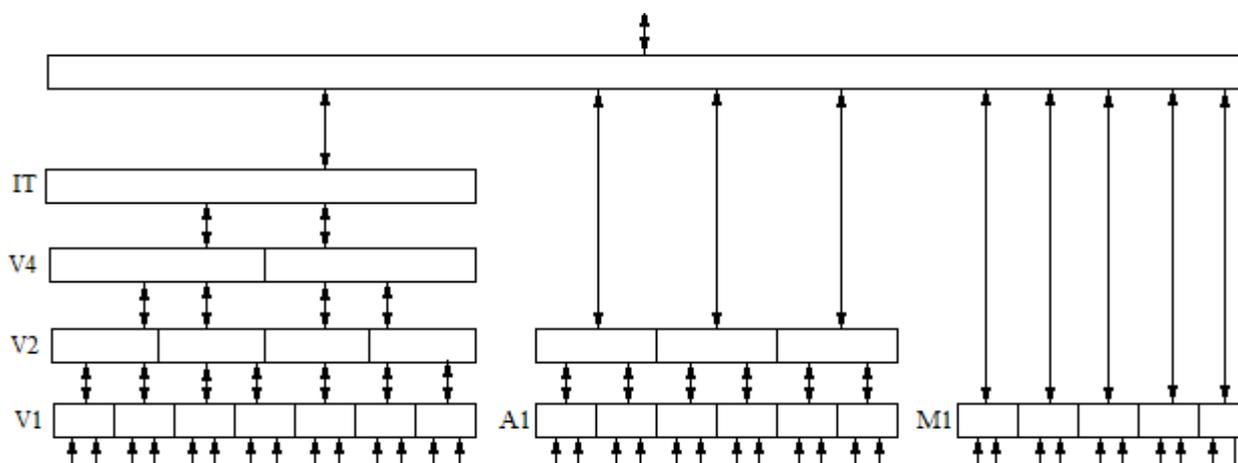


Рисунок 1 – Схематическое изображение иерархии функциональных зон

Низшие зоны иерархии (первичные сенсорные зоны) непосредственно получают информацию об окружающем мире в виде некоторого паттерна (последовательности сигналов). Они занимаются обработкой информации на самом простом, базовом уровне и затем обработанную информацию передают вверх по иерархии, и в конечном итоге информация попадает в ассоциативную память.

Чем выше находится зона в иерархии тем более сложными понятиями она оперирует. Например: зрительная зона V1 оперирует непосредственно визуальной информацией, а зона IT оперирует с известными человеку визуальными объектами реального мира (столом, книгой и т. д.). Получив и обработав последовательность сигналов, зона передает вверх по иерархии условное «имя» этой последовательности, которое тоже представляет собой последовательность сигналов. К примеру, если зона A1 получает последовательность звуков, то распознав ее, как часть известной мелодии, она передает название мелодии вверх по иерархии.

Информация в данную зону может попадать как из зоны иерархии, которая находится ниже данной, так и из зоны иерархии, которая находится выше. Если информация в зону попадает сверху, она преобразуется в более простое представление. То есть название известной мелодии, при переходе от уровня выше в иерархии к уровню ниже, преобразуется в последовательность звуков. Так же информация может попасть в зону через таламус. В частности, связь функциональных зон коры головного мозга с таламусом можно расценивать, как обратную связь с задержкой. То есть зона может передавать выходной сигнал в таламус и затем снова его получать на вход.

Каждая зона иерархии запоминает инвариантные репрезентации входных сигналов в колонках нейронов зоны. Инвариантная репрезентация — это внутренняя модель объектов в памяти человека, которая не изменяется даже в условиях информационной переменчивости.

За счет инвариантных репрезентаций и иерархии коры головного мозга мы как раз и можем классифицировать различные объекты реального мира, по тем представлениям, которые сохранились в нашей памяти.

Модель “память-предсказание”

Существует множество теорий по поводу того, как работает головной мозг. В данной работе предлагается рассмотреть модель «память-предсказание». Главная идея этой модели заключается в том, что все зоны головного мозга параллельно и непрерывно генерируют прогнозы. То есть на основании полученной части последовательности сигналов, зоны могут прогнозировать, какой сигнал будет дальше. Если функциональная зона принимает последовательность сигналов, которую может распознать, на основе сигналов она формирует «имя» последовательности, которое она поддерживает в неизменности. На протяжении всей последовательности сигналов зона непрерывно прогнозирует, какой сигнал будет дальше, если ее прогноз не сбывается (пришел сигнал, которого не ожидали), она передает последовательность сигналов на уровень выше. Если никакая из зон иерархии не может распознать последовательность, то данная последовательность запоминается в гиппокампе, из которого в дальнейшем может быть извлечена и сохранена в иерархии коры головного мозга.

Потоки информации в коре головного мозга могут двигаться в двух направлениях: вверх по иерархии и вниз. Вверх по иерархии движется информация полученная с сенсоров, вниз двигаются прогнозы от высших ассоциативных зон.

За счет возможности прогнозирования мы можем восстанавливать фрагменты каких то объектов по контексту.

Выводы

На основании иерархического строения коры головного мозга и модели «память-предсказание» можно реализовать структуру, которая нацелена на повторение структурных и алгоритмических свойств коры головного мозга.

Список литературы

1. Хокинс Дж., Блейкли С. Об интеллекте. Пер. с англ. – М.: Изд-во Вильямс, 2007. – 240 с.

ПАРАЛЛЕЛЬНЫЙ АВТОМАТ КАК СРЕДСТВО АНАЛИЗА НЕДЕТЕРМИНИРОВАННОГО ПОВЕДЕНИЯ ПАРАЛЛЕЛЬНЫХ СИСТЕМ

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

1. Отладка параллельного программного обеспечения

Использование многопоточных взаимодействий при создании промышленного программного обеспечения (ПО) становится широкодоступным. Снижение стоимости параллельных вычислений, увеличение количества ядер, асинхронная природа пользовательских интерфейсов вынуждают разработчиков использовать многопоточность. Однако параллельные системы намного сложнее, чем последовательное ПО. Порядок

многопоточных взаимодействий в параллельной системе заранее не задан и разработчик должен обеспечивать правильное поведение параллельной системы для всех возможных исполнений. Так как промышленное программное обеспечение часто подвержено изменениям, разработчик обязан обеспечивать требуемое поведение системы при каждом из них. Отладка параллельных программ на порядок сложнее отладки последовательных программ. Возрастают размеры и сложность параллельных систем, в результате стоимость производства параллельного ПО дороже стоимости производства последовательного. Это приводит к необходимости создания новых принципов разработки инструментов для анализа, отладки и тестирования параллельных алгоритмов. Инструменты автоматического анализа позволяют разработчику непрерывно обеспечивать требуемые свойства, тестировать, поддерживать высокое качество ПО.

2. Параллельный конечный автомат

Будем считать, что параллельная система выполняется на абстрактном вычислительном узле, удовлетворяющем вычислительной модели PRAM (параллельная машина с произвольным доступом к памяти). Мы рассматриваем вычислительную модель PRAM, т.к. она наиболее полно отражает возможности вычислительных устройств на сегодняшний день. Большинство промышленных языков программирования являются объектными, поэтому будем считать, что параллельная система реализована на объектном языке.

Пусть параллельная система $PSys$ есть взаимодействие между элементами множества потоков $T = \{mThread, t_1, t_2, \dots\}$, где t_i - уникальный идентификатор потока, и элементами множества объектов $O = \{program, o_1, o_2, \dots\}$, где o_i - уникальный идентификатор объекта. Каждый объект $o \in O$ относится к некоторому классу c из множества $Cls = \{System, c_1, c_2, \dots\}$. Будем считать, что в системе всегда существуют единственный объект $program$ класса $System$ и главный поток $mThread$. Класс $c \in Cls$ определяет множество операций $Ops(c)$, которые может выполнять поток $t \in T$ над объектом $o \in O$ класса c . Разработчик размечает некоторую семантически значимую точку кода для некоторого класса $c \in Cls$ операции $op \in Ops(c)$, метаинформацией о том, что данный фрагмент является некоторым состоянием модели $State_{op,c}$ где $State$ - уникальный идентификатор состояния для кортежа $\langle op, c \rangle$. Для каждого класса $c \in Cls$ и для операции $op \in Ops(c)$ определим начальное и конечное состояния, как $S_{op,c}$ и $F_{op,c}$ соответственно (в начале и в конце реализации операции op). Обозначим через S_{main} и F_{main} начальное и заключительное состояние объекта $program$ класса $System$. Мы будем строить автоматную модель, отражающую многопоточные взаимодействия параллельной системы, на основе данных состояний и переходов между ними по коду приложения.

3. Транспонирование матрицы как пример многопоточного взаимодействия

Рассмотрим работу параллельной системы на примере задачи о транспонировании матрицы. Входными данными для данного приложения являются пара (n, A) , где n - размерность матрицы A . Выходными данными является матрица $B = A^T$. Данная задача широко используется как операция в матричных вычислениях, поэтому представляет интерес для вычисления ее параллельно. Параллельная реализация алгоритма, основана на соображении, что операции транспонирования элементов можно выполнять независимо друг от друга в произвольном количестве потоков согласно принципу, изображённому на рис. 1.

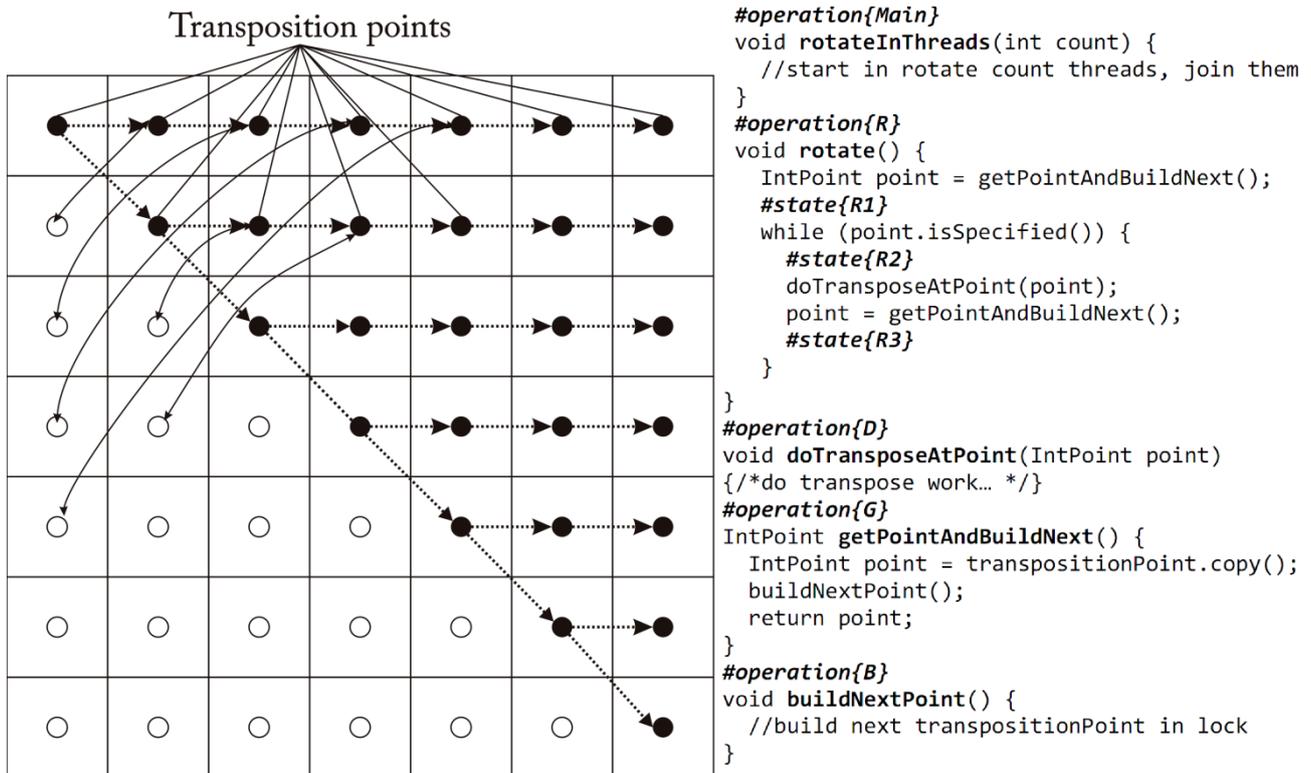


Рисунок 1 – Задача о транспонировании матрицы и параллельная реализация

С первого взгляда код параллельной реализации идентичен последовательной, т.к. вычисление следующей точки транспозиции *buildNextPoint()* происходит в критической секции и не может выполняться параллельно в нескольких потоках. Однако мы покажем, что при переходе от последовательной имплементации к параллельной программе произошло нарушение инварианта в параллельной системе, который в последовательной имплементации был достаточно очевиден. Данный факт мы установим с помощью применения параллельного автомата и ограничивающей системы неравенств.

4. Примитивы и взаимодействия

Формально опишем и определим примитивы, которые мы будем использовать для отражения состояний и взаимодействий автоматной модели. Во время работы многопоточной программы осуществляются следующие виды взаимодействий:

1. Синхронные взаимодействия – это последовательные вызовы функций, выполнения инструкций в коде программы (*call*-вызовы), которые происходят в одном потоке. После того, как программа выполнила *call*-вызов, и вызванная функция завершила свою работу, выполняется переход в точку возврата, которая находится в стеке программы.
2. Асинхронные взаимодействия - операции создания нового потока (*fork*-взаимодействие). После того, как программа создала новый поток, она продолжает выполняться последовательно по коду.

Выделим три вида переходов (взаимодействий) в нашей автоматной модели:

1. Последовательный переход (*call*-переход) - соответствует выполнению *call*-вызовов.
2. Асинхронный переход (*fork*-переход) - соответствует выполнению *fork*-взаимодействий.
3. Возвратный переход (*return*-переход) - соответствует событию возврата в некоторую точку, адрес которой находится в стеке программы.

Чтобы выразить в автоматной модели семантику взаимодействий реальных многопоточных программ введем понятия синхронной и асинхронной *call*-вершины. *Call*-вершина – это примитив автоматной модели, который имеет две составляющие: множество исполняемых переходов и возвратный переход. Множество исполняемых переходов содержит только последовательные или асинхронные переходы (для синхронной и асинхронной вершин соответственно). Автоматная модель, попадая в некоторую *call*-вершину c , должна выполнить все ее исполняемые переходы в некотором порядке. После этого она должна перейти в состояние, указанное возвратным переходом вершины c . В силу того, что *call*-вершины нужны только для отображения семантики многопоточных взаимодействий, они будут добавляться искусственно, без указания уникального идентификатора на рисунках. Будем изображать примитивы автоматной модели так, как показано на рис.2.

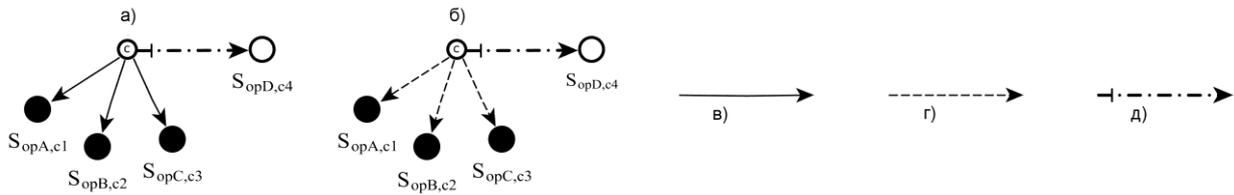


Рисунок 2 – Примитивы автоматной модели: а) - синхронная *call*-вершина, б) - асинхронная *call*-вершина, в) - последовательный переход, г) - асинхронный переход, д) - возвратный переход

Определим формально множества вершин для автомата и свяжем их соотношениями, на их основе введем определение параллельного автомата:

$$\begin{aligned}
 S^{all} &= \bigcup_{\substack{c \in Cls \\ op \in Ops(c)}} S_{op,c}^{all} - \text{множество состояний системы} \\
 S^{call} &= S^{call,sync} \cup S^{call,async} - \text{множество } call\text{-вершин} \\
 S^{call,sync} &= \bigcup_{\substack{c \in Cls \\ op \in Ops(c)}} S_{op,c}^{call,sync} - \text{множество синхронных } call\text{-вершин} \\
 S^{call,async} &= \bigcup_{\substack{c \in Cls \\ op \in Ops(c)}} S_{op,c}^{call,async} - \text{множество асинхронных } call\text{-вершин} \\
 S^{mark} &= \bigcup_{\substack{c \in Cls \\ op \in Ops(c)}} S_{op,c}^{mark} - \text{множество состояний, размеченных мета-информацией} \\
 S^{start} &= \bigcup_{\substack{c \in Cls \\ op \in Ops(c)}} S_{op,c} - \text{множество начальных состояний} \\
 S^{finish} &= \bigcup_{\substack{c \in Cls \\ op \in Ops(c)}} F_{op,c} - \text{множество конечных состояний} \\
 S_{op,c}^{all} &= S_{op,c} \cup F_{op,c} \cup S_{op,c}^{mark} \cup S_{op,c}^{call} - \text{множество состояний операции } \langle op, c \rangle \\
 S_{op,c}^{call} &= S_{op,c}^{call,sync} \cup S_{op,c}^{call,async} - \text{множество } call\text{-вершин для операции } \langle op, c \rangle
 \end{aligned} \tag{1}$$

Параллельным автоматом A назовем автомат вида:

$$A = (S^{all}, \delta, S_{main}, F = \{F_{main}\}, T = T_1 \cup T_2 \cup T_3),$$

где $S_{main}, F_{main} \in S^{all}, T_1, T_2, T_3$ – множество call-, fork-, return-переходов,
 δ – функция переходов :

$$\bigcup_{\substack{c \in Cls \\ op \in Ops(c)}} (S_{op,c}^{mark} \cup S_{op,c}) \times T_1 \rightarrow S_{op,c}^{all}, S^{call} \times T_1 \rightarrow S^{start}$$

$$S^{call, sync} \times T_2 \rightarrow S^{start}, \bigcup_{\substack{c \in Cls \\ op \in Ops(c)}} S_{op,c}^{call} \times T_3 \rightarrow S_{op,c}^{all}.$$

Построим пример параллельного автомата для задачи о транспонировании матрицы (рис. 3.), псевдокод которого представлен на рис. 2.

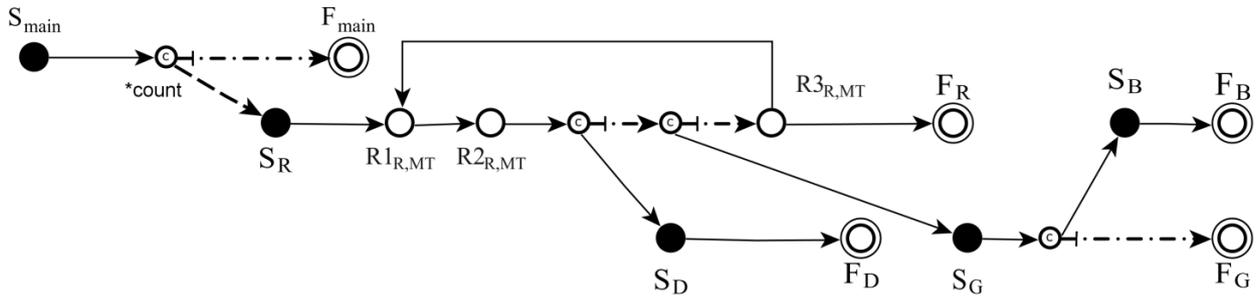


Рисунок 3 – Параллельный автомат для задачи о транспонировании матрицы

Параллельный автомат отражает в системе принципиальные моменты с точки зрения параллелизма. Например, мы можем увидеть, что параллельно может выполняться не только операция B (метод *buildNextPoint(...)*), но и операция G (метод *getPointAndBuildNext(...)*). Введем нерефлексивный частичный порядок $<_S$ на элементах множества S , который эквивалентен отношению “happened before” для распределенных систем [1,2]. Сделаем это следующим образом:

$$\forall S1, S2 \in S : S1 <_S S2 \Leftrightarrow \exists t \in T_1 \cup T_2, \text{ где } S1 \xrightarrow{t} S2. \quad (3)$$

Исполнение потоком (потоковым исполнением) $t \in T$ состояния $S_{op,c}$, где $c \in Cls, op \in Ops(c)$ будем обозначать как $S_{op,c}^t$. Исполнение потоком $mThread$ состояния $S_{op,c}$, будем обозначать как $S_{op,c}$. Исполнением $Exec(A)$ параллельного автомата A назовем множество всевозможных историй исполнения. Под историей исполнения [3] мы понимаем конечную последовательность $H = (h_1, h_2, \dots, h_n)$, где h_i – возможное исполнение потоком из множества T состояния автомата из множества S^{all} .

На множестве потоковых исполнений $S^T = S \times T$ введем нерефлексивный частичный порядок $<_{S^T}$ следующим образом:

$$S_{opA,c1} <_S S_{opB,c2} \Leftrightarrow \forall t \in T : S_{opA,c1}^t <_{S^T} S_{opB,c2}^t \quad (4)$$

Согласно (5), операции $\langle opA, c_1 \rangle$ и $\langle opB, c_2 \rangle$ немонопольные \Leftrightarrow существует $H \in Exec(A)$, обеспечивающая их параллельное исполнение [4].

$$\langle opA, c_1 \rangle \text{ и } \langle opB, c_2 \rangle - \text{немонопольны} \Leftrightarrow \exists(t_1, t_2 \in T) : \begin{cases} S_{opA, c_2}^{t_1} < S_{opB, c_2}^{t_2} \\ S_{opB, c_2}^{t_2} < F_{opA, c_2}^{t_1} \end{cases} \quad (5)$$

$$\langle opA, c_1 \rangle \text{ и } \langle opB, c_2 \rangle - \text{монопольны} \Leftrightarrow \exists(t_1, t_2 \in T) : F_{opA, c_2}^{t_1} < S_{opB, c_2}^{t_2}$$

Таким образом, мы имеем возможность задать систему ограничивающих условий (например, условие монопольного исполнения) в терминах потоковых исполнений. Отношение $<_{s^T}$ между потоковыми исполнениями получены на основе отношения $<_S$ параллельного автомата A , а так же заданы искусственно набором ограничивающих условий. Если получившаяся система неравенств будет совместна, то мы получим исполнение, которое недопустимо с точки зрения разработчика, так как данное решение обеспечивает выполнение ограничивающих условий без нарушения работы параллельной системы.

5. Анализ некорректного поведения

Работу параллельного автомата рассмотрим на примере многопоточной системы, псевдокод которой изображен на рис. 1. Пусть $count = 2$, тогда в системе взаимодействуют множество потоков $T = \{mThread, t_1, t_2\}$. Так как метод $buildNextPoint(...)$ исполняется в критической секции, то сформулируем условие монопольного исполнения (взаимоисключения) для операции $\langle B, MT \rangle$ (метод $buildNextPoint(...)$) и потоков $t_1, t_2 \in T$. Будем считать, что в процессе работы была ровно одна итерация цикла $R1_{R, MT} \rightarrow R2_{R, MT} \rightarrow R3_{R, MT}$ для потоков t_1, t_2 , и поток t_1 исполнился раньше, чем поток t_2 . Используя отношение $<_S$ на множестве S^{all} и соотношение (5) получим систему ограничений на возможные потоковые исполнения для t_1, t_2 . Заметим, что система неравенств (6) содержит только те неравенства, которые принципиальны с точки зрения набора ограничивающих условий (7).

$$\begin{cases} F_{B, Rep}^{t_1} < S_{B, MT}^{t_2} : 1, S_{G, Rep}^{t_1} < S_{B, Rep}^{t_1} : 2 \\ S_{G, Rep}^{t_2} < S_{B, Rep}^{t_2} : 3, F_{B, MT}^{t_1} < F_{G, MT}^{t_1} : 4 \\ F_{B, MT}^{t_2} < F_{G, MT}^{t_2} : 5, S_{B, MT}^{t_1} < F_{B, MT}^{t_1} : 6, S_{B, MT}^{t_2} < F_{B, MT}^{t_2} : 7 \end{cases} \quad (6)$$

В качестве инварианта ограничивающей системы неравенств возьмем следующее утверждение: операции $\langle B, MT \rangle$ и $\langle G, MT \rangle$ должны быть исполняться монопольно. С точки зрения последовательной системы данный факт является очевидным. Однако при появлении в системе параллелизма программист вынужден обеспечивать данный инвариант, несмотря на то, что операция модификации точки транспозиции в методе $buildNextPoint(...)$ выполняется монопольно. Данный факт можно выразить следующей системой неравенств:

$$\begin{cases} S_{G, MT}^{t_1} < S_{B, MT}^{t_2} : 8 \\ F_{G, MT}^{t_1} < F_{B, MT}^{t_2} : 9 \end{cases} \quad (7)$$

Построим графическое отображение объединения системы неравенств (6) и (7) в виде графа $G = (V, E)$, где $V \subset S^T$ - множество вершин входящих в набор ограничивающих условий, E - множество пар вершин связанных отношением $<_{s^T}$ на V . Заметим, что система

совместна тогда и только тогда когда граф ациклический, т.е. не имеет циклов. Пример графа G для набора ограничивающих условий (6) и (7) представлен на рис. 4.

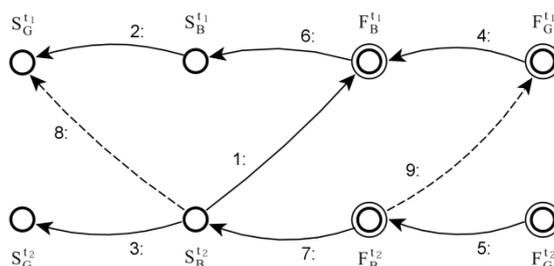


Рисунок 4 – Система неравенств согласно набору ограничивающих условий в виде графа

Если же граф является ациклическим, то выполним топологическую сортировку данного графа. Полученный граф назовем линейризацией истории выполнения параллельного автомата. На рисунке 5 представлена одна из возможных линейризаций графа G .

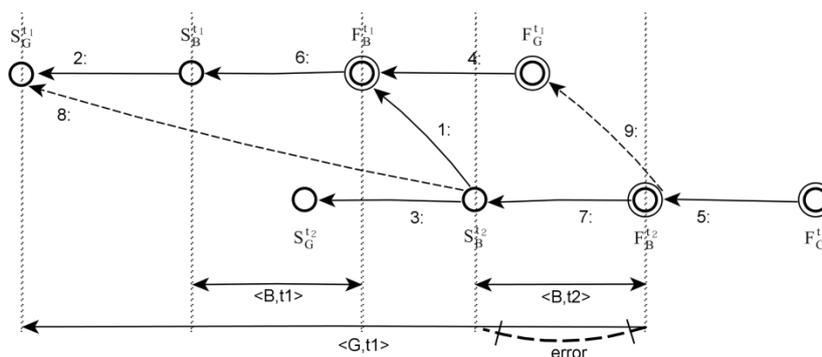


Рисунок 5 – Линейризация графа G построенного согласно набору ограничивающих условий

Физический смысл линейризации заключается в том, что это один из вариантов исполнения параллельного автомата A с использованием одного процессора, а это означает, что мы получили сценарий функционирования параллельной системы, при котором система ограничивающих условий выполняется. Такой подход дает возможность программисту формулировать наборы ограничивающих условий и проверять их выполнимость на этапе разработки многопоточного приложения, а не на этапе тестирования и эксплуатации. В работе выполнена формализация многопоточных взаимодействий параллельной системы, а так же рассмотрены возможные подходы для применения подобной автоматной модели.

Список литературы

1. Finding and reproducing heisenbugs in concurrent programs. / M. Musuvathi, S. Qadeer, T. Ball et al. // OSDI / Ed. by R. Draves, R. van Renesse. – USENIX Association, 2008. – PP. 267–280. <http://dblp.uni-trier.de/db/conf/osdi/osdi2008.html#MusuvathiQBBNN08>.
2. Garg V.K. Concurrent and Distributed Computing in Java / V. K.Garg. – New Jersey: Wiley-IEEE Press, 2004. – 336 p.
3. Мутилин В.С. Метод проверки линейризуемости многопоточных Java программ. // Труды Института системного программирования РАН, 2009., том 16, – С. 89-106.
4. Эндрюс Г.Р. Основы многопоточного, параллельного и распределенного программирования / Г.Р. Эндрюс. – М.: Издательский дом —Вильямс, 2003. – 512 с.

ЭФФЕКТИВНЫЕ АЛГОРИТМЫ ОБРАБОТКИ MIND MAP

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

1. Программное обеспечение обработки интеллект - карт

Mind Map (интеллект-карты) – конструкции, предназначенные для описания отношений между конструкциями, концептуально описывающими некоторые семантические отношения в интеллектуальной деятельности человека. Mind Map позволяют автоматизировать процессы поиска решения, организовать последовательный переход от идеи к реальным результатам, от правил поведения к конкретным инструкциям по решению проблем в процессе перемещения от понятия до плана действия. Ведутся интенсивные разработки программного обеспечения по созданию и использованию интеллект-карт как технологического инструмента решения производственных проблем. Примерами таких разработок являются Carsoo от NuLav Inc., Flying Logic от Sciral, MindManager от Mindjet и т.п. Как правило, такие системы позволяют только создавать карты и не содержат эффективных алгоритмов их обработки. Это приводит к необходимости создания новых инструментов для автоматического анализа интеллект-карт, которые позволяют пользователю непрерывно обеспечивать требуемые свойства производственного процесса. В данной работе рассматривается одна из проблем эффективной обработки: поиск плана разбиения процесса решения на основе Mind Map.

2. Алгоритм обработки дерева интеллект-карты

Будем считать, что интеллект-карта – это разветвленное дерево с большим количеством узлов, каждый из которых представляет собой описание некоторого этапа решения задачи в производственной фирме, занятой обслуживанием клиентов. Как правило, каждый день в такой фирме возникает множество задач, требующих решения при наличии фиксированного числа исполнителей. Поэтому рассмотрим задачу разбиения поддерева решений на отдельные поддеревья - подзадачи. В этом случае останутся нереализованными некоторые k работ, которые представимы в графе дугами, связывающими выделенные поддеревья. Очевидно, что таких работ должно остаться минимальное количество. Таким образом, сформулируем требования: необходимо минимизировать количество дуг, которые связывают отдельные поддеревья графа. Фактически здесь мы рассматриваем не только задачу минимизации количества невыполненных стыковочных работ, но и задачу разбиения всего списка работ на отдельные группы, что более полно отражает проблему распределения ежедневных работ внутри обслуживающей фирмы.

Пусть в дереве T вершина v соответствует поддереву T_v . Для любого целого $1 \leq i \leq |T_v|$ величина $t[v][i]$ - минимальное количество удаленных ребер в вершине v , так что в результате поддерево T_v содержит ровно i вершин. Вычисление массива $t[v][i]$ методом динамического программирования потребует алгоритмической сложности $O(n^3)$: для всех $a+b=i, j=0, 1, \dots, k$ вычисляем значение

$$t[j][v][i] = \min (t[j-1][v][i]+1, \min(t[j-1][v][a] + t[j-1][v][b])).$$

Однако, с учетом леммы [2] вычисления можно выполнить за время $O(n^2)$. Вызывает интерес вопрос эффективного преобразования дерева карты с тем, чтобы минимизировать время поиска. Произвольно преобразовать дерево нельзя, так как иерархическая зависимость характерна для дерева в целом. Однако, если в дереве выделить независимые фрагменты, то преобразование таких поддеревьев на основе декартовых деревьев дает существенный выигрыш во времени обработки.

Список литературы

1. Farrand Paul, Hussain Fearzana and Hennessy Enid (May 2002). The efficacy of the 'mind map' study technique. - *Medical Education* **36** (5). – PP.426–431.
2. Andrychowicz M. Looking for a Challenge?—Warsaw: University of Warsaw, 2012. – 44 p.

СОЗДАНИЕ ДИСТРИБУТИВА НА ОСНОВЕ СВОБОДНО-РАСПРОСТРАНЯЕМОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ДЛЯ РЕАЛИЗАЦИИ УЧЕБНОЙ ДЕЯТЕЛЬНОСТИ ШКОЛЬНИКА

Барильник К.С. – ученица школы, Чемяков А.О. – учитель школы
Муниципальное бюджетное образовательное учреждение «СОШ №10» (г. Славгород)

Актуальность темы данной работы вытекает из того факта, что отсутствует стандартизированный дистрибутив свободно распространяемых программных продуктов, достаточных по функциональности и необходимых школьнику в учебной деятельности.

Объектом исследования выступает программное обеспечение компьютера, необходимое для учебной деятельности школьника.

Предмет исследования – дистрибутивы свободно распространяемого программного обеспечения.

Целью работы является исследование возможности создания дистрибутива свободно распространяемого программного обеспечения для реализации учебной деятельности школьника.

Результаты работы. На основе изучения нормативно-правовой базы использования свободно-распространяемого программного обеспечения в РФ, выявления достоинств и недостатков программного обеспечения, предложенного для использования в образовательных учреждениях, программных средств, необходимых школьнику для реализации учебной деятельности, исследования способов создания дистрибутива и архитектуры файловой системы GNU/Linux создан дистрибутив операционной системы и прикладных программ для реализации учебной деятельности учащимся в школе и дома.

В дистрибутив добавлены дополнительные репозитории Medibuntu, GetDeb, Remastersys и необходимые программные средства: текстовый, графический, аудио- и видео- редакторы, среды объектно-ориентированного программирования на языках: Gambas, Lazarus, Python, среды разработки на языках: Free Pascal, Free Basic, Geany, среда Web-разработки, СУБД, программы для работы в сети Интернет и другие. С целью выделения группы программ для работы с CD и DVD внесены изменения в исходный код и добавлены файлы в соответствующие директории. Установлены недостающие медиакодеки.

Таким образом, на основе выявленных требований к функциональным возможностям программного обеспечения учебной деятельности школьника, создан собственный дистрибутив на основе свободно распространяемого программного обеспечения, что позволит стандартизировать операционную среду для работы школьников в Linux.

ЗАДАЧА О ФИГУРЕ, СОДЕРЖАЩЕЙСЯ В ЛЮБОМ ТРЕУГОЛЬНИКЕ ЕДИНИЧНОЙ ШИРИНЫ

Гичёва Н.И. – ученица школы*, Кизбикенов К.О. – к.ф.-м.н., доцент**

Краевая государственная бюджетная общеобразовательная школа-интернат лицей-интернат
«Алтайский краевой педагогический лицей»* (г. Барнаул)

Кафедра геометрии и математических методов в экономике**

Алтайская государственная педагогическая академия** (г. Барнаул)

Актуальность работы: вопрос о фигуре, которую можно поместить в любую фигуру единичной ширины, стоит длительное время и пока не получил окончательного решения. Задача возникла в начале XX века и имеет связь с теоремой о выпуклых фигурах, открытой в 1913 году австрийским математиком Эдуардом Хелли. Известно одно из её решений – теорема Вильгельма Бляшке. Однако было доказано, что найденная Бляшке фигура не является максимальной по площади. Поискам универсальной фигуры, которая содержится в каждом треугольнике ширины 1, большей по площади среди известных ранее фигур, и посвящена наша работа.

Объектом исследования является универсальная фигура максимальной площади, которая помещается в любом треугольнике ширины 1.

Предмет исследования – треугольник единичной ширины.

Цель нашей работы – найти универсальную фигуру как можно большей площади, которая содержится в любом треугольнике единичной ширины.

Постановка задач. В соответствии с указанной целью были поставлены задачи:

1. Предложить новую, наибольшую по площади среди известных, универсальную фигуру, которая содержится в любом треугольнике единичной ширины.
2. Вычислить площади найденных фигур и сравнить их с площадями известных фигур, помещающихся внутри каждого треугольника единичной ширины.
3. Изучить треугольник Рело, вписанный в правильный треугольник единичной ширины.

Результаты. В ходе работы нами были найдены несколько универсальных фигур, содержащихся в произвольном треугольнике единичной ширины.

Вычислены площади всех найденных фигур, проведено их сравнение с площадями ранее известных универсальных фигур. Получено, что все найденные фигуры имеют площади, большие, чем известные ранее фигуры.

Проведено сравнение найденных фигур по площади между собой и с треугольником, в который они помещаются.

Изучен треугольник Рело, вписанный в правильный треугольник, найдены его площадь и ширина.

Список литературы

1. Болтянский В.Г., Яглом И.М. Выпуклые фигуры. «Библиотека математического кружка». Выпуск 4. – М. – Л.: ГТТИ, 1951. – 343 с.
2. Геометрия: Учеб. для 7–9 кл. сред. шк./Л. С. Атанасян, ГЗБ В. Ф. Бутузов, С.Б. Кадомцев и др. – 3-е изд.– М.: Просвещение, 1992. – 335 с.
3. Радемахер Г., Теплиц О. Числа и фигуры. Опыты математического мышления. «Библиотека математического кружка». Выпуск 10 / Пер. с нем. В. И. Контовта. – М.: Физматгиз, 1962. – 263 с.
4. Эдуард Хелли [Электронный ресурс] / Ю. Белецкий. // Личности. – ООО Издательский дом «Личности», 2009-2011. URL: <http://persons-info.com>.

5. Савин А.П. Энциклопедический словарь юного математика. – М.: Педагогика, 1985. – 352 с.

ОБ ОДНОМ НЕРАВЕНСТВЕ И НЕКОТОРЫХ ТОЖДЕСТВАХ ТРЕУГОЛЬНИКА

Дерипаскин В.С. – ученик гимназии*, Ефремова Е.Н. – учитель математики*

Мальцев Ю.Н. – д.ф.-м.н., профессор**

Муниципальное общеобразовательное учреждение «Гимназия №74»* (г. Барнаул)

Кафедра алгебры**

Алтайская государственная педагогическая академия ** (г. Барнаул)

Актуальность работы: На третьей международной математической олимпиаде 1961 г. была предложена следующая задача: Доказать, что имеет место неравенство:

$$a^2 + b^2 + c^2 \geq 4\sqrt{3}S \quad (1)$$

В книге Д. О. Шклярский и И. М. Яглом «Геометрические неравенства и задачи на максимум и минимум» предложена задача №100: доказать, что для каждого треугольника справедливо:

$$а) a^2 + b^2 + c^2 \geq 4\sqrt{3}S$$

$$б) a^2 + b^2 + c^2 \geq 4\sqrt{3}S + (a-b)^2 + (a-c)^2 + (b-c)^2, \quad (2)$$

где равенство имеет место лишь для равностороннего треугольника.

В недавней работе –On Weitzenboeck's inequality and its generalization” получены новые интересные результаты в этом направлении:

Для каждого треугольника справедливы неравенства:

$$1. \frac{1}{a^2} + \frac{1}{b^2} + \frac{1}{c^2} \leq \frac{3\sqrt{3}}{4S} + \frac{1}{2} \left\{ \left(\frac{1}{a} - \frac{1}{b} \right)^2 + \left(\frac{1}{a} - \frac{1}{c} \right)^2 + \left(\frac{1}{b} - \frac{1}{c} \right)^2 \right\} \quad (3)$$

$$2. a^2 + b^2 + c^2 \leq 4\sqrt{3}S + 3\{(a-b)^2 + (a-c)^2 + (b-c)^2\} \quad (4)$$

$$3. a^2 + b^2 + c^2 \leq 4\sqrt{3}S + \frac{3}{2} \{(a-b)^2 + (a-c)^2 + (b-c)^2\} + 2R^2 \{(\cos \alpha - \cos \beta)^2 + (\cos \alpha - \cos \gamma)^2 + (\cos \beta - \cos \gamma)^2\} \quad (5)$$

$$4. a^{2\lambda} + b^{2\lambda} + c^{2\lambda} \leq 4^\lambda \times 3^{1-\frac{\lambda}{2}} \times S^\lambda + \{(a^\lambda - b^\lambda)^2 + (a^\lambda - c^\lambda)^2 + (b^\lambda - c^\lambda)^2\} \quad (6)$$

Из приведенного выше перечня работ следует, что интерес к получению подобных неравенств и тождеств в треугольнике не утихает, поэтому новые результаты в этом направлении обязательно будут востребованы.

Объект исследования: объектом исследования является геометрия треугольника.

Предмет исследования: предметом исследования являются соотношения, связывающие основные параметры треугольника.

Цель исследования: получить обобщения и уточнения неравенств (1-2), усилить утверждения работы –On Weitzenboeck's inequality and its generalization”.

Решаемые задачи:

1. Указать точные значения разностей правых и левых частей неравенств (3), (4), (5).

2. Предложить другие доказательства неравенств (3), (4), (5).
3. Уточнить неравенство (6) для $\lambda = 2$.
4. Получить новое выражение для $a^2 + b^2 + c^2 - 4\sqrt{3}S$.

Методы исследования: Алгебраические преобразования, использование тождеств и неравенств [2].

Результаты, полученные в работе: Для любого треугольника справедливы следующие соотношения:

Теорема 1.

$$\frac{3\sqrt{3}}{4S} + \frac{1}{2} \left[\left(\frac{1}{a} - \frac{1}{b} \right)^2 + \left(\frac{1}{a} - \frac{1}{c} \right)^2 + \left(\frac{1}{b} - \frac{1}{c} \right)^2 \right] = \frac{1}{a^2} + \frac{1}{b^2} + \frac{1}{c^2} + \frac{3\sqrt{3}R-p}{2pRr}.$$

Теорема 2.

$$\begin{aligned} 4\sqrt{3}S + 3((a-b)^2 + (a-c)^2 + (b-c)^2) = \\ = a^2 + b^2 + c^2 + 4\sqrt{3}r(p - 3\sqrt{3}r) + 4[p^2 - (16Rr - 5r^2)]. \end{aligned}$$

Теорема 3.

$$\begin{aligned} a^2 + b^2 + c^2 + 2\{(4R^2 + 4Rr + 3r^2) - p^2\} + 4\sqrt{3}(p - 3\sqrt{3}r)r + 8(R - 2r)^2 + 4r(R - 2r) = \\ = 4\sqrt{3}S + \frac{3}{2}\{(a-b)^2 + (a-c)^2 + (b-c)^2\} + \\ + 2R^2\{(\cos \alpha - \cos \beta)^2 + (\cos \alpha - \cos \gamma)^2 + (\cos \beta - \cos \gamma)^2\}. \end{aligned}$$

Теорема 4.

$$a^4 + b^4 + c^4 = 16S^2 + \{(a^2 - b^2)^2 + (a^2 - c^2)^2 + (b^2 - c^2)^2\}.$$

Следствие 1.

$$a^4 + b^4 + c^4 = 16S^2 \Leftrightarrow a = b = c.$$

Теорема 5.

$$\begin{aligned} a^2 + b^2 + c^2 + [p^2 - (16Rr - 5r^2)] + 4\sqrt{3}(p - 3\sqrt{3}r)r = \\ = 4\sqrt{3}S + \frac{3}{2}[(a-b)^2 + (a-c)^2 + (b-c)^2] + 12r(R - 2r). \end{aligned}$$

Следствие 2.

$$a^2 + b^2 + c^2 \leq 4\sqrt{3}S + \frac{3}{2}[(a-b)^2 + (a-c)^2 + (b-c)^2] + 12r(R - 2r).$$

Список литературы

1. Шклярский Д.О., Ченцов Н.Н., Яглом И.М. Геометрические неравенства и задачи на максимум и минимум. – Москва: Изд-во Наука, 1978.
2. Солтан В.П., Мейдман С.И. Тождества и неравенства в треугольнике. – Кишинев: Изд-во Штиинца, 1982.
3. Shan-He Wu, Zhi-Hua Zhang, Zhen-Gang Xiao On Weitzenboeck's inequality and its generalization. – <http://rgmia.org/papers/v6n4/Weitzenboeck.pdf>

О НЕКОТОРЫХ НЕРАВЕНСТВАХ В ГЕОМЕТРИИ ТРЕУГОЛЬНИКА

Кадилов Р.В. – ученик гимназии

Муниципальное бюджетное образовательное учреждение «Гимназия № 40» (г. Барнаул)

Актуальность работы. Геометрия треугольника справедливо считается одним из интереснейших разделов элементарной геометрии. Достаточно много публикаций представлено в научных и научно-популярных изданиях, которые посвящены исследованиям соотношений некоторых элементов треугольника.

Цель работы. Получить новые неравенства для медиан через основные параметры треугольника (p , r , R).

Методы исследования. Анализ результатов из книги В. П. Солтана, С. И. Мейдмана «Тождества и неравенства в треугольнике», работы J. L. Diaz-Barrero, J. Gibergans-Baguena «NewInequalitiesfortheTriangle».

В ходе исследования были получены следующие результаты, сформулированные в виде теорем:

Теорема 1

$$\text{а) } 9p^2r^2 \leq m_a^2m_b^2 + m_a^2m_c^2 + m_b^2m_c^2;$$

$$\text{б) } 3pr \leq \frac{m_a m_b m_c (m_a + m_b + m_c)}{\sqrt{m_a^2 m_b^2 + m_a^2 m_c^2 + m_b^2 m_c^2}}.$$

Теорема 2

$$\text{а) } 3\sqrt{3}pr \leq m_a m_b m_c \left(\frac{1}{m_a + m_b - m_c} + \frac{1}{m_a + m_c - m_b} + \frac{1}{m_b + m_c - m_a} \right);$$

$$\text{б) } 9\sqrt{3}p^3r^3 \left(\frac{1}{m_a + m_b - m_c} + \frac{1}{m_a + m_c - m_b} + \frac{1}{m_b + m_c - m_a} \right) \leq m_a m_b m_c (m_a + m_b + m_c)^2$$

Теорема 3

$$\text{а) } 9 \leq (m_a + m_b + m_c) \left(\frac{1}{m_a + m_b - m_c} + \frac{1}{m_a + m_c - m_b} + \frac{1}{m_b + m_c - m_a} \right);$$

$$\text{б) } \frac{1}{m_a + m_b - m_c} + \frac{1}{m_a + m_c - m_b} + \frac{1}{m_b + m_c - m_a} \leq \frac{m_a m_b m_c}{(pr)^2}.$$

Теорема 4

$$\text{а) } pr \leq \frac{m_a m_b m_c}{\sqrt{m_a m_b + m_b m_c + m_a m_c}};$$

$$\text{б) } 9pr \leq (m_a + m_b + m_c) \sqrt{m_a m_b + m_b m_c + m_a m_c}.$$

В работе доказаны новые нетривиальные неравенства для медиан произвольного треугольника. Полученные оценки зависят от основных параметров треугольника (p , r).

ОБ ОДНОЙ ЗАДАЧЕ ИЗ ЖУРНАЛА «CRUX MATHEMATICORUM»

Мальцев Р.А. – ученик гимназии, Мальцев Ю.Н. – д.ф.-м.н., профессор*
Муниципальное бюджетное образовательное учреждение «Гимназия №74» (г.Барнаул)
Кафедра алгебры*
Алтайская государственная педагогическая академия * (г. Барнаул)

В журнале Crux Mathematicorum была предложена задача (№ 1130 , стр.36):

доказать, что $a^{\frac{3}{2}} + b^{\frac{3}{2}} + c^{\frac{3}{2}} \leq 3^{\frac{7}{4}} \cdot R^{\frac{3}{2}}$, где a, b, c, R – соответственно длины сторон и радиус описанной окружности произвольного треугольника.

Цель работы - исследовать сумму $a^\lambda + b^\lambda + c^\lambda$, где $\lambda = \frac{m}{2}$, $m \in N$.

Объект исследования – геометрия треугольника.

Предмет исследования – неравенства для основных параметров треугольника.

В работе получены следующие результаты:

- 1) $a^n + b^n + c^n \leq 3^n R^n$, где $n \geq 2, n \in Z$;
- 2) $a^{\frac{n}{2}} + b^{\frac{n}{2}} + c^{\frac{n}{2}} \leq 3^{\frac{2n+1}{4}} R^{\frac{n}{2}}$, где n – нечётное целое число большее или равное 3.
- 3) $a^n + b^n + c^n \geq 2^n \cdot 3^{1+\frac{n}{2}} \cdot r^n$, где n – любое натуральное число.

Для исследования были использованы:

- метод математической индукции;
- неравенство Коши – Буняковского;
- неравенство Чебышева;
- неравенства, взятые из книги «Тождества и неравенства в треугольнике».