

ВЫЧИСЛИТЕЛЬНЫЕ СИСТЕМЫ И ПРОГРАММИРОВАНИЕ

УДК 004.021

Д. М. Азаров – студент кафедры комплексной защиты информации

В. А. Килимник (канд. техн. наук, ст. науч. сотр) – научный руководитель

ВИЗУАЛИЗАЦИЯ ДАННЫХ С ВЫСОКОЙ СКОРОСТЬЮ ИХ ПОСТУПЛЕНИЯ В ЭРГАТИЧЕСКИХ СИСТЕМАХ УПРАВЛЕНИЯ В РЕЖИМЕ РЕАЛЬНОГО ВРЕМЕНИ

Эргатические системы, как системы управления, одним из элементов которой является человек, нашли свое применение там, где вмешательство оператора в работу объекта является на сегодняшний день необходимым условием обеспечения надежной работы. Одними из основных критериев оптимальности системы является скорость и качество отображения данных с возможностью их предварительной обработки и выводом на экран, в режиме оптимальном для разрешающей способности глаза человека. Основной задачей проделанной работы являлось построение системы для визуализации данных с высокой скоростью их поступления в эргатических системах управления в режиме реального времени.

Задача решалась в операционной системе Windows 7, на аппаратной платформе IBM PC x86/x64.

На первом этапе исследований для решения поставленной задачи был выбран Windows Presentation Framework (далее WPF) – графическая подсистема .NET Framework. .NET Framework – программная платформа компании Microsoft. Одной из основных идей Microsoft.NET является совместимость программных частей, написанных на разных языках программирования. Одним из важных плюсов данной системы является возможность использования уже готовых объектов для рисования из системы WPF. Идеально для систем с количеством объектов для рисования меньше 50000. Однако при количестве объектов больше 50000 наблюдается сильная нагрузка на оперативную память и снижение производительности операционной системы [1, 3].

На следующем этапе решения задачи был выбран способ визуализации данных – Native WPF. Native WPF – метод низкоуровневой прорисовки геометрических объектов для приложений WPF. Решением является переопределение методов рисования элементов управления и использование низкоуровневых функций рисования, доступных в .NET. С этой целью нами была протестирована библиотека с открытым исходным кодом WPF Dynamic Data Display. На этом шаге система легко справлялась с нагрузками при количестве объектов до 50000 и на порядок лучше использовала оперативную память и ресурсы ПК.

Далее, в целях оптимизации системы была проведена оценка целесообразности использования способа отображения данных в эргатических системах на базе Direct2D – программном интерфейсе (API), использующем специализированные технические средства компьютера для вывода изображения на экран (графический процессор видеокарты), который обеспечивает высокую производительность и высококачественное отображение двумерных геометрических объектов, растровых изображений и текста. Direct2D API

является продуктом компании Microsoft для создания приложений под управлением операционной системы Windows и для взаимодействия с существующим кодом, который использует технологии GDI, GDI+, или Direct3D.

Direct 2D – низкоуровневый набор инструкций, написанных на C++, который можно использовать в приложениях с управляемым кодом с минимальной абстракцией. Концепция Direct 2D – «моментальный режим». В отличие от других аналогичных систем (DirectX, OpenGL) в Direct2D нет «идеи сцен», и рисование происходит при соответствующих вызовах методов рисования [2].

Вместе с тем, Direct 2D может уменьшить нагрузку на центральный процессор и использовать визуализацию на видеокarte, которая поддерживает Direct 3D, начиная от версии 10.1. Direct 2D поддерживает высокоуровневые возможности, такие как:

- вывод на экран текста ClearType (с использованием DirectWrite);
- сглаживанием по примитивам;
- высокоуровневое рисование геометрических фигур (кривые, окружности и т. д.).

Приложения, которые используют Direct2D могут добиться более высокого качества графики, в отличие от результата, предоставляемого GDI – стандартной системы отображения для более ранних версий Windows.

Сравнение технологических характеристик систем визуализации данных WPF, WPF Native, Direct2D для использования их в эргатических системах приведено в табл. 1.

Таблица 1

Сравнение технологических характеристик систем визуализации данных WPF, WPF Native, Direct2D

	WPF	Native WPF	Direct2D
FPS (24k points)	8	14	98
Memory (24k points), MB	48	47	1

Таким образом, сравнение 3-х систем визуализации данных позволяет сделать вывод, что из выбранных для анализа систем только использование Direct2D позволит с достаточной скоростью и хорошим качеством визуализировать данные с высокой скоростью их поступления в эргатических системах управления в режиме реального времени.

Библиографический список

1. <http://jeremiahmorrill.com/2011/02/14/a-critical-deep-dive-into-the-wpf-rendering-system/>
2. <http://msdn.microsoft.com/en-us/library/dd370990.aspx>
3. Мак-Дональд М. WPF. Windows Presentation Foundation v.NET 3.0 для профессионалов, ISBN 978-5-8459-1344-9, 1-59-059782-6.

УДК 004.021

Д. М. Азаров – студент кафедры комплексной защиты информации
В. А. Килимник (канд. техн. наук, ст. науч. сотр.) – научный руководитель

**РИТМОЛОГИЧЕСКИЙ КОМПЛЕКС
 ДЛЯ ОЦЕНКИ ФУНКЦИОНАЛЬНОГО СОСТОЯНИЯ**

Настоящее исследование являлось фрагментом работы, основной задачей которой являлась создание аналого-цифрового автономного многоканального комплекса для оценки функционального состояния биологического объекта.

Основные задачи проводившегося исследования заключались в создании программы обработки данных в модельных условиях и разработке интуитивно понятного интерфейса программного комплекса.

Для разработки и отладки программы использовался источник сигнала, формируемый ЭКГ-модулем Alive ECG (Alive Technologies Ltd, Австралия), представленный на рис. 1, обеспечивающий съем электрокардиографического сигнала контактным способом, с последующей телеметрической передачей информации на базовую станцию (ПК).

Разработка программного обеспечения осуществлялась в операционной системе Windows 7, на аппаратной платформе IBM PC x86/x64. На данный момент реализовано 3 блока обработки данных и их визуального представления функционального диагностирования человека. Все блоки реализованы в виде.net компонент [1]. Такой метод создания программного обеспечения, как компонентное программирование, появился относительно недавно. Его можно охарактеризовать как технологию создания программного обеспечения из готовых блоков. Эта технология позволяет использовать готовые компоненты, как в разрабатываемом комплексе, так и в его модификациях.

Разработка велась для платформы Microsoft.NET Framework 4.0 с использованием Windows Presentation Foundation — новейшей технологии создания пользовательских интерфейсов для Windows. Windows Presentation Foundation — это полностью векторная технология реализации элементов управления с произвольной визуализацией и большим количеством готовых типовых элементов. Основным принципом разработки являлось отделение части обработки и визуализации информации от ее получения. Это позволило добиться возможности легкой замены одного источника данных на другой.

Все компоненты обрабатывают данные в многопоточном режиме. Многопоточные приложения более оперативно реагируют на различные действия пользователя, поскольку пользовательский интерфейс остается активным, в то время как задачи, требующие интенсивной работы процессора, выполняются в других потоках.

Первый разработанный компонент – модуль вариационной пульсометрии. Сущность метода состоит в изучении закона распределения кардиоинтервалов как случайных величин в исследуемом ряду их значений с помощью специфических показателей статистического анализа. Анализ основан на интерпретации показателей, получаемых с помощью специального графика – интервальной гистограммы. Для построения графика кардиоинтервалы группируются по своему значению с различной шириной разряда. Полученные виды гистограмм варьируются в зависимости от выбранной ширины разряда. Это позволяет провести более глубокий анализ для каждого исследования [2].

Второй компонент – модуль двумерной скаттерграммы. Сущность программной обработки данных состоит в построении графического изображения точек, каждая из которых соответствует продолжительности двух соседних R-R интервалов, при этом ордината точки соответствует текущему, а абсцисса – последующему R-R интервалу. Данный вид исследования позволяет квалифицированному специалисту произвести первичный анализ отклонения кардиосигнала от предполагаемой нормы.

Третий компонент – модуль ритмограммы. Ритмограмма – графическое изображение продолжительности R-R интервалов. При построении ритмограммы на оси абсцисс откладывается время записи, а на оси ординат – продолжительность каждого кардиоинтервала.

Графическая иллюстрация данных, полученных с помощью Alive ECG и разработанного программного обеспечения, приведена на рис. 2.



Рис. 1. ЭКГ-модуль Alive ECG (Alive Technologies Ltd, Австралия)

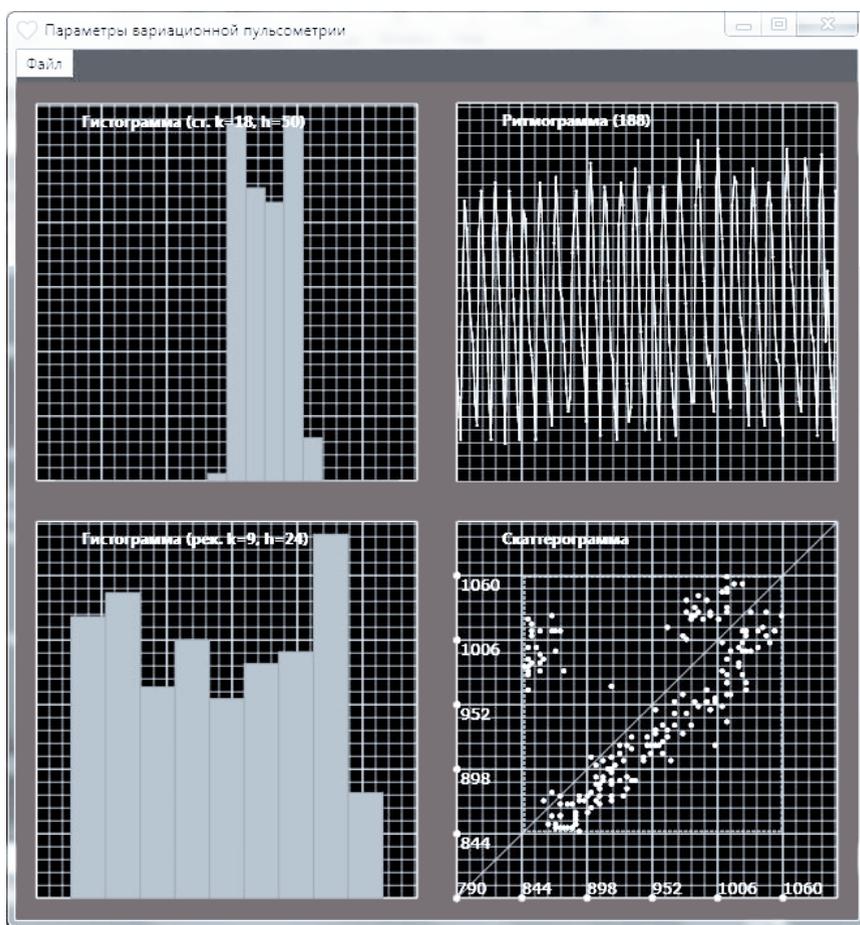


Рис. 2. Графическая иллюстрация параметров вариационной пульсометрии, полученных с помощью Alive ECG и разработанного программного обеспечения

Полученные результаты позволяют производить дальнейшее развитие комплекса, а также статистическую обработку и графическое представление обрабатываемых данных.

Библиографический список

1. <http://msdn.microsoft.com/en-us/library/ms973807.aspx>
2. Бабунц, И. В. Азбука анализа вариальности сердечного ритма / И. В. Бабунц, Э. М. Мириджаниян, Ю. А. Машаех. Ставрополь, 2002. С. 112.

УДК 004.416.6: 681.324

М. А. Алексеев – магистрант кафедры вычислительных систем и сетей
А. В. Гордеев (д-р техн. наук, проф.) – научный руководитель

СИСТЕМА NetEmul ДЛЯ ИМИТАЦИОННОГО МОДЕЛИРОВАНИЯ ЛОКАЛЬНЫХ ВЫЧИСЛИТЕЛЬНЫХ СЕТЕЙ

Программа NetEmul ставит своей целью предоставить возможность визуализации происходящих в сети процессов, связанных с передачей служебной и пользовательской информации (рис. 1).

Кроме визуализации работы телекоммуникационной сети, следует отметить возможность более детального рассмотрения ее работы с помощью отображаемой статистики и службы для гибкой трассировки происходящих событий, для каждого сетевого устройства. Как правило, в небольших сетях отсутствуют аппаратные сетевые компоненты, которые могут предоставить пользователю статистику, управление, и вызывать события на различные ситуации работы сети. В более крупных, корпоративных сетях далеко не всегда начинающего сетевого администратора, тем более студента-практиканта, допустят до экспериментирования и управления сетью.

В результате данный проект лучшим образом может подходить для использования его в качестве электронного, интерактивного методического пособия всеми техническими специалистами по дисциплинам, напрямую или косвенно касающимся телекоммуникационных сетей передачи данных.

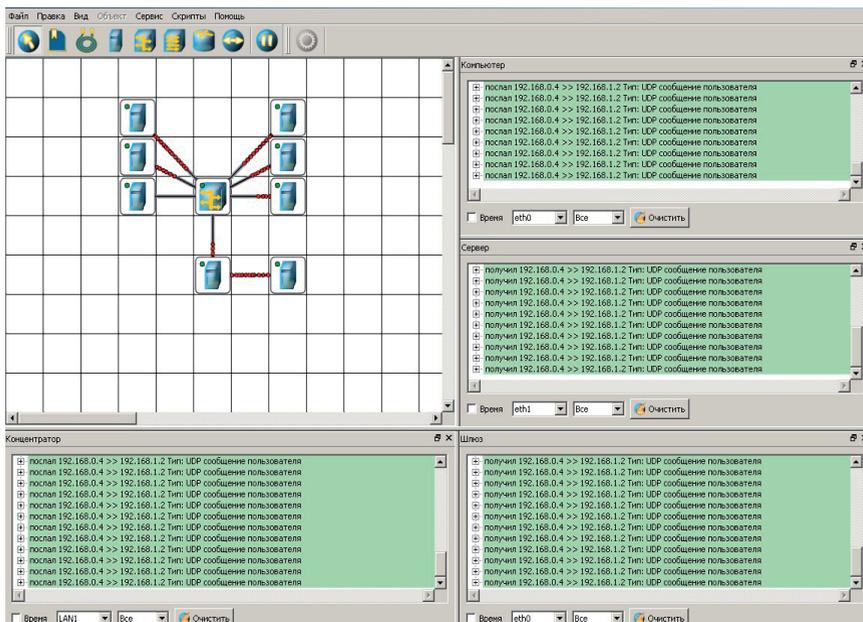


Рис. 1

Основными характеристиками системы NetEmul являются:

- интуитивно понятный, «дружественный» интерфейс;
- полноценная справочная система;
- наличие готовых примеров сетей;
- наличие готовых методических разработок;
- использование визуальной симуляции работы основных сетевых протоколов;
- поддержка разработчиками через сайт проекта;
- возможности настройки среды, локализации интерфейса и устройств;
- расширяемость функциональности используя встроенный язык сценариев;
- кроссплатформенность (работа в операционных системах Windows, Linux, Mac OS), благодаря этому программа может быть полезна как в учебных учреждениях, использующих свободное ПО, так и в домашних условиях обычными пользователями различных ОС.

NetEmul свободно распространяется по лицензии GPL.

Программа предоставляет возможность преподавателю:

- проводить разнообразные уроки с использованием наглядного пособия:
 - мастер-классы;
 - практические и лабораторные занятия;
 - дистанционные занятия;
 - домашнее задание;
 - быстро составлять примеры различных сетей;
 - прививать больший интерес к специальным предметам, так как практическая сторона предметной области сильнее захватывает, чем любая, даже очень хорошо подготовленная теоретическая часть;
 - оценить практические навыки студентов.
- Программа дает следующие возможности студенту:
- самостоятельно изучать компьютерные сети;
 - убедиться в работоспособности сконфигурированной сети без помощи преподавателя;
 - проверить эффективность той или иной конфигурации и/или топологии сети;
 - проводить эксперименты по созданию, настройке и использованию сетей;
 - при повышенном уровне образования, более детально разобраться в предметной области;
 - самостоятельно повышать практические навыки создания работоспособной сети.

Программа может быть полезной и в других случаях. Например, при неисправно работающей физической сети можно достаточно быстро создать ее виртуальную модель и наглядно увидеть большинство стандартно проявляющихся неисправностей, или при более детальном моделировании ситуаций разобраться практически с любой логической неисправностью. Это свойство программы сложно переоценить для молодого специалиста, начинающего администрировать или обслуживать локальные компьютерные сети.

Кроме того, в программе реализованы следующие возможности:

- добавление и настройка компьютеров, маршрутизаторов, коммутаторов и концентраторов;
- настройка IP адресов, подсетей и шлюзов;
- визуализация движения пакетов по сети;
- работают протоколы RIP, ARP, DHCP;
- встроенный скриптовый язык.

Из недостатков программы можно выделить такие вещи, как невозможность отключить использование визуальной симуляции работы основных сетевых протоколов. Нет возможности сбора статистики, а так же не хватает базы данных с возможностью занесения данных статистики, с последующей обработкой разнообразными специализированными пакетами. Не реализована возможность создавать сеть из сетей.

Нет консоли с утилитами ping, traceroute, так же было бы очень полезно иметь возможность настраивать таблицу маршрутизации из командной строки. В связи с этим мною было принято решение по доработке данного продукта. По исходному коду программы NetEmul была изучена её архитектура. Для изучения исходников была использована программа Doxygen, так как программный код программы состоит из 208 файлов исходного кода (общий объем текста составляет 1 105 920 байтов).

В процессе работы над этим проектом были написаны утилиты ping, traceroute, добавлен протокол SNMP (Simple Network Management Protocol). Теперь эта программа обладает дополнительными возможностями.

УДК 621.317.08; 681.2.001.5; 615.47

В. В. Белов – студент кафедры моделирования вычислительных и электронных систем
О. О. Жаринов (канд. техн. наук, доц.) – научный руководитель

ОДИН ИЗ МЕТОДОВ ПОВЫШЕНИЯ ПОМЕХОУСТОЙЧИВОСТИ ПОРТАТИВНЫХ КАРДИОМОНИТОРОВ

На сегодняшний день в медицине широкое распространение получило использование портативных средств регистрации электрокардиосигнала. Данные устройства используются в стационаре и после выписки из стационара для контроля таких изменений состояния сердечной деятельности за весь период суточной активности, которые не могут быть выявлены во время непродолжительного ЭКГ-исследования в покое. На основании полученных данных производится выбор и дозировка лекарственных препаратов и определение допустимых физических нагрузок. Малые габаритные размеры, масса и автономное питание позволяют применять портативные кардиомониторы в повседневной жизни, не ограничивая свободу действий пациента в течение 24 ч. Существует ряд проблем, решение которых в дальнейшем позволит улучшить качество записываемого сигнала, а также предотвратить потерю информации при возникновении различных условий в процессе регистрации ЭКГ.

Параметры информационного сигнала и помех

Электрокардиосигнал (ЭКС) представляет собой запись изменяющейся во времени разности потенциалов, создаваемой электрической активности сердца [1]. Известно, что электрическая активность сердца представляет собой квазипериодический процесс, представляющий собой последовательную смену процессов деполяризации и реполяризации клеток миокарда, и отражает процесс электрического возбуждения, а не сокращения сердца. Электрокардиограмма содержит ряд характерных элементов, обозначения которых приведены на рис. 1.

Полярность и амплитуда зубцов зависят от расположения электродов на поверхности тела, а длительность зубцов, интервалов и сегментов – от особенностей состояния сердца обследуемого.

Поверхностные потенциалы образуются

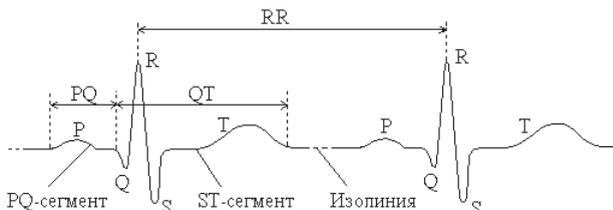


Рис. 1. Элементы электрокардиосигнала

не только электрической активностью сердца, но и помеховыми источниками, как внутренними, так и внешними по отношению к объектам исследования, что затрудняет построение автоматических методов обработки. В основном помехи обусловлены четырьмя составляющими:

- 1) наводка от электросети с частотой основной гармоники 50 Гц;
- 2) шумы аппаратуры (тепловые шумы усилителей и электродов);
- 3) физиологический шум (в первую очередь сигналы электрического возбуждения скелетной мускулатуры);
- 4) эффект поляризации электродов, приводящий к медленному дрейфу нуля.

Структурная схема типового портативного трехканального кардиомонитора.

В состав современных типовых портативных регистраторов ЭКГ входят:

1. Усилитель электрокардиосигнала, в состав которого входят:

– инструментальный усилитель [2], при выборе которого обращают внимание на такие характеристики, как высокий коэффициент подавления синфазного сигнала, низкий уровень собственных шумов, низкий потребляемый ток;

- RC-цепочка, для блокировки постоянной и медленно меняющейся составляющей сигнала;
- операционный усилитель.
2. Мультиплексор, предназначенный для переключения каналов.

3. Аналого-цифровой преобразователь, основной задачей которого является преобразование аналогового сигнала в цифровой код.

4. Микроконтроллер, осуществляющий управление АЦП, мультиплексором.

5. Для последующей обработки записанного сигнала на персональном компьютере, необходимо осуществить передачу данных, в качестве устройств передачи, как правило, используется USB разъем.

Структурная схема типового портативного трехканального кардиомонитора изображена на рис. 2.

Формулировка проблемы

Основной задачей портативного кардиомонитора является обеспечение качественного съема ЭКГ в течение длительного интервала времени. Однако существует несколько ситуаций, при возникновении которых происходит ухудшение качества записываемого сигнала, а также потеря данных. Одной из таких ситуаций является нарушение контакта одного из измерительных электродов. При возникновении данной ситуации происходит дрейф нуля электродного потенциала. На рис. 3 приведен общий вид дрейфа нуля электродного потенциала при нарушении контакта одного из трех измерительных электродов. Если дрейфовая составляющая сигнала начинает превышать максимальные входные значения аналого-цифрового преобразователя, то происходит потеря информации, которую в дальнейшем невозможно компенсировать при последующей обработке

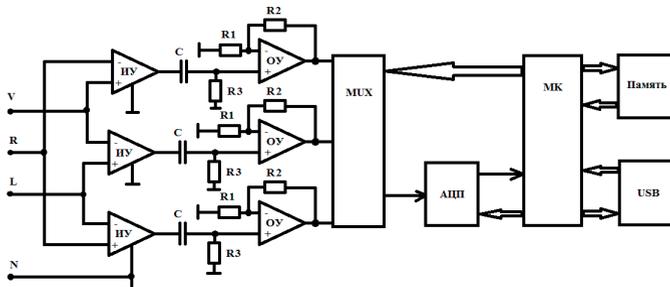


Рис. 2. Структурная схема типового портативного трехканального регистратора ЭКГ

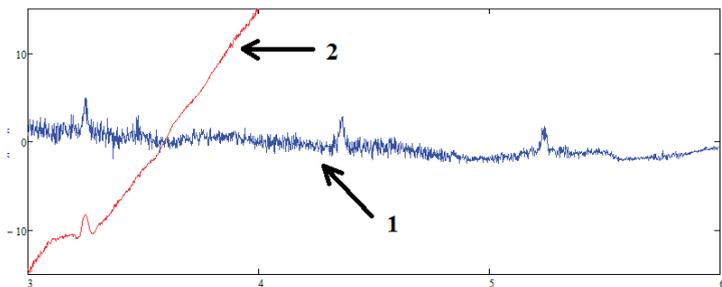


Рис. 3. Пример общего вида дрейфа нуля электродного потенциала при нарушении контакта одного из электродов (1 – типовой ЭКС, 2 – ЭКС при нарушении контакта электрода)

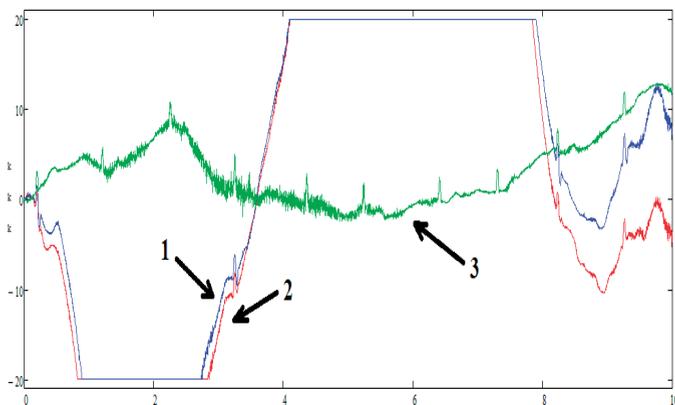


Рис. 4. Результат записи ЭКС стандартным трехканальным портативным кардиомонитором при нарушении контакта с кожей одного из трех измерительных электродов (1, 2 – ЭКС при нарушении контакта электрода с кожей, 3 – типовой ЭКС)

на ПК. На рис. 4 приведен результат записи ЭКС стандартным трехканальным портативным кардиомонитором при нарушении контакта одного из трех измерительных электродов. В современных доступных источниках информации практически нет никаких сведений о попытках решения проблемы потери информации при нарушении контакта одного из измерительных электродов. Хотя ситуации, когда нарушается контакт одного, или нескольких электродов случаются довольно часто, так как запись ЭКГ ведется в течении длительного интервала времени (24 часа) и при этом большая часть записи может оказаться совершенно непригодной для последующей компьютерной обработки.

Решение проблемы

С целью повышения помехоустойчивости отведений съема (предотвращения потери информации) при нарушении контакта одного из трех измерительных электродов с кожей, был предложен метод сброса дрейфовой составляющей, предполагающий модификацию аппаратной части типового портативного трехканального кардиомонитора. Модификация предполагает изменение структурной схемы типового портативного

трехканального регистратора ЭКГ, путем ввода в ее состав трех ключей, трех светодиодов и динамика. Структурная схема модифицированного кардиомонитора приведена на рис. 5. Метод заключается в сравнении выходных кодов АЦП, поступающих на вход микроконтроллера, с максимальными выходными значениями аналого-цифрового преобразователя. Если полученный контроллером код, является максимальным, то подается команда коррекции. Коррекция будет осуществляться путем замыкания на «землю» резистора в RC-контуре. Далее, производится определение электрода, на котором произошло нарушение контакта с кожей. После того, как электрод с нарушением контакта будет определен, последует индикация соответствующего светодиода и будет подан звуковой сигнал. Светодиоды будут отвечать за индикацию, показывая на каком из трех электродов произошло нарушение контакта, а наличие звукового сигнала, позволит увеличить вероятность того, что пациент заметит сигналы светодиодов и примет меры по устранению нарушения контакта. Если же пациент спит, или находится в месте с повышенными акустическими воздействиями, то метод коррекции дрейфовой составляющей решит проблему потери информации. Алгоритм коррекции дрейфовой составляющей приведен на рис. 6. Данное усовершенствование не увеличит габаритные размеры портативного кардиомонитора, так как размеры современных полевых транзисторов и твердотельных реле, использование которых предполагается

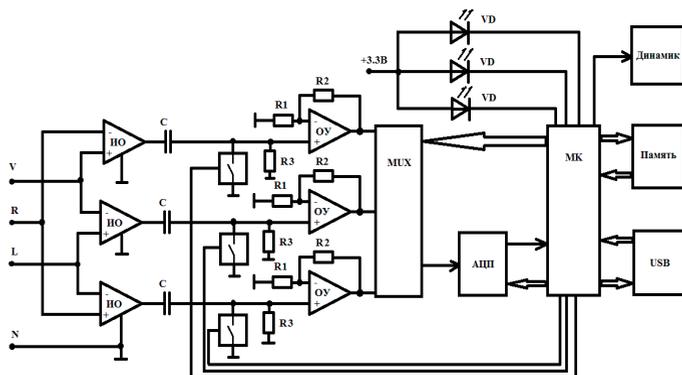


Рис. 5. Алгоритм сброса дрейфовой составляющей

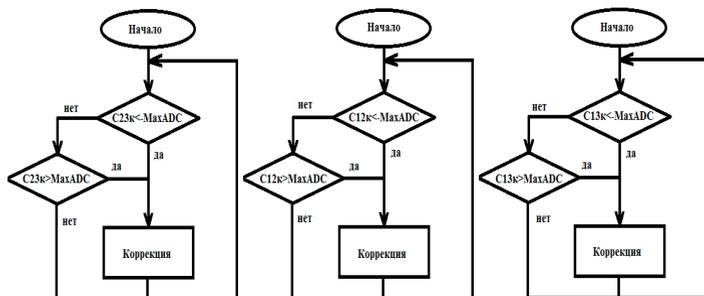


Рис. 6. Алгоритм коррекции дрейфовой составляющей

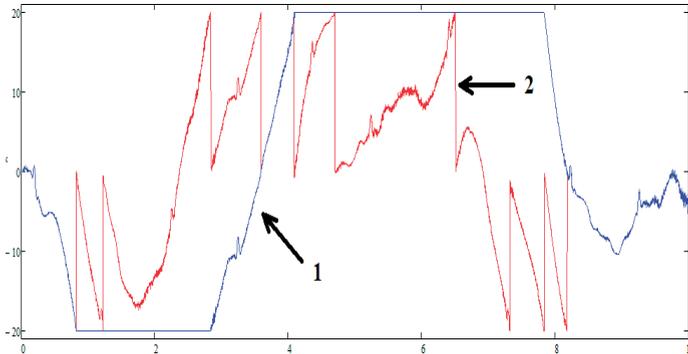


Рис. 7. Пример работы метода коррекции дрейфовой составляющей ЭКС
(1 – ЭКС в типовом КМ, 2 – скорректированный ЭКС)

в качестве ключей, невелики, как и размеры светодиодов. Также эта доработка не значительно увеличит стоимость портативного регистратора ЭКГ. Пример работы данного метода приведен на рис. 7.

Заключение

Как было показано, данный метод позволяет значительно улучшить помехоустойчивость портативных средств регистрации ЭКГ при нарушении контакта одного из измерительных электродов. Предотвращение потери информации, при длительном съеме ЭКС, является необходимым условием для дальнейшей автоматизированной обработки информации и последующего вынесения диагноза.

Библиографический список

1. Жаринов О. О., Белов В. В. Методические подходы к автоматическому анализу электрокардиограмм / Научно-техническая конференция ГУАП: Сб. докл. Ч. 1. Технические науки. СПб.: СПбГУ-АП, 2010. С. 205–208.
2. Зайченко К. В., Жаринов О. О., Кулин А. Н. Съём и обработка биоэлектрических сигналов: учебн. пособие. СПб.: СПбГУАП, 2001.

УДК 004.946, 004.031.42.

Д. А. Булгаков – магистрант кафедры вычислительных систем и сетей
Н. Н. Решетникова (канд. техн. наук, доц.) – научный руководитель

РЕАЛИЗАЦИЯ ТРЕХМЕРНОЙ ИНТЕРАКТИВНОЙ МОДЕЛИ ФРАГМЕНТА ГОРОДСКОЙ ТРАНСПОРТНОЙ СИСТЕМЫ В ПАКЕТЕ VIRTOOLS

С каждым годом все больше растет число автомобилистов и, как следствие, усложняется обстановка на дорогах. Молодой водитель может неуверенно себя чувствовать в плотном потоке машин. Пестрые рекламные вывески отвлекают внимание от грязных дорожных знаков и стертой разметки. Реальный городской трафик оказывается куда сложнее, чем типовые ситуации, рассмотренные при обучении. Совокупность

всех этих факторов может приводить к нарушениям и авариям, порой случайным, а не умысленным.

В данной статье рассматривается часть проекта по созданию интерактивной обучающей программы изучения правил дорожного движения (ПДД), в которой реализуется фрагмент транспортной системы в пакете Virtools [1, 3].

Для реализации транспортной системы, во-первых, нужно создать модель городской среды, включающую в себя сеть дорог с различным количеством полос и различной разметкой, перекрестки, а также средства регулирования дорожного движения (светофоры и дорожные знаки). Во-вторых, нужен персонаж, которым будет управлять пользователь. В данном случае роль персонажа будет выполнять модель автомобиля. В-третьих, необходимо соединить модели в одной сцене, настроить материалы, источники освещения и корректно расположить светофоры и знаки. Наконец, используя язык сценариев, запрограммировать определенные ПДД.

Несколько слов об используемой программной платформе.

3DVIA Virtools – это конечная платформа для разработки интерактивных трехмерных сцен. В ней присутствует поддержка многопользовательского режима, 3D отображения сцены, возможно подключение внешних контроллеров. Для реализации физического взаимодействия используется популярный движок Havok [4]. Virtools не является 3D редактором, следовательно он не позволяет создавать и редактировать модели. Для работы с моделями использовался популярный редактор 3ds Max 2009 [5].

Внешний вид программы показан на рис. 1.

Рассмотрим подробнее некоторые наиболее важные правила и методы их программной реализации.

Согласно правилам, автомобиль имеет право двигаться только по одной из полос попутного движения, исключая случаи поворота налево и разворота в разрешенных для этого местах, когда происходит вынужденное пересечение полосы встречного движения. Запрещено двигаться по разделительной дорожной разметке, разделительной полосе и тротуарам.

Для реализации контроля необходимо, чтобы система реагировала на перемещение автомобиля пользователя и регистрировала каждое вхождение объекта в запретную зону. Для этого используем стандартный объект движка Virtools – сетку (grid) [2].

Сетка представляет собой объемную невидимую фигуру, состоящую из одинаковых квадратов (по умолчанию). Её особенность в том, что возможно для каждого квадрата задать цветовой оттенок от 0 до 255, а затем использовать эти оттенки для программирования поведения системы.

Установив такие сетки на разделительной полосе и на тротуаре, определяют две зоны, при пересечении которых автомобилем, система регистрирует вхождение и выводит на экран сообщение о нарушении.



Рис. 1. Внешний вид модели транспортной системы

Алгоритм, контролирующий движение по полосе, изображен на рис. 2.

Светофор, безусловно, является неотъемлемой частью системы дорожного движения, поэтому реализовать его правильную работу также крайне необходимо.

Стандартный дорожный светофор Т.1 работает по заданному алгоритму. Сначала горит красный, затем загорается желтый, при этом красный остается гореть, далее красный и желтый

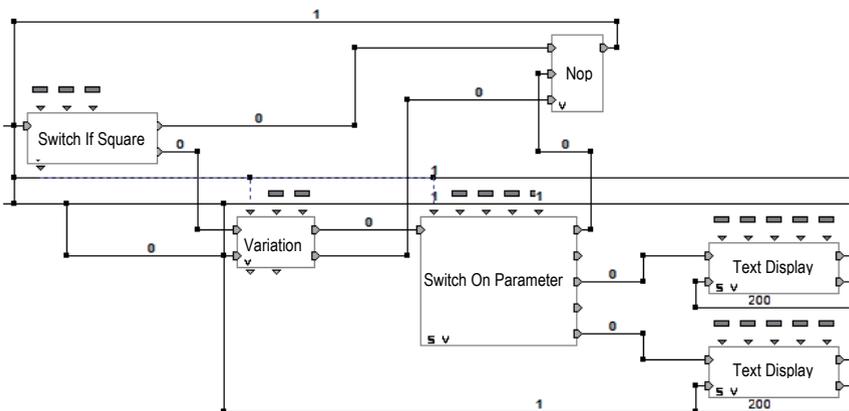


Рис. 2. Алгоритм контроля движения

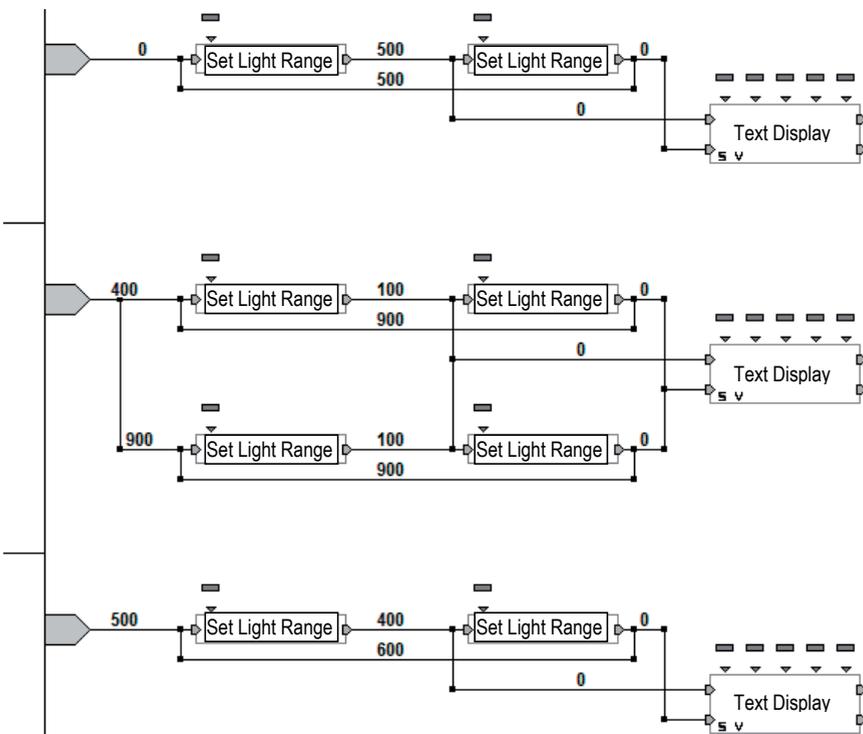


Рис. 3. Алгоритм работы светофора

гаснут, и загорается зеленый. В конце своего цикла, зеленый два раза мигает, гаснет, и загорается один желтый, затем один красный, и цикл повторяется.

Алгоритм работы светофора в программе Virtools показан на рис. 3.

Для каждого цвета используется свой собственный сценарий. Такая организация упрощает настройку временных интервалов и повышает наглядность.

После запуска программы красный свет загорается сразу, для желтого и зеленого запускаются таймеры. Через установленное время срабатывает второй блок Set Light Range, красный свет гаснет, и запускается таймер на повторный проход цикла.

Желтый и зеленый сигналы управляются таким же образом, но в случае с желтым, используется не два, а четыре блока с разными задержками. Они отвечают за включение желтого вместе с красным и желтого после зеленого.

Поскольку светофоры всегда ставятся парами, второй светофор работает по такому же алгоритму, но начинает цикл с зеленого.

Проезд перекрестка должен осуществляться на зеленый сигнал светофора, в противном случае система должна выдавать сообщение о нарушении правил.

Нарушение при проезде перекрестка фиксируется при выполнении двух условий:

- первое – автомобиль пересекает сетку перекрестка;
- второе – на светофоре в момент пересечения сетки горит красный свет.

Важно, чтобы эти условия выполнялись одновременно.

Так как в данном случае требуется информация о текущем состоянии источника света светофора, а соединять напрямую блоки, находящиеся в разных сценариях нельзя, то можно воспользоваться системными сообщениями.

Принцип работы системных сообщений в Virtools показан на рис. 4.

При получении входного сигнала, блок «Send Message» отправляет указанное сообщение заданному 3D объекту или сущности. Блок «Wait Message» располагается в сценарии того объекта, которому было послано сообщение и «слушает». При получении верного сообщения, блок посылает выходной сигнал.

Блок «Switch On Message» получает одно или несколько сообщений и активирует соответствующие выходы.

С выхода блока Switch On Message снимается первый сигнал, означающий, что горит красный свет. Получив этот сигнал, сетка на перекрестке меняет свой цвет. Эта же сетка регистрирует выезд автомобиля на перекресток.

Пакет разработки Virtools предоставляет широкие возможности по созданию интерактивных трехмерных сцен и даже сложных виртуальных миров. Использование структуры строительных блоков делает процесс создания сценариев проще и нагляднее, чем при написании кода вручную.

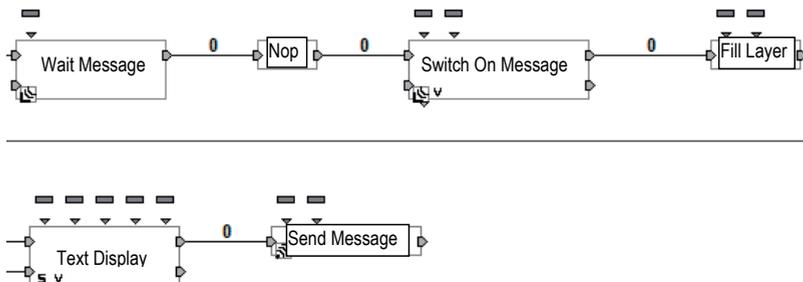


Рис. 4. Принцип работы системных сообщений

Используя систему сеток, удалось реализовать модель транспортной системы, реагирующую на перемещение персонажа пользователя и контролирующую соблюдение правил дорожного движения.

Библиографический список

1. Русскоязычный форум Virtools (электронный ресурс). Режим доступа: <http://virtools.ve-group.ru/> свободный.
2. Королев Сергей «Основы Virtools» 2-я редакция, 2008 г.
3. Официальный сайт 3DVIA Virtools (электронный ресурс). Режим доступа: <http://www.3ds.com/products/3dvia/3dvia-virtools/> свободный.
4. Официальный сайт Havok (электронный ресурс). Режим доступа: <http://www.havok.com/> свободный.
5. Официальный сайт Autodesk, разработчика 3ds Max (электронный ресурс). Режим доступа: <http://usa.autodesk.com/> свободный.

УДК 629.735.33

К. П. Васильева – студентка кафедры моделирования вычислительных и электронных систем

В. И. Исаков (канд. техн. наук, доц.) – научный руководитель

СЖАТИЕ И ВОССТАНОВЛЕНИЕ ИНФОРМАЦИИ В СИСТЕМАХ ОБРАБОТКИ БИМЕДИЦИНСКИХ СИГНАЛОВ

В настоящее время в медицинской диагностике, в задачах комплексной оценки функционального состояния человека в системах управления, в исследованиях интеллектуальной деятельности широко применяются электронные устройства различного назначения. К числу таких устройств относятся электрофизиологические приборы: кардиографы, электроэнцефалографы, электромиографы, электрокардиографы и т. д. [1–3].

Данные биомедицинских сигналов, в частности электроэнцефалографических, могут занимать довольно значительные объемы памяти (например, данные суточного электроэнцефалографического исследования занимают порядка 1 Гб памяти). Вследствие этого проблема хранения и передачи таких объемов данных является актуальной. Одним из путей решения этих проблем служит программная компрессия данных.

Для представления биомедицинских сигналов в сжатом виде возможно использование различных алгоритмов сжатия. Большинство сигналов имеет сложные частотно-временные характеристики. Как правило, такие сигналы состоят из близких по времени, короткоживущих высокочастотных компонент и долговременных, близких по частоте низкочастотных компонент. Для анализа таких сигналов нужен метод, способный обеспечить хорошее разрешение и по частоте, и по времени. Первое требуется для локализации низкочастотных составляющих, второе – для разрешения компонент высокой частоты. Существуют различные методы сжатия: с потерями (методы неискажающего сжатия) и без потерь (методы искажающего сжатия). [4, 5] Причина использования методов сжатия с потерями заключается в том, что они обеспечивают большую степень сжатия, чем методы сжатия без потерь. В этом случае учитываются особенности систем восприятия человека. Одним из наиболее перспективных и уже широко применяемых направлений здесь является сжатие на базе вейвлет-преобразования [5–8].

Благодаря хорошей приспособленности к анализу нестационарных сигналов вейвлет-преобразование стало мощной альтернативой преобразованию Фурье в ряде медицин-

ских приложений. Так как многие медицинские сигналы нестационарны, вейвлетные методы используются для сжатия информации с минимальными потерями диагностической информации.

Огромным достоинством вейвлет-преобразования является то, что оно не вносит дополнительной избыточности в исходные данные, и сигнал может быть полностью восстановлен с использованием тех же самых фильтров. Кроме того, отделение в результате преобразования деталей от основного сигнала позволяет очень просто реализовать сжатие с потерями – достаточно просто отбросить детали на тех масштабах, где они несут незначительную информацию (несущественность заключается в первую очередь в нарушении самых важных с точки зрения восприятия человеком характеристик сигнала).

Основной смысл сжатия заключается в следующем: сначала к сигналу применяется вейвлет-преобразование для разложения исходного сигнала на детализирующие и аппроксимирующие коэффициенты, а затем из данных преобразованного сигнала удаляются некоторые коэффициенты меньше порогового значения, (которое может задаваться в ручном режиме и в автоматическом), а оставшиеся коэффициенты кодируются с помощью какого-либо метода сжатия. Сжатый сигнал восстанавливается путем обратного вейвлет-преобразования с использованием модифицированных wavelet-коэффициентов. Схема алгоритма сжатия с использованием вейвлет-преобразования представлена на рис. 1.

Здесь H_1 (G_1) – высокочастотный фильтр анализа (синтеза), а H_2 (G_2) – низкочастотный фильтр анализа (синтеза).

Для программной реализации алгоритма используется пакет Wavelet Toolbox системы MatLab. Он обеспечивает работу с более чем пятьюдесятью различными типами вейвлетов [9, 10]. Необоснованное применение того или иного типа вейвлета способно привести к существенным потерям информации и временным затратам. И, напротив, удачный выбор может существенно повысить эффективность решаемой задачи.

На основании алгоритма вейвлетного сжатия информации разработана программа, которая позволяет производить выбор типа вейвлета и уровень разложения сигнала, загрузку сигналов электроэнцефалограмм из файла, выбор критериев оценки качества сжатия и восстановления сигналов, сжатие и восстановление сигналов, выбор оптимальных параметров по выбранным критериям, вывод данных в табличном и графическом виде. Схема алгоритма разработанной программы представлена на рис. 2.

В ходе работы было проанализировано качество вейвлетного сжатия и восстановления сигналов на основе выбранных критериев и определен наилучший вейвлет – биортогональный парный вейвлет с компактным носителем и уровень разложения – 5. Результаты сжатия и восстановления сигнала представлены на рис. 3.

Рассмотренные в данной работе методические и алгоритмические особенности компрессии биомедицинских сигналов позволяют обеспечивать высокую точность, сравнительно высокий уровень компрессии и характеризуются низкой вычислительной трудоемкостью (по сравнению с фрактальным сжатием).

Результаты работы могут быть использованы как в системах медицинской диагностики функционирования головного мозга, так и в системах исследования и моделирования интеллектуальной деятельности человека.

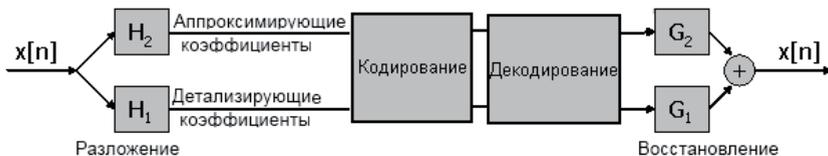


Рис. 1. Схема алгоритма сжатия с использованием вейвлет-преобразования

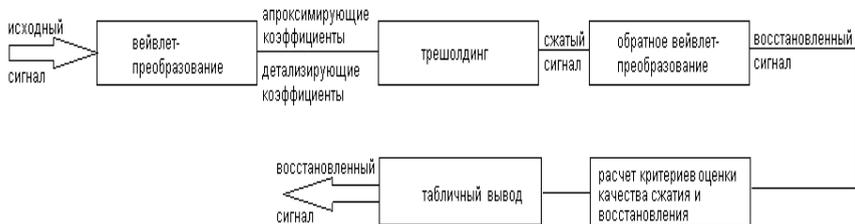


Рис. 2. Схема алгоритма разработанной программы

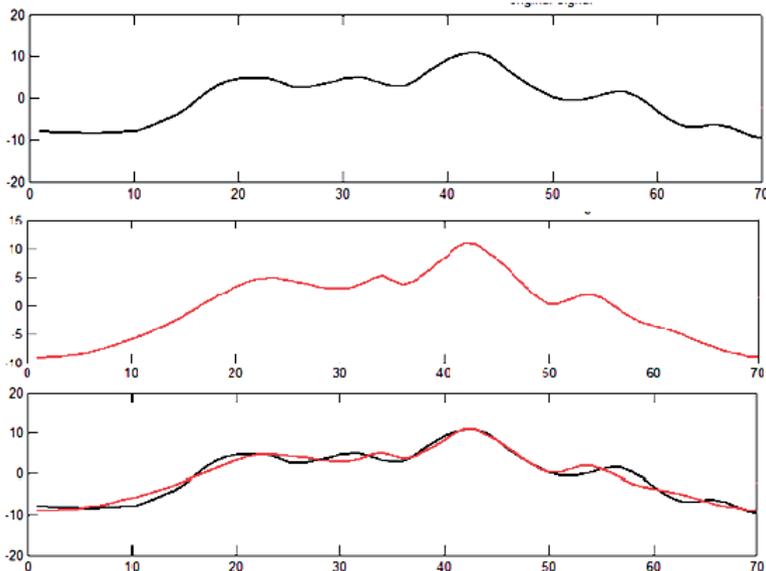


Рис. 3. Результаты сжатия и восстановления сигнала

Библиографический список

1. Агаханян Т. М., Никитаев В. Г. Электронные устройства в медицинских приборах. БИНОМ. Лаборатория знаний, 2005.
2. Гусельников В. И. Электрофизиология головного мозга. М.: Высшая школа, 1976.
3. Жаринов И. О., Трофимова А. А. Исследование методов и разработка программных средств быстрой сегментации ЭЭГ-сигнала / Сборник трудов 4-ой Санкт-Петербургской Ассамблеи молодых ученых и специалистов. СПб.: ГУ, 1999. С. 55.
4. Земцов А. Н. Разработка спектральных методов компрессии триангуляционных моделей на основе дискретного вейвлет-преобразования, 2005.
5. Кобелев В. Ю. Поиск оптимальных вейвлетов для сжатия цифровых сигналов / Сб. тез. докл. Науч.-техн. конф. «Современные проблемы естествознания. Физика». Ярославль, 1999.

6. Астафьева Н. М. Вейвлет-анализ: Основы теории и примеры применения. – Успехи физических наук, Т. 166, № 11, 1996. С. 1145–1170.
7. Добеши И. Десять лекций по вейвлетам, перевод. Ижевск: НИЦ Регулярная и хаотическая динамика, 2001. 464 с.
8. Яковлев А. Н. Введение в вейвлет-преобразования: учебн. пособие. Новосибирск: НГТУ, 2003. 104 с.
9. Смоленцев Н. К. Основы теории Вейвлетов. Вейвлеты в Matlab. ДМК, 2005. 304 с.
10. Дьяконов В. П. Matlab 6.5 SP1/7 + Simulink 5/6. Обработка сигналов и проектирование фильтров. М.: СОЛОН-Пресс, 2005. 576 с.

УДК 004.422.8

Е. С. Дороненкова – магистрант кафедры моделирования вычислительных и электронных систем

Г. С. Евсеев (канд. техн. наук, доц.) – научный руководитель

ОПТИМИЗАЦИЯ СТРУКТУРЫ ИНФОРМАЦИОННОЙ СИСТЕМЫ ДЛЯ ПСИХОЛОГИЧЕСКОГО ЦЕНТРА

Информационная система психологического центра должна обеспечивать оптимальную работу всех сотрудников и решать следующие задачи:

- обеспечивать надежное хранение и накопление необходимой информации, а также осуществлять информационно-справочные функции;
- осуществлять проведение психологических тестирований и исследований;
- освободить пользователя от трудоемких рутинных операций;
- обрабатывать результаты тестирования, а также увеличивать оперативность предоставления результатов диагностики;
- проводить дальнейшие математические исследования полученных результатов тестирования;
- охватывать все структурные подразделения психологического центра.

Непосредственными пользователями данной ИС являются:

- психологи;
- администратор центра;
- сотрудники отдела кадров;
- сотрудники отдела тренингов;
- гость (клиенты и т.п., обращающиеся за справочной информацией).

В общем виде структурная схема информационной системы представлена на рис. 1.

Оптимизации подлежит та часть системы, которая будет наращиваться с увеличением психологического центра. При развитии центра на работу будут приниматься новые специалисты психологи, а штат вспомогательных подразделений (таких как отдел кадров или администрация) будет оставаться прежним. Увеличение информационной системы психологического центра заключается в увеличении числа рабочих мест психологов. Исходя из этого, можно сделать вывод, что оптимизации должна подлежать часть системы, характеризующая работу специалистов-психологов, она представлена на рис. 2.

Психологи после проведения тестирования отправляют анкету с данными на статистическую обработку. Результаты обработки отправляются на SQL сервер. После записи данных, психолог отправляет на сервер запрос на получение результатов статистической обработки.

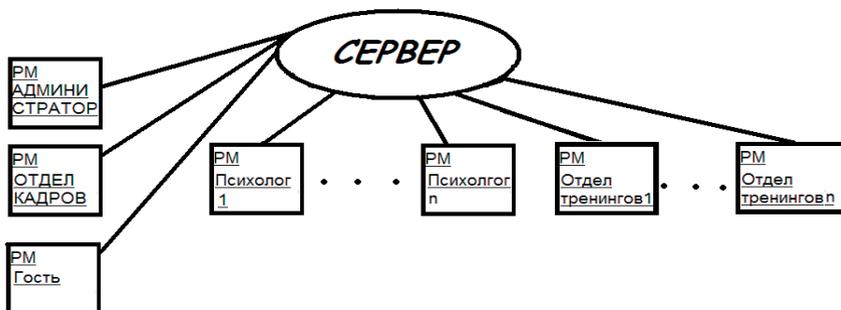


Рис. 1. Структурная схема информационной системы

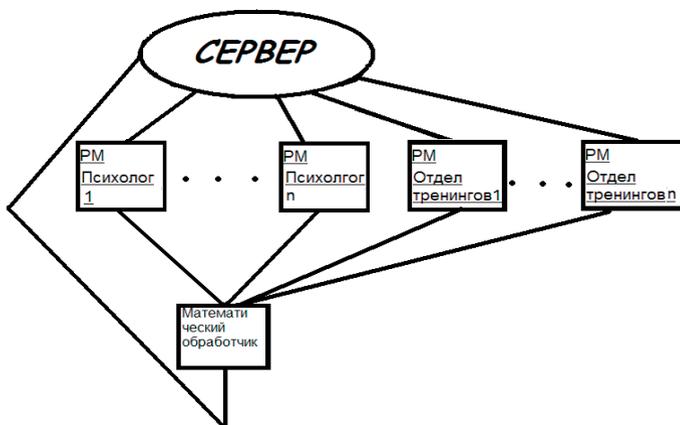


Рис. 2. Часть структурной схемы информационной системы, подлежащий оптимизации

Для оптимизации информационной системы психологического центра значимыми являются следующие характеристики:

1. Объем информации, передаваемый по локальной сети (информация, передаваемая между компьютерами).
2. Среднее число запросов к серверу, отправляемых за единицу времени.
3. Оперативность ответа (оперативность получения запрашиваемой информации).

За единицу объема, передаваемой информации передаваемой по локальной сети, берется одна анкета. Каждая анкета содержит результаты проведенного одного тестирования одного клиента и является исходными данными для статистической обработки. Отправляемые запросы на обработку могут быть как одноанкетными, так и многоанкетными. Многоанкетные запросы формируются из нескольких анкет одного клиента одного и того же тестирования проведенного несколько раз. Подобные запросы используются, например, для того, чтоб с помощью статистической обработки определить изменения психологических черт испытуемых или определить эффективности сеансов психологического консультирования. Помимо этого многоанкетные запросы могут формироваться

из нескольких анкет разных клиентов. Такие запросы используются для определения, с помощью статистической обработки, эффективности различных тренингов.

В небольшом психологическом центре штат психологов консультантов составляет 15 человек, а штат психологов тренеров – 7 человек. В результате общий объем, передаваемый по локальной сети информации, в среднем составляет 10050 анкет в сутки (в рабочее время).

При развитии психологического центра и увеличении штата сотрудником увеличивается и объем передаваемый по локальной сети информации. При увеличении штата до 45 психологов консультантов и до 20 психологов тренеров, объем передаваемой информации увеличился до 28950 анкет (табл. 1).

Таблица 1

Характеристики потока запросов

	Сотрудник	Количество сотрудников	Тип запроса	Количество анкет в запросе	Частота запросов на обработку от одного сотрудника	Общее количество анкет
До увеличения штата	Психолог-консультант	15	Одно-анкетный	1	30 запросов	10050
			Много-анкетный	2	40 запросов	
	Психолог-тренер	7	Много-анкетный	20	20 запросов	
			Много-анкетный	40	20 запросов	
После увеличение штата	Психолог-консультант	45	Одно-анкетный	1	30 запросов	28950
			Много-анкетный	2	40 запросов	
	Психолог-тренер	20	Много-анкетный	20	20 запросов	
			Много-анкетный	40	20 запросов	

После статистической обработки данных и записи результатов на сервер, психологи-консультанты и психологи-тренеры посылают запросы (запросы на выборку) на SQL сервер для того, чтобы получить результаты статистической обработки. Помимо запросов на выборку психологи посылают запросы на редактирование и на добавление.

Третьей важной характеристикой информационной системы психологического центра является оперативность ответа (оперативность получение запрашиваемых данных с SQL сервера).

Время на получение запрашиваемой информации с сервера складывается из следующих компонентов:

1. Псылка запроса на статистическую обработку (время передачи запроса, а также время нахождения запроса в очереди на статистическую обработку).
2. Обработка запроса (время обработки запроса статистическим обработчиком).
3. Псылка результатов на SQL сервер (время передачи данных, а так же время на запись данных на сервер).
4. Получение результатов с SQL сервера (время нахождения запроса на получение результатов в очереди на обработку, а также время обработки запроса).

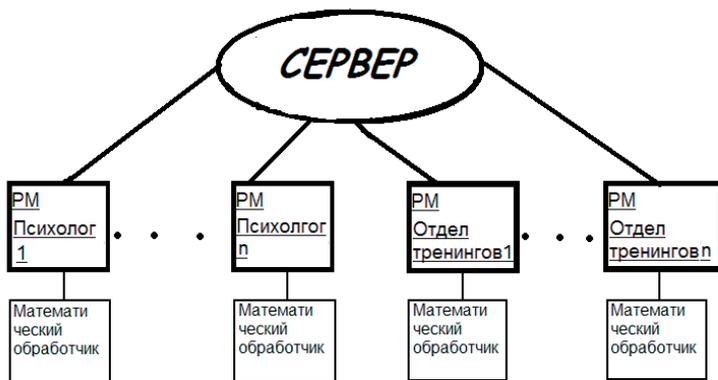


Рис. 3. Оптимизированная структурная схема

Цель оптимизации информационной системы психологического центра заключается в том, чтобы модернизировать ту часть системы, которая будет наращиваться, так чтобы характеристики всей информационной системы при увеличении ее не ухудшались.

С достижения цели оптимизации работы системы на каждом рабочем месте устанавливается свое СУБД и свой математический обработчик данных.

На рис. 3 представлена структурная схема разработанной информационной системы психологического центра с учетом проделанных модификаций:

Данные преобразования структурной схемы позволяют вести независимую и параллельную статистическую обработку данных отдельно на каждом рабочем месте.

За счет такой модернизации информационной системы возрастает оперативность получения данных о результатах статистической обработки. Время на получение ответа определяется только временем самой статистической обработки. При увеличении числа рабочих мест оперативность получения информации останется прежней, поскольку статистическая обработка ведется независимо на каждом рабочем месте.

Также при расширении информационной системы почти не возрастет нагрузка на локальную сеть за счет установки СУБД и математического обработчика данных на каждом рабочем месте.

Запись на SQL сервер накопленной за рабочую смену информации можно делать один раз в конце рабочего дня, в то время когда нагрузка на локальную сеть минимальна.

В результате оптимизации информационной системы психологического центра были достигнуты следующие результаты:

1. Разгрузка локальной сети от передаваемой по ней информации.
2. Оперативность получения запрашиваемой психологами информации.
3. При увеличении числа рабочих мест характеристики всей информационной системы не ухудшаются.

Библиографический список

1. Карпова, Т.С. Базы данных: модели, разработка, реализация / Т. С. Карпова. СПб.: Питер, 2001. 304 с.
2. Преснякова, Г.С. Проектирование интегральных реляционных баз данных / Г. С. Преснякова. СПб.: Петроглиф, 2007. 224 с.
3. Дейт, К. Дж. Введение в системы баз данных / К. Дж. Дейт. Вильямс, 2006. 1328 с.

Г. А. Емцов – магистрант кафедры вычислительных систем и сетей

В. А. Галанина (канд. техн. наук, доц.) – научный руководитель

ПРОБЛЕМА ПОИСКА АНАЛОГОВ ПРИ АВТОМАТИЗАЦИИ ПРОЕКТИРОВАНИЯ

Формализация процесса проектирования (ПП) на начальных этапах, особенно на этапе согласования технического задания (ТЗ), встречает серьезные затруднения. При разработке сложных технических объектов, таких как гироскопические системы, системы автоматического управления и др. возникают проблемы с выбором наилучшей структуры, оптимальной организации взаимодействия элементов, определением требуемых режимов функционирования как объекта в целом, так и его составных частей. При проектировании подобных объектов широкое распространение получил иерархический подход к их описанию. Объект проектирования (ОП) расчленяется на иерархические уровни, каждому уровню соответствуют свои цели проектирования. Исходя из этого, можно сформировать иерархию – «дерево» целей проектирования. Цели, совместно с критериями, ограничениями и условиями эксплуатации, отражаются в ТЗ на проектирование. Дерево целей трансформируется в дерево ТЗ.

Процесс поиска аналога объекта проектирования можно разбить на два этапа. Первый – это непосредственно распознавание объекта проектирования и выборка всех аналогов, и второй – ранжирование аналогов по и определению наиболее близкого аналога.

Различные авторы дают различную типологию методов распознавания образов. Д. А. Поспелов (1990) выделяет два основных способа представления знаний:

1. Интенциональное представление – в виде схемы связей между атрибутами (признаками).
2. Экстенциональное представление – с помощью конкретных фактов (объекты, примеры).

Ни один из этих методов, взятый отдельно от другого, не позволяет сформировать адекватное отражение Реальности. Перспективные системы распознавания должны обеспечивать реализацию обоих этих методов, а не только какого-либо одного из них.

Интенциональные методы

Методы, основанные на оценках плотностей распределения значений признаков

Объекты исследования рассматриваются как реализации многомерной случайной величины, распределённой в пространстве признаков по какому либо закону. Данные методы сводятся к определению отношения правдоподобия в различных областях многомерного пространства признаков.

Методы, основанные на предположениях о классе решающих функций

В данной группе методов считается известным общий вид решающей функции и задан функционал её качества. На основании этого функционала по обучающей последовательности ищется наилучшее приближение решающей функции. Основным достоинством методов, является ясность математической постановки задачи распознавания, как задачи поиска экстремума.

Логические методы

Логические методы распознавания образов базируются на аппарате алгебры логики и позволяют оперировать информацией, заключённой не только в отдельных признаках, но и в сочетаниях значений признаков. В этих методах значения какого-либо признака рассматриваются как элементарные события.

Лингвистические методы

Лингвистические методы распознавания образов основаны на использовании специальных грамматик порождающих языки, с помощью которых может описываться совокупность свойств распознаваемых объектов. Каждый объект представляется со-

вокупностью непроектируемых элементов, «соединённых» между собой теми или иными способами, «предложением» некоторого «языка». Путём синтаксического анализа «предложения» устанавливается его синтаксическая «правильность».

Экстенциональные методы

Метод сравнения с прототипом

Для классификации неизвестного объекта находится ближайший к нему прототип, и объект относится к тому же классу, что и этот прототип. Очевидно, никаких обобщённых образов классов в данном методе не формируется. На практике, перед исследователем, встают две проблемы: во-первых, выбор меры близости (метрики), от которой может существенно измениться пространственная конфигурация распределения объектов, во-вторых, анализ многомерных структур экспериментальных данных при условии многомерности пространства признаков.

Метод k-ближайших соседей

При классификации неизвестного объекта заданное число (k) геометрически ближайших к нему в пространстве признаков других объектов (ближайших соседей) с уже известной принадлежностью к распознаваемым классам. Решение об отнесении объекта принимается путём анализа информации его ближайших соседей. Чрезвычайно обостряется проблема выбора метрики в условиях высокой размерности пространства признаков вследствие достаточной трудоёмкости данного метода, которая становится значимой даже для высокопроизводительных компьютеров.

Алгоритмы вычисления оценок (голосования)

Принцип действия алгоритмов вычисления оценок (АВО) состоит в вычислении приоритета (оценок сходства), характеризующих «близость» распознаваемого и эталонного объектов по системе ансамблей признаков, представляющей собой систему подмножеств заданного множества признаков. Близость объектов рассматривается только в одном заданном пространстве признаков.

Для распознавания ОП будем использовать Лингвистический метод, т. к. во-первых данный метод реализуется, как синтаксический анализатор, что не требует больших машинных мощностей, при этом правила «языка» могут храниться буквально в «текстовом» виде.

Во-вторых, поскольку объекты проектирования сложны, и число требуемых признаков велико, это делает привлекательней идею описания сложного объекта в виде иерархической структуры более простых объектов. Так же данный подход соответствует представлению ТЗ в виде дерева ТЗ, что позволяет на каждом уровне иерархии формулировать и решать задачи приемлемой сложности.

Для того чтобы описать образ при помощи более простых подобразов, а каждый подобраз снова описывать ещё более простыми подобразами и т. д., был предложен синтаксический, или структурный, подход. Этот подход основан на аналогии между структурой образов и синтаксисом языков. В рамках синтаксического подхода считается, что образы строятся из соединённых различными способами подобразов, так же как фразы и предложения строятся путём соединения слов, а слова соединяются из букв. Очевидно, что такой подход полезен только в том случае, когда распознать выбранные простейшие подобразы, называемые непроектируемыми элементами, легче, чем сами образы.

Для представления структурной информации образа используется «граф отношений». Использование графа отношений для описания образов позволяет расширить класс допустимых отношений, включив в него любое отношение, которое удобно определяется из образа. Заметим, что, во-первых, конкатенация – единственная естественная операция для одномерных языков, и во-вторых, что граф, содержит циклы. Поэтому при помощи графа можно выразить более богатые описания.

Основной единице структурного представления знаний является семантический граф. Он может служить для представления текущего состояния системы, множества состояний, посылок структурного вывода и т. д.

Чтобы определить на сколько аналог близок к параметрам ОП указанным в ТЗ, используем понятие нечёткого множества, что позволит «размыть» границы параметров. Основываясь на экспертных оценках, можно будет судить, насколько аналог близок.

Под нечётким множеством A понимается совокупность:

$$A = \{(x, \mu_A(x)) \mid x \in X\},$$

где X – универсальное множество, а $\mu_A(x)$ – функция принадлежности, характеризующая степень принадлежности элемента x нечёткому множеству A .

Функция $\mu_A(x)$ принимает значения в некотором вполне упорядоченном множестве M . Множество M называют множеством принадлежностей, часто в качестве M выбирается отрезок $[0;1]$.

Лингвистическая переменная – переменная, которая может принимать значения фраз из естественного или искусственного языка. Например, лингвистическая переменная «скорость» может иметь значения «высокая», «средняя», «очень низкая» и т. д. Фразы, значение которых принимает переменная, в свою очередь являются именами нечётких переменных и описываются нечетким множеством.

При использовании лингвистической переменной, задаваемой на непрерывном носителе (интервале действительных чисел), возникает задача хранения функций принадлежности в памяти ЭВМ. Одним из возможных способов хранения, является представление функции принадлежности в виде стандартной π – функции $\mu_A(x) = \pi(x, \eta, x_1)$, которая имеет следующий вид (формулы (1), (2), рис. 1):

$$\pi(x, \eta, x_1) = \begin{cases} S(x, x_1 - \eta, x_1 - \eta/2, x_1) & \text{при } x \leq x_1 \\ 1 - S(x, x_1 - \eta, x_1 - \eta/2, x_1) & \text{при } x > x_1 \end{cases} \quad (1)$$

$$S(x, \varepsilon, t, x_1) = \begin{cases} 0 & \text{при } x \leq \varepsilon \\ 2(x_1 - x)^2 / (x - \varepsilon)^2 & \text{при } \varepsilon < x \leq t \\ 1 - 2(x_1 - x)^2 / (x - \varepsilon)^2 & \text{при } t < x \leq x_1 \\ 1 & \text{при } x > x_1 \end{cases} \quad (2)$$

Таким образом, функция принадлежности может быть представлена в памяти ЭВМ двумя параметрами: x_1 – величиной, при которой значение функции равно 1; η – величиной, при которой значение функции равно 0,5.

Определим лингвистическую переменную β_i , имеющую имя « i -й параметр». Эта переменная имеет два базовых значения $T = \{\alpha_{i1}, \alpha_{i2}\} = \{\text{минимальное значение } i\text{-го параметра, максимальное значение } i\text{-го параметра}\}$. Пусть функции принадлежности μ_{i1} и μ_{i2} , соответствующие базовым значениям лингвистической переменной β_i , изображённые на рис. 2, представляются в виде следующих π – функций:

$$\mu_{i1}(x) = \pi(x, \lambda_{i\min}, x_{i\min}), \quad (3)$$

$$\mu_{i2}(x) = \pi(x, \lambda_{i\max}, x_{i\max}), \quad (4)$$

$$\mu(x) = 1 / ((x_i - x) / \lambda_i)^2, \quad (5)$$

Как видно из рис. 2, эти две функции имеют различную форму, что определяется значением λ . Очевидно, что λ определяет «скорость» падения значения функции принадлежности.

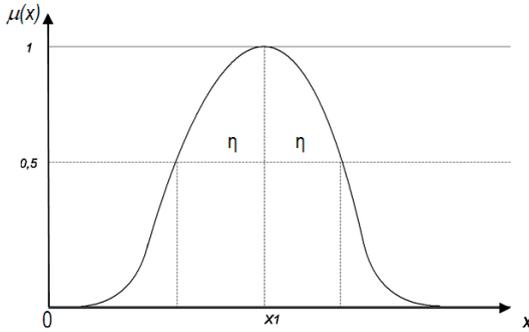


Рис. 1

$$\mu(x) = \frac{1}{\left(\frac{(x_1 - x)}{\lambda_1}\right)^2}$$

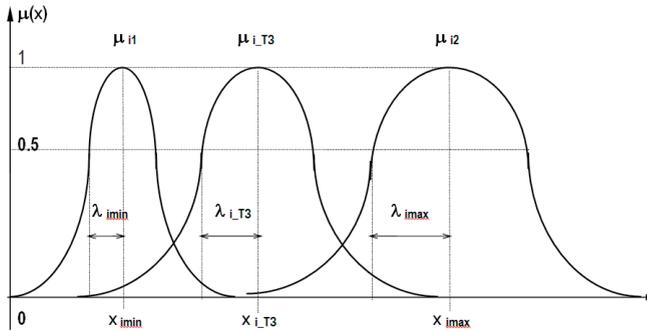


Рис. 2

Обозначим значение лингвистической переменной β_j , определяемое в соответствии с требованием ТЗ, как α_{i_T3} и определим соответствующее значение функции принадлежности μ_{i_T3} . Для этого воспользуемся формулами (4, 5, 6), и получим:

где

$$\mu_{i_T3} = \pi(x, \lambda_{i_T3}, x_{i_T3}),$$

$$\lambda_{i_T3} = \lambda_{imax} + (1 - \gamma) \lambda_{imin},$$

$$\gamma = (x_{imax} - x) / (x_{imax} - x_{imin}).$$

Далее, определим степень аналогичности j -го объекта по i -му параметру объекту проектирования следующим образом:

$$r(x_{ji} / x_{ji_T3}) = \pi(x_{ji}, \lambda_{ji_T3}, x_{ju_T3}).$$

Очевидно, что по каждому параметру из множества параметров x_{ji} могут быть заданы аналогичным образом свои функции принадлежности. Таким образом, имеем

множество $\mu_{i, T3}$ и соответственно множество $\{r_i(x_j/x_{j, T3})\}$, $i = 1..n$, где n – количество параметров, по которым производится сравнение.

Аналогичным образом можно определить подобное множество $\{r_j\}$ для остальных объектов, претендующих на роль аналога. В результате получим: $r_{11}, r_{12}, \dots, r_{1n}$ – степени аналогичности объекта 1 по 1-му, 2-му, ..., n -му параметрам; $r_{m1}, r_{m2}, \dots, r_{mn}$ – то же для объекта m .

Предположим, что мерой аналогичности j -го объекта по всем параметрам может служить среднее арифметическое значений r_{ji} , т. е.

$$r_j = \frac{\sum_{i=1}^n r_{ji}}{n}$$

Алгоритм определения аналога ОП при наличии нескольких претендентов на аналог включает следующие шаги.

1. Определение аргументов π – функций ($\lambda_{ji, T3}$) по каждому i -му параметру j -го претендента на аналог, значение которого не совпадает с требованиями ТЗ.
2. Определение степени аналогичности $r(x_{ji}/x_{ji, T3})$.
3. Определение среднего значения степеней аналогичности.
4. Ранжирование аналогов с использованием системы нечетких высказываний

УДК 20.23.17

Е. М. Ермолина – магистрант кафедры моделирования вычислительных и электронных систем

Г. С. Евсеев (канд. техн. наук, доц.) – научный руководитель

ОПТИМИЗАЦИЯ БАЗЫ ДАННЫХ ДЛЯ СЕТИ МЕДИЦИНСКИХ УЧРЕЖДЕНИЙ

При создании информационной системы существуют различные способы ее организации. Можно разместить ИС на одном компьютере, создать клиент-серверную систему либо воспользоваться возможностями распределенной информационной системы. У каждого способа организации есть свои плюсы и минусы, а, зачастую, какой-то вариант просто неприемлем для заданных масштабов системы. Ниже рассмотрены варианты организации информационной системы для сети медицинских учреждений.

Рассматриваемая сеть медицинских учреждений включает в себя:

- поликлиники, которые непосредственно оказывают медицинские услуги,
- единую лабораторию, которая производит медицинские анализы для всех поликлиник,
- муниципальный медицинский центр, который периодически требует от поликлиник предоставить некоторые статистические данные.

Запросы пользователей к информационной системе.

Сотрудники лаборатории – заносят в информационную систему результаты проведенных анализов.

Врачи – имеют доступ к историям болезней пациентов. Получают перечни записанных к ним больных.

Сотрудники отдела кадров – работают с личной информацией о сотрудниках.

Сотрудники регистратуры – заносят в информационную систему личные данные пациентов, производят запись на приемы к врачам.

Пациенты – могут ознакомиться со своей историей болезни, просматривать информацию о наличии свободного времени у врачей.

Муниципальный центр – запрашивает статистические данные о сотрудниках поликлиник, об уровне их загруженности и о заболеваемости в регионе.

1. Локальная информационная система (на одном компьютере)

Решение, состоящее в размещении такой системы на одном компьютере, противоречит здравому смыслу, ибо задействовано несколько организаций, которые не могут располагаться в одном помещении.

2. Клиент-серверная система

Создание клиент-серверной системы предполагает наличие одного сервера, на котором размещены все таблицы базы данных, и некоторого количества компьютеров-пользователей, обладающих своим собственным интерфейсом.

В данном случае сервер, на котором и будет располагаться вся информация, предполагается расположить в муниципальном центре – районном учреждении, занимающемся сбором статистической информации. Каждая организация, в свою очередь, будет снабжена интерфейсом, позволяющим получить с сервера все необходимые данные (рис. 1).

Ощутимым минусом такой системы будет значительная загрузка линий связи. Следствие этого – длительное время выполнения запросов, поскольку все необходимые данные каждый раз должны передаваться с сервера на компьютер-клиент. Поток запросов для такой системы представлены ниже (табл. 1) на примере запросов пользователя «Врач».

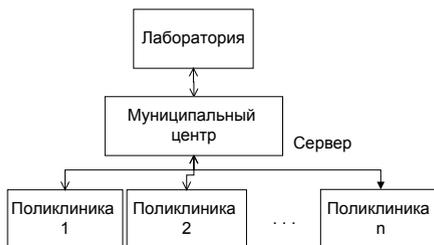


Рис. 1. Клиент-серверная система

Таблица 1

Потоки запросов пользователя «Врач» для клиент-серверной системы

Тип запроса	Таблица(ы)	Распределение во времени	Частота запросов в час (в среднем)
Выборка	Исследование	10:00–20:00	12 000
Выборка	Результаты	10:00–20:00	12 000
Добавление	Карточка	10:00–20:00	12 000
Редактирование	Карточка	10:00–20:00	10
Выборка	Карточка	10:00–20:00	6 000
Выборка	Номерки	10:00–20:00	12 000

3. Распределенная система

Следующий вариант – это использование коллектива серверов, то есть, распределенной системы. Если в каждом учреждении разместить свой сервер базы данных, содержащий основную информацию, относящуюся к данной поликлинике, лаборатории и т. д., это позволит значительно разгрузить линии связи и уменьшить время выполнения запросов. Теперь по сети будут передаваться данные между серверами только в отдельных случаях: необходимо запросить у лаборатории результаты анализов пациен-

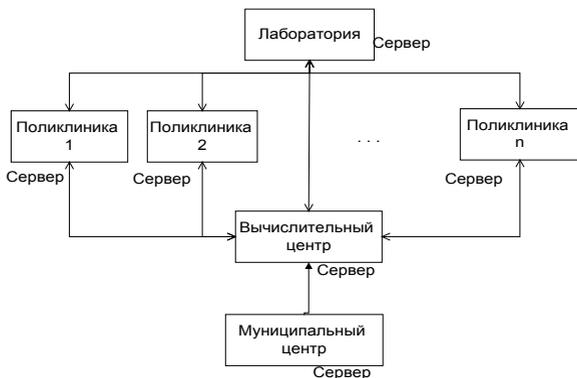


Рис. 2. Распределенная система

та или пациент одной поликлиники обратился в другую, и нужно передать его данные и карточку. Дополнительно вводится сервер вычислительного центра. Функции этого центра заключаются в сборе данных с поликлиник, в обработке их для отчетов в муниципальный центр, а также в обмене данными между поликлиниками. На этот сервер с какой-то периодичностью, допустим, раз в неделю, списываются все новые данные из всех поликлиник и лаборатории. Таким образом, с сервером вышестоящей инстанции общается только сервер вычислительного центра, а не серверы всех поликлиник. Такая организация информационной системы (рис. 2) обеспечивает значительную экономию средств связи, кроме того, расположение вычислительного центра можно выбрать с учетом стоимости связи между поликлиниками. А, в настоящее время, дешевле купить мощный компьютер, чем организовать протяженные связи между серверами. Потoki запросов при таком построении системы уменьшатся (табл. 2).

Таблица 2

Потоки запросов пользователя «Врач» для распределенной системы

Тип запроса	Таблица(ы)	Распределение во времени	Частота запросов в час (в среднем)
Выборка	Исследование	10:00–20:00	120
Выборка	Результаты	10:00–20:00	120
Добавление	Карточка	10:00–20:00	120
Редактирование	Карточка	10:00–20:00	0,1
Выборка	Карточка	10:00–20:00	60
Выборка	Номерки	10:00–20:00	120

4. Коллектив серверов внутри каждой поликлиники

Создание коллектива серверов внутри поликлиники является расширением идеи создания распределенной системы. Усовершенствование состоит в создании дополнительно отдельных серверов внутри поликлиники в каждом кабинете врача. При такой организации работы специалисту нет необходимости отправлять запрос на сервер при осмотре каждого нового пациента, чтобы получить доступ к его карточке. На основании

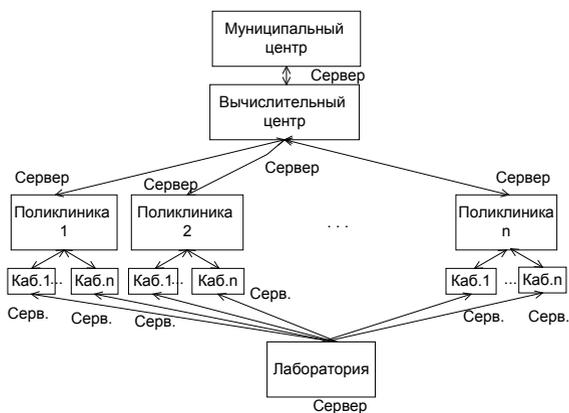


Рис. 3. Коллектив серверов внутри каждой поликлиники

выданных номерков на заданный день делается заранее выборка карточек пациентов для каждого кабинета. Она сохраняется в специальной временной таблице. Врач заносит в нее сведения по результатам приема пациентов, а в конце дня все новые данные за сутки переписываются на сервер поликлиники. Это позволяет сократить потоки запросов (табл. 3). Схема описанной организации системы (рис. 3) представлена ниже.

Таблица 3

Потоки запросов пользователя «Врач» для коллектива серверов

Тип запроса	Таблица(ы)	Распределение во времени	Частота запросов в час (в среднем)
Добавление	Карточка	10:00–20:00	2,5
Редактирование	Карточка	10:00–20:00	Отсутствуют
Выборка	Карточка	10:00–20:00	2,5
Выборка	Номерки	10:00–20:00	2,5

Такой подход позволяет значительно сократить нагрузку на общий сервер поликлиники, так как сервер каждого кабинета взаимодействует с ним только 2 раза в день, в определенное время, которое можно разнести, чтобы сделать загрузку сервера равномерной. Благодаря этому мощный серверный компьютер можно одновременно загрузить дополнительными системами (для отдела кадров, бухгалтерии и т. д.). А компьютеры на рабочих местах можно поставить более дешевые – достаточно, чтобы они поддерживали MS Office.

Библиографический список

1. Карпова, Т. С. Базы данных: модели, разработка, реализация / Т. С. Карпова. СПб.: Питер, 2001. 304 с.
2. Преснякова, Г. С. Проектирование интегральных реляционных баз данных / Г. С. Преснякова. СПб.: Петроглиф, 2007. 224 с.
3. Дейт, К. Дж. Введение в системы баз данных / К. Дж. Дейт. Вильямс, 2006. 1328 с.

УДК 616.831

Н. Ю. Зуева – студентка кафедры моделирование вычислительных и электронных систем

О. О. Жаринов (канд. техн. наук, доц.) – научный руководитель

МОДЕЛИРОВАНИЕ АЛГОРИТМОВ ОБРАБОТКИ ЭЛЕКТРОЭНЦЕФАЛОГРАММ

Электроэнцефалограмма (ЭЭГ) – график электрической активности головного мозга, получаемый в процессе электроэнцефалографии. Одной из основных характеристик ЭЭГ является частота. В зависимости от частотного диапазона, но также и от амплитуды, формы волны различают ритмы ЭЭГ и нестационарные феномены [1].

При анализе ЭЭГ специалисту в первую очередь необходимо получать информацию о параметрах частотных ритмов. Наибольшее распространение в электроэнцефалографии получил метод спектрального анализа ЭЭГ. Вычисление спектров мощности (спектрограмм – СГ) позволяет получить энергию каждой из частотных составляющих данной ЭЭГ и оценить их соотношение. Это дает возможность сравнивать электрические процессы разных отделов коры на одном отрезке времени, проводить сравнение спектров каждой данной области в динамике при повторных исследованиях, а также сравнивать в количественных величинах ЭЭГ разных обследуемых.

Различают способы анализа в частотной и временной области. Первый заключается в использовании алгоритма быстрого преобразования Фурье (БПФ). Методы обработки сигнала во временной области используют построение автокорреляционной и кросскорреляционной функции ЭЭГ. Использование автокорреляционной функции (АКФ) так же, как и спектральный анализ позволяет осуществлять количественный анализ гармонических колебаний в случайном процессе. При этом позволяет вычислять числовые величины, характеризующие обрабатываемую ЭЭГ [2].

Для анализа сигналов ЭЭГ успешно применяются подходы на основе параметрических моделей авторегрессии-скользящего среднего (АРСС), дающих косвенное описание признаков ЭЭГ. АРСС анализ базируется на предположении, что текущее значение ЭЭГ имеют существенную статистическую связь с ее предысторией. Информацию о признаках классов сигналов ЭЭГ опосредованно заложена в коэффициентах АРСС модели [3].

Через параметры АРСС модели могут быть выражены оценки частотных и корреляционных функций, определены отрезки стационарности ЭЭГ, решены задачи сегментации и классификации ЭЭГ. В частности, при известных параметрах АРСС модели можно получить оценку спектральной плотности.

Для выделения нестационарных феноменов хорошо себя зарекомендовал вейвлет-анализ.

При обработке ЭЭГ традиционно принято решать задачи сегментации и классификации (по ритмам, по признакам нестационарных феноменов).

Для моделирования алгоритма обработки необходимо задавать модель не стационарного сигнала. Это легко реализовать на основе АРСС – моделей, с изменяемыми коэффициентами.

АРСС – модель квазистационарного участка i -ого класса ЭЭГ представляет значения дискретных отсчетов $y^{(i)}[n]$ посредством линейного соотношения вида:

$$y^{(i)}[n] + \sum_{k=1}^{p^{(i)}} a_k^{(i)} \times y^{(i)}[n - k] = b_0^{(i)} \times x[n] + \sum_{k=1}^{q^{(i)}} b_k^{(i)} \times x[n - k],$$

где $\{x[n]\}$ – последовательность независимых, нормально распределенных случайных величин с нулевым математическим ожиданием и единичной дисперсией [3].

Решение задачи сегментации можно осуществлять на основе метода спектрального анализа на скользящем окне. Полный временной интервал сигнала, особенно при большой его длительности, разделяется на подынтервалы – временные окна, и преобразование проводится последовательно для каждого окна в отдельности.

Оконное преобразование выполняется в соответствии с выражением:

$$S(f, \tau) = \sum_{n=1}^N w[n] e^{-j2\pi f n \Delta t} y[n + \tau].$$

Здесь функция $w(n)$ представляет собой функцию окна сдвига преобразования по координате n , где параметром τ задаются фиксированные значения сдвига. В качестве окна преобразования может использоваться как простейшее прямоугольное окно, так и специальные весовые окна (Хэмминга, Бартлетта, Гаусса, Кайзера), обеспечивающие малые искажения спектра за счет граничных условий вырезки оконных отрезков. При этом для каждого положения окна на временной оси сигнала вычисляется свой комплексный спектр.

Далше определяем, к какому типу ритма относится подынтервал. Для этого вычисляются индексы ритмов по формуле:

$$P_{\text{ритм}, \tau} = \frac{\sum_{f=f_n}^{f_k} |S(f, \tau)|^2}{\sum_{f=1}^{f_{\max}} |S(f, \tau)|^2}$$

где f_n и f_k – значения начальной и конечной частот, выбираемые из табл. 1.

Таблица 1

Частотные характеристики ритмов ЭЭГ

Ритм				Δ	
f_n	8	14	31	1	4
f_k	13	30	120	3	7

Определяем максимальный индекс ритма и присваиваем подынтервалу значение ритма.

Параметр τ выбирается так, чтобы подынтервал делился на 3 части, в итоге для каждого участка длины τ , окажется 3 в общем случае различных решений о типе ритма. Окончательное решение принимаем по мажоритарному принципу, если же все решения различны, тип ритма не определен, и принимается решение о наличие нестационарного феномена.

Так же за основу решения задачи сегментации, возможно, взять метод динамических эталонов. Метод динамических эталонов заключается в сравнение некоторого «эталонного» значения отсчета в каждый момент времени с реально наблюдаемым значением.

Эталонное значение для каждого $l = 1, 2, \dots, M$ класса ЭЭГ формируется на основе статического прогноза на l дискретов вперед, осуществляемого с использованием некоторого количества выборочных значений уже известной части реализации наблюдаемого класса. В каждый момент времени $y[n]$ динамический эталон определяет ожидаемое среднее значение процесса соответствующего класса. Отклонение фактически

наблюдаемого выборочного значения процесса $y(i)[n]$ от ожидаемого «эталонного» $\hat{y}^{(i)}[n]$ используется для выработки решения о соответствии наблюдаемой реализации ЭЭГ одному из классов [3].

Принято использовать прогноз не более чем на 1 дискретный отчет ($l = 1$), поскольку дисперсия ошибки прогноза при больших временных интервалах возрастает, и точность прогноза становится неудовлетворительной. Когда поступает действительное значение реализации $y[l]$, определяется отклонение $e^{(i)}[l] = y[l] - \hat{y}^{(i)}[l]$ спрогнозированного значения от действительного для каждого из M классов, и вычисляется квадратическая ошибка прогноза $E^{(i)}[l] = (e^{(i)}[l])^2$ [3]. Решение о принадлежности участка ЭЭГ к определенному классу принимается в пользу минимального значения $E^{(i)}[l]$. Окончательное решение принимается, также по мажоритарному принципу.

Моделирующее программное обеспечение обработки ЭЭГ реализовано на языке Java в среде разработки Eclipse.

Написанная программа обладает следующими возможностями:

1. Задавать модель нестационарной ЭЭГ, используя параметрический способ задания каждого сегмента ЭЭГ (параметры спектра и длительность интервалов) (рис. 1).

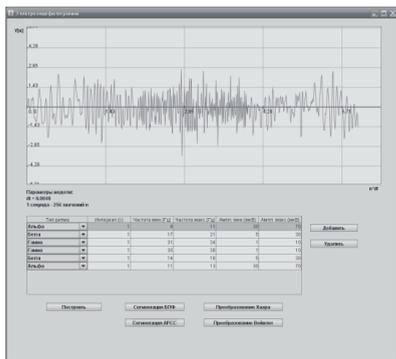


Рис. 1

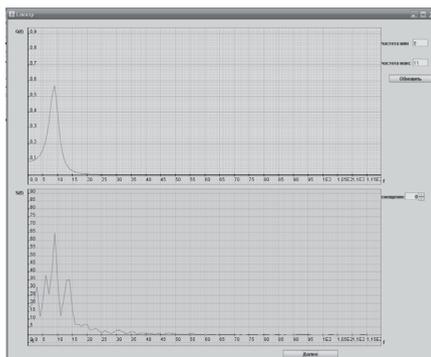


Рис. 2

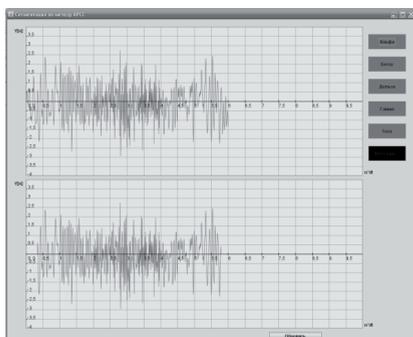


Рис. 3

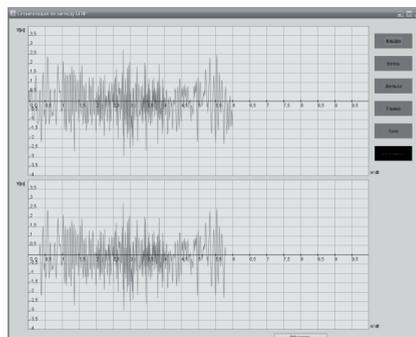


Рис. 4

2. Для каждого положения окна на временной оси сигнала вычислять оценку спектра мощности (рис. 2).

3. Реализовывать алгоритм спектрального анализа на скользящем окне и алгоритм динамических эталонов тоже на скользящем окне.

4. Позволяет вычислять оценку вероятности правильной сегментации для любого заданного пользователем нестационарного сигнала ЭЭГ.

На рис. 3 показан результат моделирования алгоритма сегментации и классификации спектрального анализа на скользящем окне. Решение о принадлежности интервала к определенному ритму визуализируется выделением цвета. На рис. 4 показан алгоритм динамических эталонов на скользящем окне.

На рис. 3 и 4 верхний график иллюстрирует истинные значения ритмов, а на нижнем графике окончательное принятое решение по выбранному пользователем алгоритму обработки.

В результате проведенной серии статистических испытаний алгоритма сегментации выяснилось, что вероятность правильной сегментации составляет 93–97%.

Данное исследование посвящено разработке и исследованию алгоритмов обработке электроэнцефалограмм. В исследовании был реализован и исследован метод сегментации и классификации на основе метода скользящего спектрального анализа и метод динамических эталонов тоже на скользящем окне.

Библиографический список

1. <http://ru.wikipedia.org>
2. Сахаров В. Л., Андреев А. С. Методы математической обработки электроэнцефалограмм: учебн/ пособие. Таганрог: Антон, 2000.
3. Жаринов И. О. Методы и устройства быстрой сегментации ЭЭГ-сигнала: автореферат. СПб., 2004.

УДК 621

П. С. Иванов – студент кафедры вычислительных систем и сетей

П. Л. Волков – научный руководитель

МОДЕРНИЗАЦИЯ ДРАЙВЕРОВ МОСТОВ SPACEWIRE

Стандарт SpaceWire ECSS-E-50-12A разработан международной группой специалистов под эгидой Европейского центра космических технологий (ESTEC) Европейского космического агентства (ESA) и принят в 2003 г. Он разработан на основе стандартов IEEE1355-1995 и TIA/EIA-644 с адаптацией и развитием в соответствии с требованиями аэрокосмических применений:

1. Устойчивость к отказам и сбоям.
2. Низкое энергопотребление.
3. Требования ЭМС.
4. Компактная реализация в СБИС.
5. Поддержка систем реального времени, системных функций бортовых комплексов.

Данный стандарт поддерживается, развивается и внедряется в проектах космической техники национальными космическими агентствами такими, как ESA (European Space Agency), NASA (USA), JAXA (Japan), Роскосмос (РФ).

Стандарт SpaceWire описывает физические разъемы и кабели, электрические сигналы, служебные символы и символы данных, решает задачи управления потоком, поддержания соединения, синхронизация сети, передачи пакетов по сети любой топологии.

Стек протоколов SpaceWire соответствует трём нижним уровням модели OSI. Используются дуплексные каналы со скоростью передачи данных 2-400 Мбит/с в каждом направлении. Длина кабеля до 10м (и выше при снижении максимальной скорости).

На данный момент много стран заинтересовано в продвижении технологии SpaceWire. Так, например, в России компанией ГУП НПЦ «ЭЛВИС» (при участии ЗАО НПЦ «Мит» и дизайн-центров ЗАО ЦП «Ангстрем-СБИС» и ЗАО «Ангстрем-М») разработана серия многоядерных (MultiCore) микросхем «Мультиборт», содержащих интерфейсы SpaceWire.

В состав этой серии входят следующие микросхемы:

1. MC-24R – Радиационно-стойкий сигнальный микропроцессор (2 встроенных канала, скорость прием/передача до 400 Мбит/сек).
2. МСТ-01 – Периферийный контроллер (2 встроенных канала, скорость прием/передача до 400 Мбит/сек).
3. МСК-01 Коммутатор пакетной передачи данных (16 каналов, скорость прием/передача до 400 Мбит/сек).
4. МСВ-01.2.ASIC – Контроллер пакетной передачи данных (4 канала, скорость прием/передача до 600 Мбит/с).

На базе этих микросхем можно организовывать целые бортовые подсистемы. При этом в качестве вычислительных модулей могут использоваться MC-24R и МСТ-01, а в качестве оборудования для создания сети МСК-01 и МСВ-01.

Перед автором стояла задача: увеличение функциональности и модернизация драйвера для серии адаптеров (контроллеров) пакетной передачи данных для мостов МСВ-01, а именно:

- добавить обработчик события «Connection Established»;
- улучшить алгоритм приема Interrupt/Acknowledge кодов (кодов распределенных прерываний).

Драйвер для мостов (МСВ-01.2.ASIC, МСВ-01.2.FPGA, МСВ-01.2m.FPGA, MC-24R, SW.2.ASIC) разработан таким образом, что один проект является драйвером для всех устройств. Для сборки драйвера для определенного устройства перед компиляцией проекта нужно оставить ту директиву условной компиляции, которая является директивой компиляции для нужного устройства.

Для драйвера разработано десять функций (табл.1).

Таблица 1

Функции драйвера

Функции драйвера	
Название функции	Описание функции
InterruptHandler	Обработчик прерываний от моста или DMA (ошибка канала, пакет принят, принят управляющий код из сети)
SetChannelMode	Настроить канал: скорость передачи, режим работы (рабочий или отладочный), и т.п.
GetChannelState	Опросить канал: состояние соединения, скорость приема, действительная скорость передачи, информация об ошибках и т. п.
Reset Channel	Функция сброса канала
ResetHandler	Функция сброса устройства
runSendPacketFromDevMem	Функция отправки пакета из памяти устройства
runSendPacket	Функция отправки пакета из памяти приложения
SendControlCode	Функция отправки управляющего кода
AddEventHandler	Добавить функцию в очередь событий
DeleteEventHandler	Удалить функцию из очереди событий

До добавления события «Соединение установлено», в драйвере было четыре события:

1. Пакет принят.
2. Пакет передан.
3. Принят управляющий код.
4. Ошибка соединения.

Пользовательскому приложению нужно знать состояние соединения в любой момент времени

Для драйвера важно знать правильное состояние соединения, чтобы не пытаться отсылать данные в тот момент, когда это физически невозможно. До изменения можно было узнать, что соединение разорвано (событие «Ошибка соединения»), в этом случае очередь останавливалась, `runSendPacketFromDevMem` или `runSendPacket` не вызывался. Для драйвера важно знать правильное состояние соединения, чтобы не пытаться отсылать данные в тот момент, когда это физически невозможно, с этой целью было разработано событие «Соединение установлено».

Протокол SpaceWire включает в себя механизм распределенных прерываний. На символьном уровне стека проколов SpaceWire под распределенные прерывания и подтверждения выделяются специальные символы, которые называем Interrupt-кодами и `Interrupt_Acknowledge`-кодами соответственно. Эти символы имеют более высокий приоритет, чем у символов данных, что позволяет быстро передавать распределенные прерывания в условиях сильной загрузки сети передачей данных, обеспечивать прохождение этих кодов без задержки через загруженные или заблокированные данными пути. Механизм был разработан для надежной и быстрой передачи системных сигналов (прерываний) с подтверждением по сетям на коммутаторах. Под код распределенного прерывания отводится 6 бит, что позволяет закодировать 64 типа распределенных прерываний. В общем случае, источниками и обработчиками распределенных прерываний являются терминальные узлы (узлы-источники и узлы-обработчики).

Например, распределенным прерыванием может быть какое-то событие (в узле), которое должен обработать один или несколько узлов (обработчиков) в сети, причем, узлу-источнику не требуется информации о том, какой узел является обработчиком этого прерывания. Узел-обработчик, после получения распределенного прерывания, должен отослать подтверждение узлу-источнику. Поэтому, Interrupt-коды и `Interrupt_Acknowledge` рассылаются широкоэвещательно всем узлам в сети.

В качестве защиты от повторной передачи в сетях с циклами каждый узел и каждый коммутатор имеет специальный 64-х разрядный регистр ISR, i-й бит которого соответствует типу прерывания с номером i. При возникновении в узле определенного события, формируется Interrupt-код с шести битным кодом, соответствующим этому событию. Проверяется регистр ISR. Если соответствующий бит ISR равен нулю, то он устанавливается в «1» и Interrupt-код отсылается в сеть. Если бит ISR уже установлен в «1», то это значит, что прерывание с таким кодом было отправлено в сеть, и подтверждение ещё не было еще получено, поэтому повторно прерывание не отсылается. При получении подтверждения (`Interrupt_Acknowledge`-кода) соответствующий его типу бит ISR обнуляется.

Коммутатор, при получении Interrupt-кода, проверяет соответствующий бит ISR. Если бит равен нулю, то Interrupt-код передается на все выходные порты кроме того, с которого пришел. Если бит ISR равен единице, то данный Interrupt-код игнорируется и далее не передается, что необходимо для предотвращения повторной передачи в циклических структурах. На рис. 1 в общем виде приведен пример распространения распределенного прерывания от узла-источника до узла-обработчика. Цифра внутри устройства обозначает бит в регистре ISR, соответствующий данному типу прерывания.

Драйвер предыдущей версии не обрабатывал Interrupt/Acknowledge коды, приходящие с высокой интенсивностью. Под высокой интенсивностью в нашем случае нужно понимать ситуацию, когда за одно прерывание устройства, приходило более одного

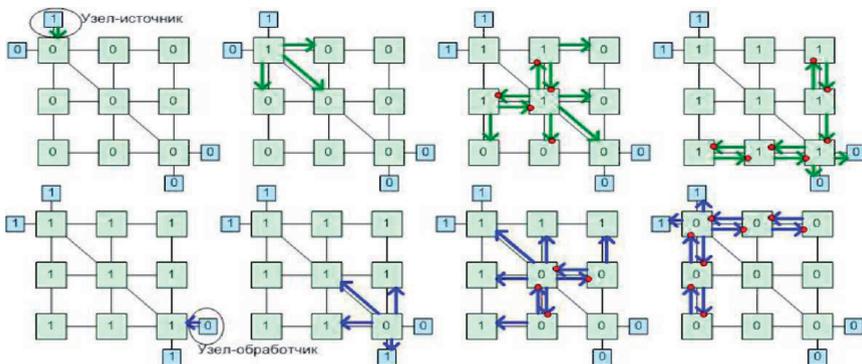


Рис. 1. Пример распространения распределенного прерывания и подтверждения

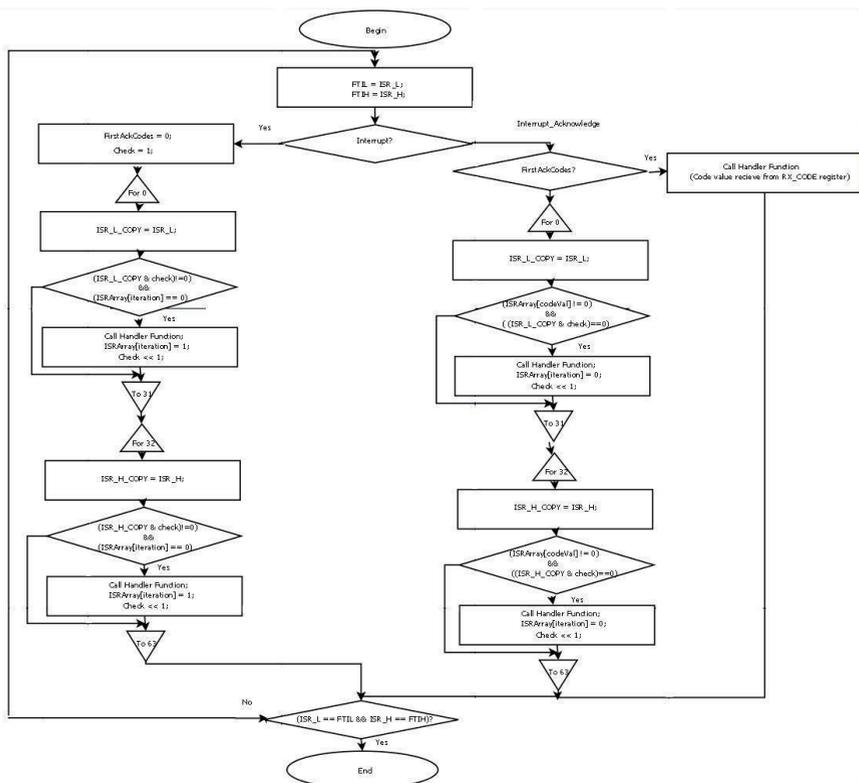


Рис. 2. Алгоритм обработки приема кодов распределенного прерывания

кода. Целью было разработать и реализовать алгоритм приема Interrupt/Acknowledge кодов с любой интенсивностью (рис. 2).

Описание алгоритма

При возникновении события, обработчик прерывания распознает тип прерывания, в данном случае будем считать, что принят код распределенного прерывания из сети. Для выполнения алгоритма существуют вспомогательные переменные: ISR_L_COPY (Переменная в которую копируется значение регистра ISR_L), ISR_H_COPY (переменная в которую копируется значение регистра ISR_H), FirstAckCodes (Флаг приема первых Interrupt Acknowledge кодов из сети), check (переменная для нахождения «взведенной 1»), ISRArray[64] (массив, в котором будем «отмечать» принятые коды, а так же эта переменная требуется для предотвращения дублирования обработки кодов). 64-х разрядный регистр ISR аппаратно реализован как два 32-х разрядных регистра ISR_L и ISR_H. Начинаем проверку младшей части регистра ISR. Копируем значение регистра ISR, если условие ((ISR_L_COPY & check!=0) и (ISRArray[iteration]==0)) выполняется, то мы обрабатываем принятый код, поняв о номере распределенного прерывания по номеру итерации и «выставляем» во вспомогательном массиве ISRArray признак, что код с этим номером был принят и обработан. Со старшей частью регистра ISR алгоритм обработки аналогичен. Алгоритмически обработка Interrupt и Interrupt_Acknowledge кодов различаются циклическими условиями и признаком в массиве ISRArray.

Результаты модернизации:

1. Событие «Соединение установлено» разработано, используется в пользовательских приложениях.
2. Драйвер обрабатывает любое количество [1..64] Interrupt кодов, принятых за одно прерывание.
3. Устройство обрабатывает любое количество принятых Interrupt-Acknowledge кодов за одно прерывание, при условии, что в этом же прерывании не будет принято Interrupt кодов.

Библиографический список

1. ECSS-E50-12A. SpaceWire – Links, nodes, routers and networks. – European Cooperation for Space Standardization (ECSS), 2003.
2. Петрикович Я. Я., Солохина Т. В., Шейнин Ю. Е. Технология SpaceWire для параллельных систем и бортовых распределенных комплексов// ЭЛЕКТРОНИКА: Наука, Технология, Бизнес. 2006. № 5. С. 64–75.
3. Liudmila Koblyakova, Yuriy Sheynin, Dmitry Raszhivin. Real-time signalling in networked embedded systems. International SpaceWire Conference, St.Petersburg 2010. Conference Proceedings. P. 385–388.

УДК 004.942

Е. В. Иванова – магистрант кафедры моделирования вычислительных и электронных систем

Е. А. Бакин – научный руководитель

МОДЕЛЬ ПЛАНИРОВЩИКА ДЛЯ СЕТИ WiMAX ПРИ НАЛИЧИИ АГРЕССИВНЫХ АБОНЕНТОВ С ОЦЕНКОЙ ИНТЕНСИВНОСТИ ОБСЛУЖИВАНИЯ

Проблемы беспроводной сети все более актуальны. Все больше абонентов нагружают сеть большим объемом неприоритетных данных (агрессивные абоненты), например передачей медиа-контента. Вследствие большой нагрузки на сети 4G выявляются такие проблемы как снижение скорости обслуживания абонентов с небольшим трафиком (неагрессивных абонентов). Рассмотрим принципы работы сети WiMax.

Практически все сети работают согласно стандарту IEEE 802.16е. Сети этого стандарта используют топологию типа «звезда», то есть каждая абонентская станция (АС) связана с базовой станцией (БС) напрямую (рис. 1).

Обмен информацией между БС и АС осуществляется с помощью последовательности кадров (рис. 2). Кадр можно условно разделить на 3 части: обмен служебными сообщениями и 2 субкадра: восходящий, в котором осуществляется передача данных от АС к БС, и нисходящий, в котором осуществляется передача данных и расписания от БС к АС.

Все передачи в сети строго синхронизированы и назначаются базовой станцией. Для этого АС доставляет на БС запросы на выделение ресурса канала.

Распределением ресурсов между абонентами занимается планировщик. Он определяет в соответствии с вычисленной метрикой абонента, какое количество слотов выделить ему в каждом кадре. Метрика планирования – характеристика, от которой зависит приоритет обслуживания абонента планировщиком БС. Метрика абонента вычисляется в соответствии с выбранной дисциплиной планирования. Компенсировать отрицательный эффект задержки передачи данных для неагрессивных абонентов может правильный выбор дисциплины планирования. Рассмотрим основные виды планировщиков.

Планировщик с круговым опросом (ПКО) используется при моделировании сети IEEE 802.16. На БС для каждого абонента заводится буфер запросов (БЗА). После успешной доставки запроса о выделении РК на БС, информация о запросе поступает в БЗА. В начале каждого кадра планировщик последовательно опрашивает буферы абонентов. Если буфер не пустой, то планировщик обслуживает один запрос, выделив абоненту запрашиваемое число слотов в субкадрах. Буферы опрашиваются циклически до тех пор,



Рис. 1. Топология сети стандарта IEEE 802.16

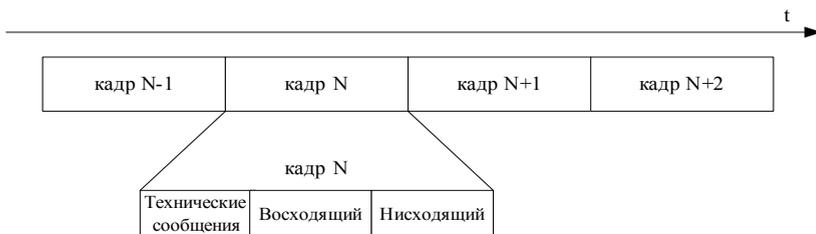


Рис. 2. Обобщенная структура кадра

пока ресурс канала не будет исчерпан. Если ресурс кадра закончился, а запросы еще остались, то со следующего кадра планировщик начнет опрашивать БЗА, следующий за тем, на котором он остановился. Если ресурсы в кадре еще есть, а запросы в буферах закончились, то остальные ячейки в кадре заполняются нулями. [1]

Рассмотрим работу данного планировщика в условиях присутствия агрессивных абонентов. В буфере запросов агрессивного абонента всегда имеются заявки, а это значит, что при каждом циклическом опросе буферов, планировщик всегда вынужден выделить им минимум M единиц ресурса в кадре. При ресурсе канала X и наличии M агрессивных абонентов для остальных абонентов N ресурс канала будет равен: $X \cdot \frac{N - M}{N}$. Планировщик Round Robin не фиксирует историю ЗВПК абонента, и при определенной его активности, агрессивный абонент может занять большинство очередей планировщика, тем самым существенно ухудшив пропускную способность остальных абонентов.

Пропорционально справедливый планировщик (ПСП) Рекомендован для моделирования ведущими разработчиками беспроводных решений [2]. Дисциплина планирования данного ПСП основана на вычислении метрик абонентов. Вычисление метрик абонентов более подробно было рассмотрено в статье [3].

Метрики обновляются после планирования каждого кадра. Стоит обратить внимание на то, что количество ресурсов выделяется пользователю в зависимости от его метрики, поэтому один пользователь может получить большинство ресурсов в кадре.

В простейшей форме ПСП вычисляет метрики для всех активных пользователей на заданном интервале планирования. Пользователю с наивысшей метрикой выделяются все необходимые ресурсы, доступные в данном интервале, метрики для всех пользователей обновляются перед следующим интервалом планирования, и процесс повторяется. Для систем OFDMA, интервалом планирования обычно является кадр, который делится между пользователями в соответствии с их метриками (но не пропорционально). Таким образом, в простейшем расширении систем OFDMA, для ПСП присутствуют следующие особенности: фиксируются ресурсы планирования, которые планируются последовательно до тех пор, пока все доступные ресурсы не будут зарезервированы.

Изменение метрики абонента зависит от изменения качества связи: чем выше качество связи, тем выше метрика абонента. Следовательно, в случае, когда качество связи абонента существенно не меняется, данный планировщик работает как рассмотренный ранее ПКО.

Планировщик кругового опроса обслуживает всех абонентов поочередно, не анализируя очередь каждого абонента, и если неагрессивный абонент с малой интенсивностью пошлет запрос, то его обслужат только в порядке очереди, несмотря на высокую интенсивность других абонентов. Для улучшения качества обслуживания неагрессивных абонентов предложена следующая дисциплина опроса буферов. Вводятся метрики, рассчитываемые как оценка интенсивности обслуживания абонентов за последние несколько кадров. Таким образом, в текущем кадре наибольший приоритет имеет тот абонент, у которого за последние N кадров (N – параметр алгоритма опроса) было обслужено наименьшее число запросов. Если таких абонентов несколько, то для их обслуживания задается случайный порядок. Такой подход позволяет обеспечить неагрессивным абонентам наименьшую задержку в обслуживании.

Для сравнения средней задержки обслуживания абонентов при работе сети с использованием планировщиков кругового опроса и с оценкой интенсивности обслуживания, было проведено моделирование, соответствующее стандартам моделирования работы сети WiMax. Целью моделирования являлось сравнение ПКО и ПОИО, так как ПСП при рассматриваемых условиях (постоянство качества передачи) сводится к ПКО.

Моделирование проводилось для следующих условий (см. таблицу 1). В данной модели максимальный суммарный поток неагрессивных абонентов ($V_{\text{негр}}$) принят равным $V_{\text{max}} = 3,5$ Мб/с, и меняется соответственно доли неагрессивных абонентов в потоке ($P_{\text{негр}}$) $V_{\text{негр}} = P_{\text{негр}} \cdot V_{\text{max}}$. Суммарный поток агрессивных абонентов ($V_{\text{агр}}$) изменя-

ется в процессе моделирования, в соответствии с долей агрессивных абонентов в потоке ($P_{\text{агр}}$) $V_{\text{агр}} = K \cdot P_{\text{агр}} \cdot V_{\text{max}}$, где K – коэффициент агрессивности. Доля агрессивных абонентов в потоке изменяется от $N_{\text{абон}} \cdot 10\%$ до $N_{\text{абон}} \cdot 90\%$.

По итогам моделирования были получены графики, показывающие среднюю задержку запросов как неагрессивных абонентов (измеряемую в кадрах) при ПКО (рис. 3) и ПОИО (рис. 4), так и агрессивных абонентов (рис. 5), в условиях увеличения доли агрессивных абонентов (от 10 до 90%) в канале связи.

Таблица 1

Параметры моделируемой системы

Время моделирования	4 минуты реального времени работы системы
Суммарный трафик от всех абонентов	От 4,9 до 16,1 Мб/с
Общее число обслуживаемых абонентов	30
Максимальный трафик неагрессивных абонентов	3 Мб/с
Количество агрессивных абонентов	от 10% до 90% от общего числа абонентов
Коэффициент агрессивности	5
Размер одного запроса абонента	1536 бит
Длительность кадра (сек)	0,005

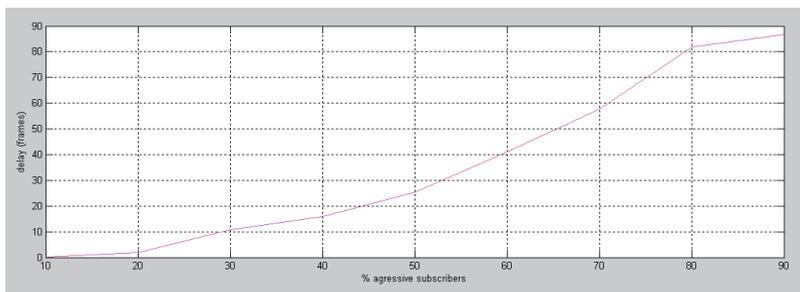


Рис. 3. Средняя задержка для неагрессивных абонентов при ПКО

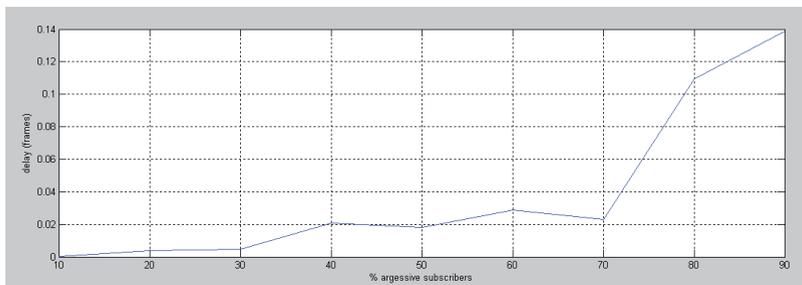


Рис. 4. Средняя задержка для неагрессивных абонентов при ПОИО

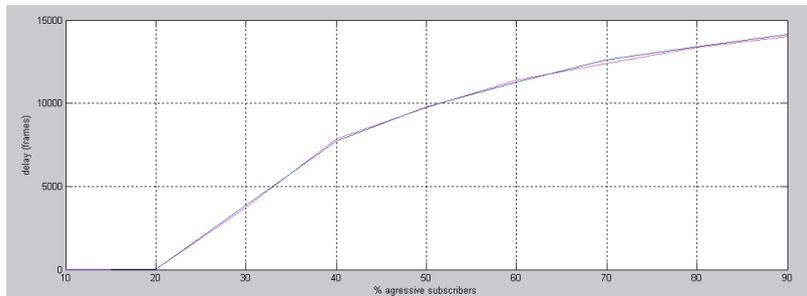


Рис. 5. Средняя задержка для агрессивных абонентов при ПОИО и ПКО

Из рассмотрения графиков были сделаны следующие выводы:

1. Для агрессивных абонентов планировщики ведут себя практически одинаково.
2. Для неагрессивных абонентов ПОИО обеспечивает существенно меньшую задержку обслуживания чем ПКО (с увеличением доли агрессивных абонентов выигрыш по задержке возрастает до нескольких порядков)

Предложенный планировщик обслуживания запросов позволяет в сравнении с ПКО существенно уменьшить задержку обслуживания запросов от неагрессивных абонентов при сохранении уровня задержки обслуживания запросов от агрессивных абонентов.

Библиографический список

1. Вишнеvский В. М., Портной С. Л., Шахнович И. В. Энциклопедия WiMAX. Путь к 4G. М.: Техносфера, 2009. 472 с.
2. IEEE 802.16m Evaluation Methodology Document (EMD). Evaluation Methodology for P802.16m-Advanced Air Interface, 2009.
3. Resource distribution disciplines for wireless regional networks. Материалы междурядного форума «Формирование современного информационного общества-проблемы, перспективы, инновационные подходы», Ivanova E., St-Petersburg, 2010.

УДК 76.01.05

Е. В. Казённая – студентка кафедры моделирования вычислительных и электронных систем

А. В. Яковлев (канд. техн. наук, доц.) – научный руководитель

ПРЕДСТАВЛЕНИЕ МЕДИЦИНСКОГО СООБЩЕСТВА КАК СТРУКТУРИРОВАННОЙ СИСТЕМЫ С ГРУППАМИ ВЗАИМОДЕЙСТВУЮЩИХ ЭЛЕМЕНТОВ

Представление бизнес-сообщества в виде таксономии является сложной задачей с практической точки зрения и относится к проблематике инженерии знаний¹. Для наших целей определим бизнес-сообщество, как множество юридических и физических лиц,

¹ Червинская К. Р., Щелкова О. Ю. Медицинская психодиагностика и инженерия знаний / Под ред. Л. И. Вассермана. СПб.: Ювента; М.: Издательский центр «Академия», 2002. 624 с.

которые производят, продают, обеспечивают продажу таких товаров, услуг, проектов, которые удовлетворяют (или частично удовлетворяют) какие-либо определенные потребности покупателей. Под таксономией бизнес-сообщества понимается классификационная схема, которая структурирует знания о предметной области определенного бизнес-сообщества и определяет отношения между элементами знаний. Эта классификационная схема в дальнейшем используется в маркетинговых информационных системах, осуществляющих поиск и изучение информации о состоянии определенного рынка, о конкурентах, о потенциальных потребителях услуг определенного бизнес-сообщества.

Стандартная структура таксономии бизнес-сообщества приведена на рис. 1.



Рис. 1. Формальное описание бизнес-сообщества в виде таксономии



Рис. 2. Схема таксономии бизнес-сообщества «Эстетическая медицина Санкт-Петербурга»

В качестве иллюстрации, рассмотрим построение таксономии бизнес-сообщества «эстетическая медицина Санкт-Петербурга». Это бизнес-сообщество является одним из самых интересных с точки зрения его насыщенности различными организациями и сложности взаимосвязей между ними. Это бизнес-сообщество удовлетворяет потребность жителей города в красоте, т.е. потребность хорошо выглядеть и чувствовать себя здоровыми. Удовлетворением этой потребности занимается большая группа компаний эстетической медицины. В свою очередь эти компании имеют тесные связи со страховыми компаниями (по полисам ДМС), с фондом обязательного медицинского страхования (ФОМС), с кредитными организациями, с отечественными и зарубежными поставщиками оборудования, расходных материалов, косметических и лечебных средств, с образовательными учреждениями, центрами профессиональной подготовки и т. д.

На рис. 2 приводится начальный этап описания бизнес-сообщества «Эстетическая медицина СПб». Указанная схема сделана в специализированном редакторе MindManager. Таксономия содержит следующие крупные группы элементов:

1. Дистрибьюторы – компании, осуществляющие оптовую закупку и сбыт товаров на региональных рынках. Фирма может иметь собственного дистрибьютора по продаже своих товаров за рубежом, где он является на основе заключенного договора ее единственным представителем (генеральный дистрибьютор). Дистрибьютор оказывает посреднические маркетинговые услуги продавцам и покупателям, а также услуги по монтажу и наладке оборудования, обучению пользования им.

2. Производители – это предприятия по производству лекарственных средств и медицинского оборудования, сотрудничающие напрямую с медицинскими центрами, или реализовывающую продукцию через дистрибьюторов.

3. Обучающие организации – государственные и частные учреждения, осуществляющие подготовку врачей и среднего медицинского персонала, административного и инженерно-технического состава.

4. Медицинские центры – организации, непосредственно оказывающие услуги.

Такие схемы таксономий, создаваемые в MindManager и других аналогичных построителях таксономий, очень удобны и понятны с точки зрения визуализации. Однако для дальнейшей обработки и анализа, данных они не подходят. В связи с этим возникла необходимость разработки удобной базы данных для хранения информации о бизнес-сообществах.

В рамках магистерской диссертации предложена система для хранения этих сведений в виде базы данных, а также программа для оценки параметров, характеризующих экономическое положение компании, работающей в определенном бизнес-сообществе, на основании анализа тесноты ее связей внутри бизнес-сообщества.

УДК 004.514

Я. В. Клементьев – магистрант кафедры моделирования вычислительных и электронных систем

В. С. Павлов (канд. техн. наук, доц.) – научный руководитель

СОВРЕМЕННЫЕ ПРОГРАММНЫЕ СРЕДСТВА, ПОВЫШАЮЩИЕ ЭФФЕКТИВНОСТЬ ИНТЕРФЕЙСОВ ВЕБ-РЕСУРСОВ

В наши дни Интернет стал неотъемлемой частью жизни людей. Для удовлетворения своих потребностей современному человеку приходится проводить время в сети, а время – это то, что нужно использовать максимально эффективно. Эффективность – это способность приводить к нужным результатам [1]. При работе в Интернете пользователи достигают своих целей, взаимодействуя с интерфейсами необходимых им

сайтов. Следовательно, повышение эффективности интерфейсов становится ключевой проблемой при разработке веб-ресурсов. Эта проблема является частью научного направления HCI (human-computer interaction), исследующего вопросы взаимодействия «человек-компьютер» с целью совершенствования методов разработки, оценки и внедрения интерактивных компьютерных систем, предназначенных для использования человеком [2].

Для того чтобы оценить, насколько эффективны интерфейсы тех или иных веб-ресурсов, необходимо провести их классификацию, так как при решении различных задач, соответственно, понятия эффективности могут различаться в зависимости от того, каким образом и какую информацию они должны донести до посетителей.

Современный Интернет можно условно разделить на Web 1.0 и Web 2.0, подходы к проектированию веб-ресурсов, для которых сильно отличаются друг от друга.

Основным признаком Web 1.0 является классическая модель организации отношений клиент-сервер – односторонний поток информации через веб-сайты, содержащие материал «только для чтения». Примерами таких сайтов служат интернет-магазины, информационные сайты и т.д.

Особенностями Web 2.0. являются принцип привлечения пользователей к наполнению и многократной выверке контента (т.е. информация на сайте формируется самими пользователями), веб-синдикация (одновременное распространение медиа-информации на разных веб-сайтах путём передачи заголовков и ссылок), mash-up (заимствование информации путем интегрирования веб-сервисов друг с другом). Происходит социализация интернет-сообщества. Примерами таких сайтов служат социальные сети, блоги, торрент-трекеры и т. д. [3]

Анализируя принципиальные различия Web 1.0 и Web 2.0, можно прийти к выводу, что эффективность интерфейсов в обоих случаях сводится к скорости нахождения нужной информации, удобству представления контента и обеспечению интерактивности.

Эффективность интерфейсов веб-ресурсов можно представить как совокупность нескольких составляющих. К ним относятся:

Дизайн. Он должен быть простым, понятным и создавать комфортную психологическую обстановку.

Интерактивные возможности интерфейса. Упрощение взаимодействия с пользователем путем быстрых динамических ответов.

Персонализация. В понятие персонализации сайта можно включить возможность индивидуальных настроек сайта и создание личной зоны (личные файлы, изображения, видео, блоги) для пользователя, чтобы он чувствовал свою уникальность.

Все эти составляющие можно обобщить как понятие «юзабилити» интерфейса веб-сайта [4].

Ключевыми современными программными средствами, позволяющими упростить диалог «человек-компьютер» и увеличить его эффективность при работе в Интернете, являются HTML 5 и CSS 3. Проведем анализ современных программных средств с позиции их влияния на перечисленные составляющие эффективности (табл. 1). Рассмотрим основные нововведения данных технологий.

1. Новые теги блоков типовой разметки header, nav, section, article, aside, footer выведены на уровень базового синтаксиса. Это упрощает процесс программирования за счет исключения ряда рутинных операций и облегчает работу поисковым системам.

2. Добавлены новые атрибуты для элементов. Атрибуты class, id, style, title и др., существовавшие в HTML4, стали применимы для всех элементов, т.е. стали глобальными атрибутами. Добавлены новые глобальные атрибуты: contenteditable, contextmenu, data-, dropzone, hidden, spellcheck и др. Это расширяет возможности дизайна и упрощает его разработку.

3. Объект векторной графики canvas — элемент HTML 5, предназначенный для создания растрового изображения с помощью JavaScript. На сегодняшний день canvas чаще

используется для построения графиков, простой анимации и игр в веб-браузерах. Это улучшает интерфейс веб-ресурса на уровне восприятия форм, что обосновано в HCI с позиций когнитивной психологии.

4. Новые элементы video, audio и др. В последние годы аудиозаписи и видеоролики получили широкое распространение в Интернете благодаря многочисленным сервисам, облегчающим их публикацию. Но для обеспечения правильного воспроизведения мультимедиа-контента приходилось использовать дополнительные технологии и, следовательно, ресурсы. HTML 5 позволяет вставлять его в веб-страницу, применяя соответствующий контейнер. Эти нововведения значительно упрощают публикацию контента в Интернете и его заимствование, что влияет на повышение интерактивности веб-ресурса.

5. Новые API-интерфейсы программирования приложений — набор готовых классов, процедур, функций: рисование 2D-картинок в реальном времени, контроль проигрывания медиа-файлов, Drag-and-drop, хранение данных в браузере и др. Эти API позволяют пользователю управлять содержимым страницы и чувствовать свою уникальность.

6. Нововведения CSS 3 (каскадные таблицы стилей) заключаются в том, что помимо новых возможностей, таких как анимация и введение переменных, в этой версии доступны закругленные углы, границы, тени, использование своих шрифтов, растягивание блоков, мультиконочность и др. возможности, выведенные на уровень базового синтаксиса. Это обеспечивает упрощенную графическую основу для введения динамических элементов веб-страниц с целью повышения их интерактивности. Кроме того, данное нововведение чрезвычайно важно для осуществления самонастройки интерфейса под конкретный тип Интернет-устройства пользователя.

Таблица 1

Влияние программных средств на составляющие эффективности веб-ресурсов

Составляющая	HTML 5	CSS 3
Дизайн	1. Новые теги блоков типовой разметки; 2. Новые атрибуты и расширение их применения для всех элементов; 3. Объект векторной графики canvas;	6. Закругленные углы, границы, тени, использование своих шрифтов, ратягивание блоков, анимация, возможность введения переменных, мультиконочность и др. выведены на уровень базового синтаксиса.
Интерактивность	4. Новые элементы video, audio, progress, meter, time и др.	
Персонализация	5. Новые API (рисование 2D-картинок в реальном времени, управление проигрыванием медиафайлов, Drag-and-drop и др.)	

Новые технологии сфокусированы на упрощении процесса программирования и расширении возможностей дизайна. Упрощение процесса программирования позволяет уделить больше внимания разработке дизайна и проблемам юзабилити при тех же затратах на разработку. Расширение возможностей дизайна позволяет создать более комфортный и эффективный интерфейс, максимально повысить его юзабилити, тем самым привлечь новых пользователей, в том числе более молодых.

Из проведенного в работе анализа следует, что новые программные средства разработки веб-систем создают существенный задел для повышения эффективности наиболее чувствительного для пользователя элемента веб-ресурса — его интерфейса. При этом большая часть нововведений непосредственно относится к интерфейсу и значительно упрощает процесс его программирования. Это создает широкие возможности

для творческого подхода в дизайне, разработки новых специфических стратегий взаимодействия с пользователем, что в конечном итоге позволяет максимально повысить эффективность интерфейса веб-ресурса.

Библиографический список

1. <http://ru.wiktionary.org/wiki/эффективность>
2. http://ru.wikipedia.org/wiki/Человеко-компьютерное_взаимодействие
3. Тим О'Рейли. Что такое Веб 2.0 // Компьютерра. 2005. № 37. С. 9-13, № 38. С. 11–14.
4. Якоб Нильсен. Разработка Юзабилити. Morgan Kaufmann, 1993. 362 с.

УДК 629.735.33

А. С. Колесов — магистрант кафедры вычислительных систем и сетей
В. А. Галанина (канд. техн. наук, доц.) — научный руководитель

СИМУЛИРОВАНИЕ ДОРОЖНОГО ДВИЖЕНИЯ

В течение последнего века наблюдается стремительный рост автомобилизации людей в развитых и развивающихся странах, что приводит к увеличению загруженности существующих дорог и строительству новых. В то время как развитые страны уже достигли насыщения рынка, в развивающихся странах продолжается стремительный рост числа автомобилей на душу населения, связанный в том числе и со стремительным ростом размеров агломераций и неадекватности темпов развития общественного транспорта к текущим запросам. Ситуация также осложняется тем, что в уже застроенных районах нет возможности увеличить пропускную способность дорожной сети за счёт увеличения ширины дорожного полотна и строительства развязок без сноса существующих зданий, что может быть в некоторых ситуациях затратным мероприятием. Кроме того, строительство новых дорог не всегда имеет смысл из-за постулата Льюиса-Моргриджа [1] и парадокса Браесса [2], утверждающих, что строительство новых дорог не обязательно приводит к увеличению пропускной способности всей системы.

Как следствие, для решения транспортных проблем часто единственным возможным шагом является создание модели транспортной сети, с помощью которой определяется наиболее приемлемый вариант строительства без экспериментов над реальными дорогами и водителями. Предварительное моделирование также поможет оценить перспективы развития системы и её способность к увеличению пропускной способности при увеличении транспортного потока в будущем.

Целью данной работы является разработка системы для моделирования дорожного движения с использованием распределённых вычислений и поддержкой как моделирования в реальном времени, так и неоперативного (допускающего относительно ощутимую задержку между постановкой задачи в очередь и началом её решения). Для моделирования дорожного движения будет использоваться собственная микросимуляционная модель с возможностью моделирования дорожных сетей и отдельных элементов дорог, например перекрёстков. Разработанная система должна обладать наглядным графическим отображением результатов моделирования.

- В основе проекта системы моделирования лежит несколько принципов:
- данные хранятся в документо-ориентированной базе данных MongoDB;
 - управление системой осуществляется через HTTP API;
 - для пользователей предусмотрен Web интерфейс;

– задания выполняются процессами-исполнителями (workers), которых может быть несколько для одновременного выполнения различных симуляционных задач;

– в системе присутствуют Планировщик, который раздаёт задания исполнителям и Супервизор, задачей которого является следить за тем, чтобы задания не попадали в тупик из-за неожиданно пропавшего исполнителя.

Большая часть общения между компонентами системы происходит с использованием легковесного протокола JSON-RPC через HTTP. Для получения данных о состоянии и результатах симулирования используются HTTP GET запросы. Общение с БД происходит в её родном формате BSON. Исполнители общаются с БД напрямую, а не через HTTP API, чтобы снизить нагрузку на сервер API.

В перспективе стоит задача создания более сложного планировщика, поддерживающего приоритеты и учитывающего статистику работы конкретных исполнителей, благодаря чему можно будет реализовать выполнение симуляционных задач в реальном времени. Кроме того, возможно, имеет смысл создать срединный слой между исполнителями и базой данных для контроля доступа к данным.

На настоящий момент реализована весьма простая симуляционная модель:

- одна дорога любой длины, с любым числом полос, движение осуществляется только в одном направлении, есть возможность добавлять светофоры;
- машины генерируются на входе с заданной частотой;
- машины могут обгонять впереди идущую машину, если та слишком медленная и есть место для обгона;
- машины тормозят и разгоняются мгновенно.

Статистическая обработка результатов моделирования пока ограничивается определением значений таких параметров как среднее время прохождения участка, время простоя на холостом ходу, пропускная способность участков дорог. Через интерфейс пользователя эти данные можно экспортировать в табличном формате для дальнейшего анализа.

В дальнейшем предполагается создание реалистичной модели торможения и разгона, создание систем дорог с перекрёстками, поддержка нескольких типов машин на дороге (грузовики, минивэны, легковушки и т. п.), а также поддержка генерации различных индивидуальных характеристик водителей, для разнообразия их поведения на дороге. А также добавление возможности построения графиков для визуального сравнения моделей и влияния различных параметров на свойства модели, построение карт интенсивности движения на разных участках дорог, а также автоматическое оценивание элементов транспортной системы по шкале уровней сервиса (level of service) [3].

С помощью такой системы можно будет создать программу автоматического перебора различных возможных вариантов транспортной системы для определения наилучшего в рамках выбранного пользователем критерия.

Библиографический список

1. http://ru.wikipedia.org/wiki/Постулат_Льюиса-Моргиджа
2. http://en.wikipedia.org/wiki/Braess%27_paradox
3. http://en.wikipedia.org/wiki/Level_of_service

УДК 004.451.9

П. Я. Кондратюк – магистрант кафедры моделирования вычислительных и электронных систем

В. С. Павлов (канд. техн. наук) — научный руководитель

СРАВНЕНИЕ ОПЕРАЦИОННЫХ СИСТЕМ LINUX И WINDOWS ПО КРИТЕРИЯМ ОЦЕНКИ КАЧЕСТВА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ (ISO 9126)

Задача анализа возможностей использования свободного программного обеспечения (СПО) является весьма актуальной в связи с политикой правительства РФ в сфере информатизации, проводимой в последние годы. В 2007–2011 гг. был опубликован ряд распоряжений, определяющих задачу перехода государственных и бюджетных учреждений к использованию СПО [1]. Возникает потребность в объективном сравнении проприетарного и свободного ПО на основе общепринятых стандартов качества, в частности операционных систем (ОС) Linux и Windows. В качестве основы для сравнения был выбран международный стандарт ISO 9126 «Характеристики и метрики оценки качества программного обеспечения» как наиболее полно отражающий критерии оценки качества ОС.

Стандарт ISO 9126 определяет шесть основных характеристик оценки качества программного обеспечения (рис. 1) и конкретизирующие их подхарактеристики [2].

Проведён анализ соответствия ОС Linux и Windows характеристикам качества программного обеспечения с точки зрения стандарта ISO 9126.



Рис. 1. Модель качества программного обеспечения согласно ISO 9126

1. Функциональность

Linux и Windows могут использоваться как в качестве персональных систем, так и в качестве серверных ОС. С этой точки зрения та или иная конфигурация операционной системы соответствует заложенной в неё функциональности.

Если включать в функциональность операционной системы функции программ и утилит, которые входят в её состав по умолчанию, Linux имеет расширенную по сравнению с Windows функциональность.

Linux по умолчанию обладает расширенными способностями к сетевому взаимодействию. Имеются возможности мультитерминальной работы и удалённого доступа, которые развивались с момента создания данной ОС [3]. В Windows эти возможности доступны в последних серверных версиях, а также с помощью дополнительных средств, например, Citrix.

Согласованность как соответствие стандартам также является немаловажным для операционных систем. Для UNIX-подобных систем определён ряд стандартов POSIX. POSIX (Portable Operating System Interface for Computer Environments – независимый от платформы системный интерфейс для компьютерного окружения) — это стандарт IEEE (Institute of Electrical and Electronics Engineers – институт инженеров по электротехнике и радиоэлектронике), описывающий системные интерфейсы для открытых операционных систем. Часть дистрибутивов Linux поддерживают стандарт POSIX. Системный интерфейс Windows не стандартизирован, системные вызовы различаются от версии к версии [4].

Защищённость обеих операционных систем обеспечивается с помощью современных средств авторизации и аутентификации. Открытость кода Linux делает модель безопасности ОС более прозрачной. Это один из факторов, определивших выбор Linux в качестве национальной программной платформы (НПП).

2. Надежность

Большой объём кода ядра как в Windows, так и в Linux определяет наличие уязвимостей, которые не могут быть полностью устранены на практике. Неблагоприятные последствия проявления таких ошибок могут быть снижены за счёт изоляции подсистем и предоставления пользовательским процессам более низкого приоритета, чем процессам ядра [5].

Существует ряд факторов, которые могут снижать надёжность Windows. Это наличие компонентов, критичных для работы системы, таких как графическая оболочка, ошибки в работе которых могут привести к сбою в системе; риск переполнения буфера из-за недостаточно эффективного механизма замещения страниц виртуальной памяти. Также к числу факторов, снижающих надёжность Windows относительно Linux, можно отнести наличие большого количества вирусов и вредоносных программ, однако оно связано, прежде всего, с популярностью Windows.

3. Практичность

Обеспечение удобства пользовательского интерфейса изначально являлось приоритетной задачей при разработке Windows [4]. Качество интерфейса Linux зависит от дистрибутива. Интерфейсы таких популярных дистрибутивов, как Ubuntu, Mandriva, ALT Linux достаточно просты в освоении и привлекательны для пользователей. Значительная часть настроек автоматизирована, имеются расширенные по сравнению с Windows возможности настройки через командную строку. Однако такое администрирование требует базовых технических знаний и навыков.

Таким образом, Windows по умолчанию обладает большей практичностью при решении стандартных задач.

4. Эффективность

Данная характеристика включает себя скорость отклика операционной системы при выполнении определённых задач, а также эффективность использования ресурсов.

В рамках независимого тестирования производительности Windows 7 и Ubuntu 10.04 было выявлено, что при решении различных задач рассматриваемые ОС могут значительно превосходить друг друга по скорости работы [5].

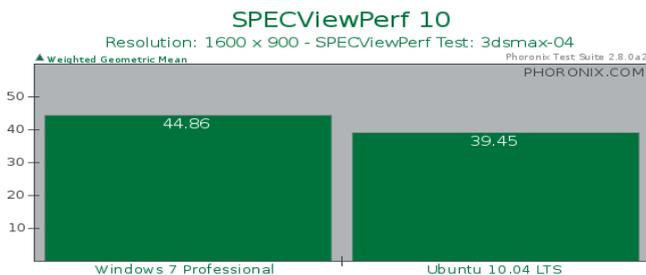


Рис. 2. Производительность ОС при выполнении графической программы 3D Studio MAX

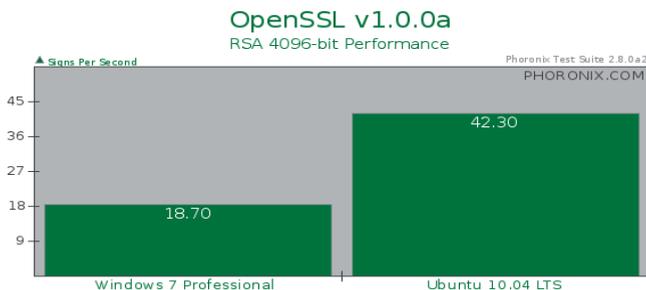


Рис. 3. Производительность ОС при использовании алгоритмов шифрования

Также тестирование показало, что Ubuntu потребляет больше мощности, чем Windows 7 при работе на нетбуке и ноутбуке [6]. В то же время возможность отключения графического режима в Linux обеспечивает более эффективное использование ресурсов при работе на сервере [2].

5. Сопровождаемость

Открытость исходных кодов Linux, а также принцип модульности, лежащий в основе операционной системы, обеспечивают для неё принципиально другой уровень сопровождаемости по сравнению с Windows. Существуют широкие возможности для модификации ОС в соответствии с решаемыми задачами и имеющимися ресурсами. Модульность Linux позволяет сгенерировать (скомпилировать) такое ядро операционной системы, которое будет оптимальным для данного конкретного персонального компьютера и решаемых на нём задач [2].

6. Мобильность

Возможность переноса операционной системы между различными аппаратными архитектурами изначально заложена в Unix-подобных ОС на уровне исходных кодов [2]. Linux совместим с различными архитектурами процессоров: Intel x86, x86-64, PowerPC, Alpha AXP, MIPS и многими другими, что обеспечивает большую мобильность по сравнению с Windows.

Что касается удобства и простоты установки, Windows вполне удовлетворяет этим требованиям. Для Linux простота установки варьируется в зависимости от дистрибутива.

Выводы. ОС Windows и Linux имеют ряд исторически сложившихся принципиальных различий. Windows изначально ориентирована на создание удобного универсального

интерфейса пользователя; Linux — на надёжности и безопасности системы, на возможностях сетевой работы. Как показало сравнение Linux и Windows согласно модели качества ISO 9126, на данный момент эти различия существенно сгладились за счёт развития обеих ОС. Предпочтительность той или иной ОС зависит от конкретной решаемой задачи. Так Windows как правило больше подходит для решения стандартных задач пользователей. Linux может использоваться для решения задач, требующих доработки и гибкой настройки ОС.

Библиографический список

1. Распоряжение от 17 декабря 2010 г. №2299-р «О плане перехода федеральных органов исполнительной власти и федеральных бюджетных учреждений на использование свободного программного обеспечения на 2011–2015 годы».
2. ISO/IEC 9126-1:2001 Software engineering — Product quality.
3. Гордеев А. В. Операционные системы. СПб.: Питер, 2007.
4. Таненбаум Э. Современные операционные системы. СПб.: Питер, 2002.
5. Таненбаум Э., Хердер Дж., Бос Х. Можем ли мы сделать операционные системы надежными и безопасными // IEEE Computer Society. – 2006. – № 5.
6. Данные тестирования быстродействия Windows 7 и Ubuntu 10.04 http://www.phoronix.com/scan.php?page=article&item=ubuntu_win7_ws&num=1
7. Данные тестирования ресурсоемкости Windows 7 и Ubuntu 10.04 http://www.phoronix.com/scan.php?page=article&item=linux_windows_part2&num=1

УДК 004.725.5

С. Ю. Крысин – магистрант кафедры вычислительных систем и сетей
В. А. Галанина (канд. техн. наук) – научный руководитель

**OPENCV – БИБЛИОТЕКА КОМПЬЮТЕРНОГО ЗРЕНИЯ
 С ОТКРЫТЫМ ИСХОДНЫМ КОДОМ**

В настоящее время, при создании систем дистанционного мониторинга, довольно актуальной является проблема выбора технологии для создания модуля видеоаналитики. Видеоаналитика – система программной логики, основанной, прежде всего, на видео анализе, позволяющая выделить наиболее важные моменты в состоянии наблюдаемых объектов (рис. 1).

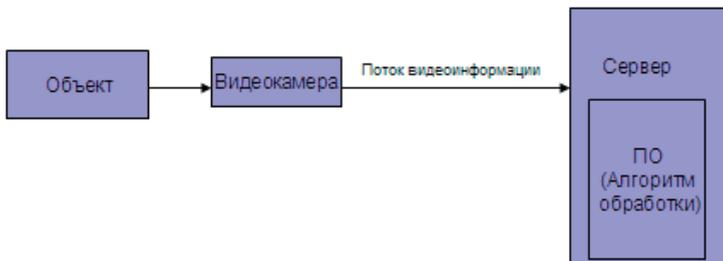


Рис. 1. Общая схема системы видеоанализа

Проблема разработки видеоаналитических систем:

- отсутствие подробной информации о технологиях разработки;
- неадекватность имеющихся открытых алгоритмов видеоаналитики «реального времени»;
- непереносимость разработок на другие платформы.

Одним из путей решения проблемы является:

- разработка альтернативных алгоритмов видеоаналитики real-time;
- использование технологии OpenCv.

OpenCV (англ. Open Source Computer Vision Library) – библиотека алгоритмов компьютерного зрения, обработки изображений и численных алгоритмов общего назначения с открытым кодом. Данная библиотека реализована на языке C/C++, также разрабатывается для Python, Ruby, Matlab, Lua и других языков. OpenCv содержит более 500 функций, предназначенных для выполнения в реальном времени.

Идея OpenCv:

- исследование машинного зрения, разработка и оптимизация кода;
- распространение информации по машинному зрению, разработка общей инфраструктуры на которой могли бы основываться разработчики;
- приложения должны быть переносимыми, оптимизированными, код которых не обязательно должен быть открытым.

При помощи технологии OpenCv мной были реализованы основные алгоритмы компьютерного распознавания: алгоритмы «попиксельного» сравнения, контурного анализа, сравнения гистограмм. Реализованные алгоритмы были протестированы по следующим показателям: возможность статического и динамического распознавания, время обработки одного кадра, ошибки при распознавании, обработка поворотов и масштабирования, восприятия естественных помех, загрузка процессора и оперативной памяти машины. Опытным путем было установлено, что вышеперечисленные алгоритмы не могут работать в режиме real-time из-за большого времени обработки одного кадра видео картинки. Помимо этого, исследуемые алгоритмы сильно загружают процессор и оперативную память машины. Анализ показал, что это происходит из-за неоптимальной обработки информации внутри самих алгоритмов. Также невозможность обработки поворотов и масштабирования сильно сужает область применения данных алгоритмов. Для создания адекватной видеоаналитической системы необходимо использовать алгоритм, который будет обрабатывать картинку в реальном времени с учетом выявленных недостатков. Таким образом, появилась потребность в разработке альтернативного алгоритма обработки видео информации.

При помощи технологии OpenCv мной был разработан альтернативный алгоритм real-time видеоаналитики. Алгоритм позволяет распознавать образы, детектировать движения и определять угол поворота наблюдаемого объекта (распознанного образа). Разработанный алгоритм обладает рядом преимуществ по сравнению с исследованными алгоритмами. Сравнительный анализ с имеющимися алгоритмами видеоанализа представлен в табл. 1.

Таблица 1

Сравнительная характеристика разработанного алгоритма с имеющимися аналогами

Алгоритм «попиксельного» сравнения	A			
Алгоритма контурного поиска	B			
Алгоритма сравнения гистограмм	C			
Разработанный алгоритм	D			

Таблица 1 (окончание)

	A	B	C	D
Статическое распознавание	+	+	+	+
Динамическое распознавание	-	+	-	+
Время обработки одного кадра,мс	233	250	450	57
Загрузка процессора IntelPentium(R)1.7, %	70	60	90	25-30
Ресурсопотребление,оперативная память, мб	40	80	400	50
Ошибки при статическом распознавании, %	1-2	2-3	1-2	1-2
Ошибки при динамическом распознавании,%	97-100	20-30	100	1-2
Обработка поворотов и масштабирования	-	+/-	-	+
Воздействие естественных помех	+	+	+	-

В заключение необходимо отметить основные преимущества OpenCv:

- открытость;
- кроссплатформенность;
- поддержка множества языков программирования;
- поддержка многих сред разработки;
- множество алгоритмов для обработки видеопотока;
- оснастки для разработки распознавания образов;
- встроенные средства калибровки видеокамер (данная задача является довольно сложной и поэтому решение ее в библиотеке дает большое преимущество при разработке);
- имеются средства оценки точности распознавания образов (тестовые наборы объектов для определение функциональности распознавания).

Библиографический список

1. <http://ru.wikipedia.org>
2. Bradsky G., Kaehler A. Learning OpenCV. O'Reilly, 2008. С. 1 (ISBN 978-0-596-51613-0).

УДК 004.92

Н. А. Лиманский – студент кафедры вычислительных систем и сетей

А. В. Никитин (канд. техн. наук, доц.) – научный руководитель

ПРИМЕНЕНИЕ ТЕХНОЛОГИИ ДОПОЛНЕННОЙ РЕАЛЬНОСТИ В МЕДИЦИНЕ

В настоящее время технология дополненной реальности становится все более популярной. Основное предназначение дополненной реальности это оказание помощи в решении каких-либо задач. Ее применение мы можем наблюдать во многих областях, начиная от рекламы и компьютерный игр и заканчивая применением в военном деле. Однако есть такие области, которых технология дополненной реальности еще пока не коснулась в России. Одна из таких областей – это медицина.

Дополненная реальность (ДР) это дополнение восприятия реального мира виртуальными, компьютерно-генерируемыми объектами. В дальнейшем мы будем рассматривать маркерную дополненную реальность. Слово «маркерная» обозначает то, что

виртуальный объект будет формироваться при распознавании определенного маркера и привязываться к нему.

Технологи дополненной реальности была примененная для решения задачи, связанной с хирургией. Эта задача состоит из 2-х этапов:

- реализация приложения на основе технологии ДР для обучения хирургов;
- реализация приложения на основе технологии ДР для внедрения в клиническую практику.

На подготовительном этапе был проведен анализ проблем традиционных медицинских технологий, а также проблем проектирования и проведения операций, и возможности их решения при помощи технологии ДР. Также был проведен анализ платформ ДР, и была выбрана платформа, полностью удовлетворяющая требованиям.

Была поставлена задача реализовать управляемый поэтапный показ сборки аппарата Илизарова на голени на базе интерактивной анимированной 3D модели анатомического органа (голень – кожа, нервы, сосуды, кость), аппарата Илизарова и хирургических инструментов с синхронной (соответствующей рассматриваемому этапу) выдачей справочной информации – текстовой, видео, аудио, изображений срезов кости и томографа.

Сначала моделируется вся операция поэтапно в трехмерной среде, включая анатомический орган, хирургические инструменты и медицинские конструкции (аппарат Г. А. Илизарова) (рис. 1) [1].

Затем, используя специальные программные продукты, позволяющие реализовывать ДР, осуществляется привязка моделей к маркеру, который приложив, например, к ноге реального пациента (к тому месту, где будет делаться данная операция), мы сможем увидеть и управлять протеканием операции в трехмерной среде.

Ниже приведена концепция создания приложения.

Создание приложения делится на 3 раздела.

Первый раздел включает в себя моделирование 3D объектов в соответствии с требованиями (текстурирование, освещение, малое число полигонов).

Второй раздел включает в себя создание 2D графики. Под 2D графикой понимается рисование всех кнопок, используемых в приложении. Также оформление всей графи-

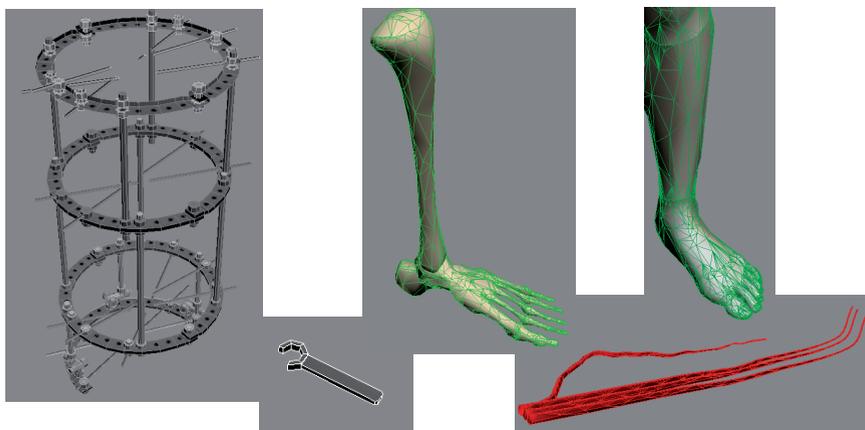


Рис. 1. 3D модели

ческой информации, которая будет представлена в приложении. Все 2D объекты тоже должны соответствовать определенным требованиям (формат, размер, глубина цвета). К 2D графике относится рисование текстур для 3D объектов. Текстуры тоже должны удовлетворять определенным требованиям.

Основная задача третьего раздела – это программирование. Написание основного кода, отвечающего за реализацию дополненной реальности. Также подключение при помощи программирования всю созданную 2D и 3D графику. [2]

На рис. 2–4 можно увидеть готовое приложение на основе технологии дополненной реальности для обучения хирургов.

Приведенный пример прототипа системы обучения хирурга на основе ДР на платформе FLARToolKit, разработанный по заказу Российского научного центра «Восстановительная травматология и ортопедия» имени академика Г. А. Илизарова. [3, 4]

Работа была выполнена в соответствии с поставленной задачей в полной мере. Полученное приложение можно использовать для начального образования хирургов. Также во время выполнения работы экспериментальным путем были получены количественные и качественные ограничения, касающиеся выбранной платформы. Также были получены данные, необходимые для выбора более профессиональной платформы для реализации приложения на основе технологии дополненной реальности для внедрения в клиническую практику.

Библиографический список

1. http://www.render.ru/books/index.php?book_cat=5 (3D моделирование)
2. http://help.adobe.com/ru_RU/AS3LRC/Flash_10.0/ (справочник по программированию)
3. <http://www.libspark.org/wiki/saqoosha/FLARToolKit/en> (инструментарий)
4. <http://www.flash.ru> (Flash плеер)

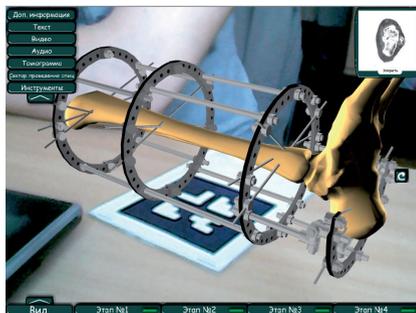


Рис. 2. Модель кости и собранного аппарата Илизарова

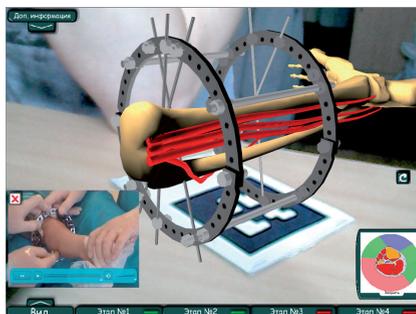


Рис. 3. Модель кости, сосудов



Рис. 4. Модель ноги

И. С. Ляпунов – магистрант кафедры вычислительных систем и сетей

В. А. Гордеев (д-р техн. наук, проф.) – научный руководитель

СИСТЕМЫ РАННЕГО ПРЕДУПРЕЖДЕНИЯ ПРИБЛИЖЕНИЯ К ЗЕМЛЕ ТТА-12 НА ЗАКЛЮЧИТЕЛЬНОМ ЭТАПЕ ПОСАДКИ САМОЛЕТА

На современном этапе развития авиации проблема повышения точности и достоверности как навигационной, так и пилотажной информации, стала приоритетной в обеспечении безопасности полетов. Данная проблема тесно переплетается также с деятельностью экипажа, который принимает решение по результатам анализа значительного объема информации, часто в условиях как дефицита времени, так и неточной информации. Это особенно относится к этапу посадки воздушного судна (ВС), являющегося самым трудным и ответственным этапом полета. Трудность этапа посадки ВС, прежде всего, обуславливается необходимостью его осуществления с высокой точностью. Системы обеспечения точного захода на посадку (ILS) хорошо изучены и доступны на рынке, например системы КУРС МП, VIM-95, NAV-4000. Также доступны системы точного захода на посадку на базе спутниковых систем. Однако данные системы обеспечивают выдерживание безопасной траектории лишь до высоты принятия решения, как правило, 60 м. Лишь наиболее крупные аэродромы в России оснащены системами захода на посадку по категориям II и IIIA, где обеспечивается высота принятия решения 30 м. Ниже этой высоты выполнение посадки осуществляется экипажем в ручном режиме. Таким образом, разработка системы предупреждения экипажа на заключительных участках захода на посадку, в том числе, о положении самолета относительно ВПП является актуальной. Для безопасного пробега ВС по взлетно посадочной полосе (ВПП) необходимо иметь достоверную и точную информацию о следующих параметрах: высота и координаты ВС, скорость полета, точное определение направления посадки на ВПП, метеорологическая обстановка и т. п.

Существуют две сертифицированные системы TAWS российского производства (СРПБЗ – ВНИИРА-Навигатор, ТТА-12(S) – ЗАО «Транзас»), а также ряд зарубежных систем (TAWs – Universal, ST3400 – Sandel, EGPWS – Honeywell) [1]. Однако ни в одной из перечисленных систем нет режима, отвечающего за безопасность на заключительном этапе посадки – пробег. Система раннего предупреждения приближения к земле (СРППЗ) ТТА-12, которая информационно связана со спутниковой навигационной системой АБРИС, имеет выходы на сигнально-переговорное устройство (СПУ), сигнально-громкоговорящее устройство (СГУ) и лампу «ЗЕМЛЯ». Система ТТА-12 имеет встроенную энергонезависимую память, позволяющую хранить заранее подготовленные звуковые файлы, а также съемный накопитель с базой данных, на которой содержится информация об аэродромах и их ВПП. Также СРППЗ ТТА-12 имеет встроенные органы управления, которые могут позволить управлять работой разрабатываемой системы [2]. Учитывая характеристики СРППЗ ТТА-12 и ее информационные связи с датчиками самолета, представляется возможным разработать систему оповещения экипажа о положении самолета относительно ВПП путем доработки программно-математического обеспечения СРППЗ. Система предупреждения о положении самолета относительно ВПП должна быть предназначена для своевременной выдачи экипажу уведомляющих, предупреждающих сообщений для предупреждения предпосылок выкатывания при заходе на посадку и взлета с неверной полосы.

Основные режимы работы разрабатываемой системы:

1. Уведомление о приближении к ВПП.
2. Предупреждение о выруливании на ВПП.
3. Предупреждение о подходе к концу ВПП.
4. Предупреждение о попытке взлета с рулежной дорожки.

5. Предупреждение об оставшемся расстоянии при прерванном взлете.
6. Предупреждение об оставшемся расстоянии при пробеге после посадки.
7. Предупреждение о недостаточном остатке ВПП при пробеге.
8. Предупреждение о попытке захода на посадку на короткую полосу (в воздухе).

Предлагаемый алгоритм работы системы для реализации 7 режима, предназначенно-го для оповещения экипажа о необходимости более активного торможения, в случае если система обнаруживает предпосылку к выкатыванию самолета за пределы полосы.

Путевая скорость (V_n) на этапе пробега определяется по формуле:

$$V_n = V_{\text{noc}} - a \times t, \quad (1)$$

где: V_{noc} – скорость во время касания земли (м/с),

a – ускорение (м/с²),

t – время проката в секундах.

Так как в момент остановки ВС путевая скорость равна нулю:

$$0 = V_{\text{noc}} - a \times t, \quad (2)$$

$$t = \frac{V_{\text{noc}}}{a}. \quad (3)$$

Расстояние, которое предположительно преодолет самолет до достижения заданной скорости при условии сохранения текущей путевой скорости и ускорения, начиная с данного момента времени ($S_{\text{ост}}$)

$$S_{\text{ост}} = V_{\text{noc}} \times t - \frac{a \times t^2}{2}. \quad (4)$$

Учитывая формулу (3), ($S_{\text{ост}}$) выражается как

$$S_{\text{ост}} = \frac{V_{\text{noc}}^2}{2 \times a}. \quad (5)$$

Из формулы (5) видно, что для уменьшения длины пробега необходимо уменьшать посадочную скорость (V_{noc}) или увеличивать тормозящие силы. Увеличение последней производится за счет применения тормозных устройств на колесах шасси, интерцепторов и реверса двигателей. Торможение колесами на высоких скоростях является не только не эффективным и опасным, но к тому же приводит к быстрому износу покрышек и тормозной системы самолета. Поэтому сразу после касания самолета, торможение обеспечивается за счет включения реверсивной тяги двигателей и за счет механизации самолета – интерцепторов. После гашения скорости примерно до 150 км/ч, реверс двигателей выключается, и торможение осуществляется за счет дисковых тормозных устройств. В некоторых случаях аварии можно было бы избежать, если бы экипаж был бы своевременно оповещен, что оставшаяся длина ВПП является не достаточной, и он бы не выключил реверсивную тягу двигателей. Повторный запуск реверса на самолете возможен, но цикл запуска занимает порядка 10 секунд, что крайне много в условиях дефицита длины полосы. На встроенном накопителе данных СРППЗ ТТА-12 имеется аэронавигационная база с мировым охватом, и содержится информация по всем ВПП (с длиной более 2 км). Эта информация и знание текущей местоположения самолета позволяет в реальном масштабе времени вычислять удаление самолета от торца ВПП, на которую была осуществлена посадка.

$$S = \sqrt{(\Delta\lambda^2 - \Delta\varphi^2)}, \quad (6)$$

где

$$\Delta\lambda = (\lambda_c - \lambda_{\text{впп}}) \frac{\pi}{180} R_3 \sin\varphi; \quad (7)$$

$$\Delta\varphi = (\varphi_c - \varphi_{впп}) \frac{\pi}{180} R_3; \quad (8)$$

S – удаление самолета от торца ВПП;

λ_c, λ_{c-} – текущие координаты ВС, измеряются в градусах;

$\lambda_{впп}, \varphi_{впп}$ – координаты порога ВПП считываемые из БД, измеряются в градусах;

R_3 – радиус Земли, измеряется в метрах.

Далее путем сравнения расстояний, определенных по формулам 5 и 6, принимается решение о наличии опасности, если разница между S и Sост меньше пороговой величины.

Вычисления, выполненные на основе полетных записей, регистрируемых системой ТТА-12 на своем встроенном накопителе. В процессе работы фиксируются текущие координаты, высота, скорость и другие параметры самолета.

Для вычислений была сделана выборка по путевой скорости, принимаемой от спутниковой системы. На основании этой выборки было оценено оставшееся расстояние до остановки самолета. Для удобства анализа на графике (рис. 1) отложено:

- по оси X – время в секундах от момента касания самолета;
- по оси Y:
- расстояние до остановки при текущей путевой скорости и ускорении торможения (Sост, км);
- расстояние, пройденное от момента касания (S, км);
- суммарное расстояние, пройденное с начала приземления плюс расстояние до остановки (Ssum, км);
- длина ВПП.

В дальнейшем планируется разработать аналитическую модель безопасного прогона самолета, которая учитывала бы важные параметры, такие как: метеорологические условия, структуру покрытия ВПП (для определения коэффициента сцепления), наличие бокового ветра и т.д. Также стоит проблема фильтрации данных для устранения ситуаций ложного срабатывания предупреждающих сигнализаций при работе режима. Определение порогового значения по дальности от торца ВПП также является открытой задачей.

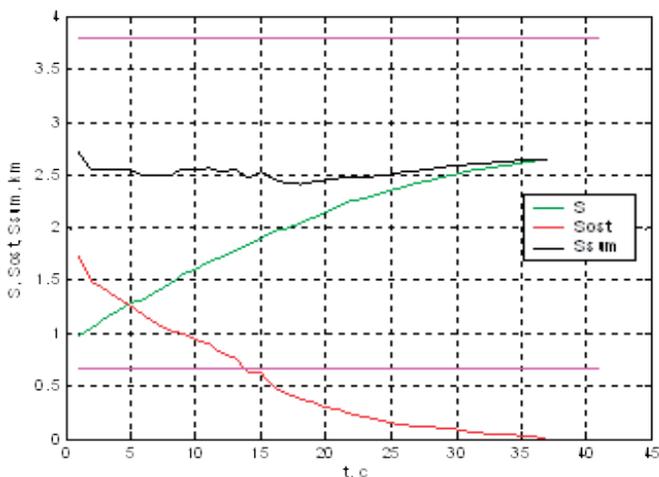


Рис. 1

Библиографический список

1. Дрягин Д.М. Комплексная система раннего предупреждения приближения к земле с расширенными функциональными возможностями и программно-алгоритмические средства, минимизирующие средства ложной сигнализации: Диссертация к. т. наук: 12.06.08 / Моск. авиа. ин-т. М., 2006. 5 с.

УДК 004.4.236

К. В. Малеев – магистрант кафедры вычислительных систем и сетей
В. А. Галанина (канд. техн. наук.) – научный руководитель

NETBEANS – ОТКРЫТАЯ СРЕДА РАЗРАБОТКИ

В последние годы значимость и необходимость использования инженерных систем безопасности многократно возросла, и их применение стало практиковаться не только на государственных и общественных объектах, но и в частном секторе. Современные системы безопасности имеют сложную структуру и включают в себя сразу несколько подсистем (охранно-тревожную сигнализацию, пожарную сигнализацию, систему управления и контроля доступа, систему видеонаблюдения и другие).

Для хранения столь различных данных используются объектные базы данных.

Для разработки системы используются среды разработки приложений.

NetBeans IDE — свободная интегрированная среда разработки приложений (IDE) на языках программирования Java, JavaFX, Ruby, Python, PHP, JavaScript, C++, Ада и ряде других.

NetBeans IDE доступна в виде готовых дистрибутивов (прекомпилированных бинарных файлов) для платформ Microsoft Windows, Linux, FreeBSD, Mac OS X, OpenSolaris и Solaris (как для SPARC, так и для x86 — Intel и AMD). Для всех остальных платформ доступна возможность скомпилировать NetBeans самостоятельно из исходных текстов.

Сборки NetBeans для сообщества доступны по двойной лицензии, состоящей из Common Development and Distribution License (CDDL) v1.0 и GNU General Public License (GPL) v2.

1. Способы установки

Существуют несколько способов установки NetBeans в операционной системе Ubuntu 10.10:

- установка из центра приложений Ubuntu. Для этого нужно выполнить: Приложения – Центр Приложений Ubuntu – Программирование – NetBeans;
- установка с помощью менеджера пакетов Synaptic или apt. Откройте терминал и наберите: `sudo apt-get install netbeans`;
- установка ранее скаченного исполняемого файла.

2. NetBeans и объектная модель данных

Объектная модель данных, интегрируя все возможности реляционной модели, обладает рядом существенных преимуществ по сравнению с последней. Эти преимущества заключаются, прежде всего, в способности естественным образом интегрировать в рамках единого информационного массива разнородные данные.

MongoDB – документо-ориентированная система управления базами данных (СУБД) с открытым исходным кодом, не требующая описания схемы таблиц. Написана на языке C++.

СУБД управляет наборами JSON-подобных документов, хранимых в двоичном виде в формате BSON. Подобно другим документо-ориентированным СУБД (CouchDB, пр.), MongoDB не является реляционной СУБД.

3. Использование системы NetBeans при разработке баз данных

3.1. Подключение к реляционной базе данных

В реляционной модели данных объекты и взаимосвязи между ними представляются с помощью таблиц. Взаимосвязи также рассматриваются в качестве объектов. Каждая таблица представляет один объект и состоит из строк и столбцов. Благодаря своей простоте и естественности представления реляционная модель получила наибольшее распространение в СУБД для персональных компьютеров.

NetBeans позволяет легко подключить реляционную базу данных.

Нажмите на вкладку Сервисы-Базы данных. Выберите из списка нужную базу данных, нажмите Подключиться.

3.2. Подключение к объектной базе данных

Рассмотрим использование NetBeans и объектной базы данных на примере MongoDB.

Необходимо подключить нужные библиотеки с помощью команды `sudo pecl install mongo`.

В файле `/etc/php5/cli/php.ini` необходимо добавить строку `extension=mongo.so`.

Создадим подключение к базе данных, заполним данными из коллекции, содержащей данные о замках, и выведем результат работы на экран. Фрагмент программы приведен ниже.

```
<?php
$m=new Mongo();
$db=$m->site;
$collection=$db->Zamki;
$obj=array(«Name»=>«KN12»,«Proizvoditel»=>Irbis»,«Massa»=>»2»,«Price»=>15000,«Opisanie»=>»Надежный замок по выгодной цене.»);
$collection->insert($obj);
$cursor=$collection->find();
foreach ($cursor as $obj)
echo $obj[«Name»]. «\n»;
echo $obj[«Price»]. «\n»;
?>
```

В результате работы программы получим результат: KN12 15000.

Использование системы NetBeans для организации нагрузочного тестирования

Для того чтобы знать, как поведет себя база данных при больших объемах данных, необходимо «нагрузить» ее тестовыми данными.

В представленной ниже программе мы создаем базу данных, заполняем ее данными и замеряем время при помощи утилиты `mongostat`.

```
import random
import json
import time
from pymongo import Connection
connection=Connection()
connection.drop_database('newtesttimeFire')
database = connection['newtesttimeFire']
collection = database['Systema_Ing_Ukr']
time.sleep(5);
for i in range(500000):
dot_x = i
dot_y = random.randint(1, 16)
collection.save({'Zamki1': {'Name', 'test', 'Proizvoditel', 'Ypakovka', 'Massa', 'Price',
dot_y*140, 'Opisanie', ''}});
for doc in collection.find({'Zamki.Price': {'$gt': 100}}):
print doc;
```

Время выполнения программы при добавлении в базу 500 тысяч записей составляет 1205 с.

Заключение

Открытая среда разработки NetBeans сочетает в себе мощное и удобное средство для разработки программного обеспечения.

С помощью данной системы проведено нагрузочное тестирование базы данных. Полученные результаты легли в основу дипломной работы.

УДК 004.65

М. А. Малофеева – студентка кафедры моделирования вычислительных и электронных систем

Ю. В. Щеткина – научный руководитель

СИСТЕМА УЧЕТА МЕМОРИАЛЬНЫХ ДОСОК И ПАМЯТНИКОВ

В нашем городе насчитывается около 2000 мемориальных досок (далее МД) и более 5000 памятников, в том числе на территории Петродворцового района зарегистрировано около 1200 мемориальных досок и памятников. Все памятники и МД зарегистрированы в общегородской реестр и реестр района. База данных разрабатывается для отдела культуры администрации Петродворцового района. До этого сведения хранились в отдельных текстовых документах, поэтому у каждого из объектов есть два регистрационных номера, по которым их можно найти.

Одно из главных условий установки новой мемориальной доски или памятника – с момента события или момента смерти выдающейся личности должно пройти не менее 30 лет. Но возможны исключительные случаи (Почетные граждане Санкт-Петербурга).

Основаниями для принятия решения об установке мемориальной доски или памятника являются: признание значения свершившегося события для развития общества и культуры, наличие особого вклада выдающейся личности в определенную сферу деятельности, официально признанные достижения в науке, технике, литературе, искусстве, культуре и спорте, государственной, общественной, политической, военной и производственной деятельности.

Для каждого регистрируемого объекта записываются следующие характеристики:

- дата создания памятника или МД, которая может быть точной, если есть соответствующие подтверждающие исторические документы, или приблизительной, если таких исторических документов нет, а известен только период;

- автор – создатель объекта (архитектор, скульптор или изготовитель МД).

- материал. Указывается вид материала, из которого изготовлен памятник, МД или скульптура. Это важно для формирования плана реставрации и восстановления исторических скульптур. Именно видом материала характеризуются сложность реставрационных работ и частота их проведения, ведь большинство мемориальных досок сделаны из белого мрамора, который очень легко приходит в негодность от условий городской среды;

- номер постановления и его дата. Установка мемориальных досок и памятников производится на основании постановлений Правительства Санкт-Петербурга (на каждый конкретный памятник). А общие положения по установке, обеспечению сохранности и демонтажу закреплены в постановлении Правительства Санкт-Петербурга от 17.01.2005 № 2 «О мемориальных досках и памятниках в Санкт-Петербурге». Чтобы в реестре можно было легко найти памятник по постановлению, указывается номер постановления и его дата. На конкретный памятник может приходиться несколько постановлений, связанных

с ним. Это может происходить в результате перестановки памятника на другое место в пределах района или города;

- местонахождение памятника. Оно может меняться в связи с постановлением администрации. При переезде памятника в базу данных заносится новый адрес, а старый остается в виде справки, чтобы можно было проследить историческое местоположение скульптуры;

- категория охраны. Каждый культурный исторический памятник имеет категорию охраны. Существует три типа данного вида: памятник федерального значения, памятник регионального значения и памятник под охраной ЮНЭСКО. Эта информация характеризует памятник по исторической значимости для государства;

- фотография. При регистрации объекта должна быть предьявлена его фотография, и в регистрационной записи должно быть указано, где ее можно найти;

- организация, за которой объект закреплен (памятник или МД). Информация о такой организации должна содержать ее название, адрес, телефон и дату назначения. Данные характеристики нужны, чтобы найти памятник, если его перевозили на новое место, или узнать о состоянии объекта на данный момент. Ответственная организация является государственным учреждением. Она отвечает за сохранность объекта, постановку на реконструкцию, следит за работами над объектом. Но чаще всего памятные объекты находятся в бесхозном состоянии. Они, как правило, находятся в аварийном или не подлежащем восстановлению состоянии. Поэтому каждый памятник и МД ставятся закрепить за государственной организацией;

- техническое состояние материала и объекта в целом. Все это определяется с помощью ответственной организации, за которой закреплен объект. Эта информация позволяет увидеть памятники, требующие немедленной реставрации. Также, благодаря этой информации, можно составить список объектов, которые нужно поставить на реставрационные и плановые работы;

- дата последнего изменения записи о памятном объекте. Это необходимо для отслеживания каких-либо изменений состояния и местонахождения объекта.

1. Требования к работе с базой данных

При работе с базой данных необходимо выполнение следующих основных запросов:

1. Запрос «По техническому состоянию памятников и мемориальных досок». Ответом на данный запрос должен быть список объектов, имеющих аварийное и подлежащее реставрации состояние. При выполнении запроса указываются следующие характеристики объекта: материал, техническое состояние, номер, название объекта.

2. Запрос «Справка». В ответ на данный запрос должны быть предоставлены следующие сведения для каждого объекта: номер по району, название. Используется информация о постановлении, об ответственной организации, а также адрес объекта. Такой запрос необходим для вывода справки по любому памятнику: список всех адресов, номера постановлений по конкретному памятнику, список всех организаций, за которыми он был закреплен. Данная информация может понадобиться при перевозке памятника на новое место, для подробных справочных данных, для уточнения или восстановления необходимых сведений об объекте.

3. Запрос «Историческая справка» выводит информацию о местоположении, авторе, дате создания, а также фотографию. Данного вида информация предоставляется в музеи, культурные центры и справочные издания.

2. Проектирование базы данных

Для данного случая может быть использована реляционная база данных (далее БД). Такой выбор обосновывается тем, что реляционные БД наиболее развиты в настоящее время и поддерживаются большим количеством программных продуктов, что позволяет сократить сроки и стоимость проектирования.

В реляционной БД все сведения представляются в табличном виде. Для рассматриваемого случая предлагается следующий вид (табл. 1).

Перечень памятников и мемориальных досок

Название поля	Свойства поля
«№ по порядку» – ключ	Тип поля – счетчик
«№ общегородской»	Тип поля – числовой
«№ по району»	Тип поля – числовой
«Наименование памятника или доски»	Тип поля – текстовый
«Дата создания»	Тип поля – текстовый, необязательно для заполнения
«Материал»	Тип поля – текстовый, список
«Местонахождение»	Тип поля – текстовый
«Категория охраны»	Тип поля – текстовый, список, необязательно для заполнения
«№ постановления»	Тип поля – числовой
«Дата постановления»	Тип поля – дата
«Автор»	Тип поля – текстовый, необязательно для заполнения
«Фотография»	Тип поля – текстовый, необязательно для заполнения
«Ответственная организация»	Тип поля – текстовый, необязательно для заполнения
«Адрес ответственной организации»	Тип поля – текстовый, необязательно для заполнения
«Телефон ответственной организации»	Тип поля – числовой, необязательно для заполнения
«Дата назначения ответственного»	Тип поля – дата, необязательно для заполнения
«Дата вставки записи»	Тип поля – дата
«Техническое состояние»	Тип поля – текстовый, список, необязательно для заполнения

Все запросы могут быть обращены к одной таблице, т. е. БД может быть одно-табличной. В теории реляционных БД часто рекомендуется нормализация таблиц. Но в данном конкретном случае нормализация не дает видимых преимуществ по следующим причинам:

1) уменьшение избыточности данных здесь неактуально, так как общий объем таблицы невелик и будет расти со временем очень медленно;

2) выполнение модификации данных в главных таблицах схемы данных здесь также неактуально, поскольку вместо модификации данных целесообразно просто вставить новый вариант записи, указав дату вставки записи. Так как число требуемых модификаций невелико (не больше 1000 в год), то это не приведет к быстрому переизбытку таблицы, но ускорит выполнение запросов для получения информации об изменениях, касающихся конкретных памятников.

3. Реализация базы данных

Данную БД целесообразно реализовать в СУБД Access, так как основные рабочие места должны быть сосредоточены в пределах одной локальной сети. Кроме того, объем БД невелик, и количество пользователей не больше пяти. Поэтому может быть использован вариант разделения БД между несколькими пользователями локальной сети, предусмотренный в последних версиях СУБД Access.

В пользовательском интерфейсе должны быть предусмотрены средства для работы администратора БД, а именно: резервное копирование, расширение функциональных возможностей за счет вставки программных модулей, а также средства для решения трех главных пользовательских задач:

- вставка записей в таблицу;
- поиск нужной информации по различным условиям фильтрации;
- составление отчетных документов на базе на основе запроса-выборки.

Перечисленные задачи легко реализуются в СУБД Access с помощью форм, отчетов и средств программирования.

Заключение

Спроектированная БД вполне удовлетворяет текущим потребностям. Ее использование требует только наличия локальной сети и пакета MS Office на рабочих местах. При необходимости сведения из этой БД можно легко экспортировать в различные приложения пакета MS Office, например в Word. Кроме того, эта БД может быть легко интегрирована в более широкую информационную систему через SQL-сервер: MS SQL, MySQL и т. п.

Библиографический список

1. Дейт К. Дж. Введение в системы баз данных – 8-е изд. М.: Вильямс, 2006.
2. Карпова Т. С. Базы данных: модели, разработка, реализация. СПб.: Питер, 2002.
3. Преснякова Г. В. Проектирование интегрированных реляционных баз данных. КДУ, 2007.
4. Мирошниченко Г. А. Реляционные базы данных: практические приемы оптимальных решений. СПб.: БХВ-Петербург, 2005.

УДК 004.732

Н. И. Матвиенко – студент кафедры вычислительных систем и сетей

А. В. Гордеев (д-р техн. наук, проф.) – научный руководитель

ПРОТОКОЛ DNSSEC

Практически 25 лет DNS-запросы не считались безопасными, хотя, по сути, служба DNS является фундаментом Интернета. К ее услугам обращаются практически все остальные популярные сервисы, предназначенные для отправки почты, веб-серфинга, VoIP, ICQ и так далее. Соответственно, если DNS выдает неправильную информацию, это сказывается на работе любого сервиса, его запрашивающего. Но после внедрения DNSSEC на корневых серверах эта проблема будет решена.

Основная идея внедрения DNSSEC – проверка подлинности полученных данных при помощи цифровой подписи, для чего задействуется механизм шифрования с открытыми ключами. Одним из преимуществ DNSSEC является иерархическая структура, при которой информация не хранится в одном месте, а на многочисленных серверах и запрашивается по мере необходимости. Особенностью реализации протокола в сети Интернет является необходимость задействования всей имеющейся иерархической структуры. Таким образом, администратор верхнего уровня выступает гарантом, удостоверяющим

подпись доменов, находящихся на уровне ниже. В результате клиент, получающий адрес, последовательно может проследить цепочку доверия к открытому ключу, начиная от самого верхнего сервера root/TLD к серверу, отвечающему за конкретную зону [1].

Самое главное – DNSSEC не является заменой DNS. Протокол предлагает подписывать не каждую отдельную запись, а все множество ресурсных записей (RRSet). Его внедрение потребовало внесения некоторых изменений в DNS-протокол, в частности в ресурсные DNS-записи: Подписанная зона включает в себя: запись общедоступного ключа DNS (DNSKEY), ресурсную запись подписи (RRSIG), ресурсную запись перечисления типов записей, существующих в данном домене (NSEC3), и ресурсную запись, формирующую цепочку аутентификации между DNS-зонами (DS). Зона, которая не имеет этих ресурсных записей, считается неподписанной. Также протокол потребовал новых полей в заголовке DNS-сообщения: Checking Disabled (CD) и Authenticated Data (AD). Данные биты созданы для коммуникации между безопасными распознавателями и безопасными рекурсивными серверами имен [2, 3].

DNSSEC определяет создание и проверку цифровых подписей с использованием пары асимметричных ключей. Один тип ключа называется Key Signing Key – KSK закрытый и открытый ключ, который будет использоваться только для подписания ключа, содержащегося в зонном файле в ресурсной записи с типом DNSKEY. Другой тип ключа называется Zone Signing Key – ZSK ключ (также открытый и закрытый), который используется для подписания всех множеств ресурсных записей в зоне включая множество ресурсных записей DNSKEY. Ключ KSK должен обновляться менее часто, чем ключ ZSK. Рекомендуемая частота обновления для ключа KSK равна одному году (при использовании RSA/MD5 с длиной ключа 2048 бит), а ключ ZSK должен обновляться каждый месяц (при использовании RSA/MD5 с длиной ключа 1024 бит) [1, 2].

OpenDNSSEC – программное обеспечение с открытым исходным кодом, которое облегчит развертывание безопасного протокола (DNSSEC) в системе доменных имен (DNS). OpenDNSSEC доступно для всех Unix-подобных операционных систем и подходит для работы как с большими зонами (например, домены верхнего уровня), так и с множеством малых (хостинг, ISP).

Особенности данного программного обеспечения:

- возможность подписания зон, содержащих от нескольких до миллионов записей;
- возможность создания универсального конфигурационного файла «Политики», который может распространяться как на одну, так и на множество зон;
- хранение важных криптографических данных в HSM, общаясь с ним, используя стандартный PKCS#11 интерфейс;
- поддержка SoftHSM – программной эмуляции HSM;
- использование RSA/SHA1 и SHA2 подписей, а также системы распознавания NSEC3.

HSM – аппаратный криптографический модуль (hardware security module), который предназначен для обеспечения безопасного хранения и использования закрытого ключа, а также для выполнения всех криптографических операций, в том числе генерации ключа в криптомодуле.

Производительность авторитарного и рекурсивного DNS сервера будет зависеть от таких параметров как длина ключа и количество хеш-итераций. В случае использования программной эмуляции криптографического хранилища – SoftHSM скорость работы будет зависеть от процессора данного dns-сервера. Кол-во предоставляемых ресурсов можно узнать, используя утилиту ods-hsmspeed. Поэтому основной проблемой при использовании SoftHSM является недостаточная производительность. Оптимизировать работу системы можно распараллеливанием процесса подписания зон.

Для решения данной задачи было решено создать вычислительный кластер на основе MPICH2 – легко портируемой быстрой бесплатной реализацией стандарта MPI. MPI – это программный интерфейс для передачи информации, который позволяет обмениваться сообщениями между процессами, выполняющими одну задачу. Система OpenDNSSEC

состоит из двух демонов: `ods-signer` и `ods-enforcer`. Первый демон – это компонент системы, который выполняет подписание зон. Сначала он анализирует информацию в файле `zonelist.xml` и далее переходит по зонам, перечисленным в данном файле, и подписывает их в случае, если зона еще не подписана или истек срок действия подписи. Запуск и остановка демона выполняется с командной строки, что позволяет нам запустить его в среде MPI и распараллелить процесс подписания зон. Особенно будет заметен прирост производительности системы в момент добавления нового списка зон во время работы OpenDNSSEC [4].

Таким образом, для создания собственной изолированной подписанной зоны протоколом DNSSEC можно ограничиться использованием лишь SoftHSM, организовав работу на вычислительном кластере, и тем самым добиться высокой производительности при работе с криптографическими операциями и избежать нагрузок на `dns`-сервер.

В случае если необходимо подписать очень большую зону (около полумиллиона записей) или несколько таких зон, устанавливается HSM. При этом из-за большого количества криптографических операций при первой подписи такого масштаба зоны ресурсы системы могут быть исчерпаны, поскольку каждый экземпляр Аудитора (функциональный компонент OpenDNSSEC) использует свой интерпретатор Ruby VM. Данную задачу можно решить очень просто – необходимо в файле-конфигурации политики в поле настройки параметров «Аудита» вписать «<Audit> <Partial />» на момент подписания зон. Частичный Аудит – это упрощенная проверка зоны, т.е. не проверяется каждая запись RRset криптографической подписи, но с полным функциональным набором действий [2, 5, 6].

После сравнительного анализа имеющихся HSM в роли хранилища ключей было выбрано Luna PCI. Это аппаратный модуль безопасности (PCI-карта), обеспечивающий все стандартные криптографические функции, а также оптимизацию и повышенную производительность при выполнении асимметричных операций (ЭЦП, PKI). Данный модуль позволит работать с большим количеством зон с высокой скоростью обработки DNS пакетов. OpenDNSSEC располагается между внутренним и публичным DNS-серверами, подписывая все проходящие ответы с помощью SafeNet Luna PCI HSM, используя криптографический стандарт PKCS #11, именуемый также Cryptoki. Ключевые характеристики HSM:

- поддерживаемые алгоритмы: RSA (до 4096 бит), SHA-256/512, AES, 3DES и др.;
- встроенное хранилище на 1280 криптообъектов (например, 1280 сертификатов или 640 пар ключей);
- поддерживается крипто-API: PKCS#11;
- позволяет выполнять до 7000 1024-битных операций RSA в секунду [7].

Для развертывания OpenDNSSEC и интеграции HSM Luna PCI необходимо настроить три конфигурационных файла в OpenDNSSEC и один в данном HSM.

Общая конфигурация OpenDNSSEC определяется содержанием файла `conf.xml`. В этом файле конфигурации необходимо указать возможность журналирования, систему путей, расположение ключевых хранилищ, привилегии, и базу данных, в которой хранятся все ключевые зоны. Также в данном файле необходимо настроить параметры `Enforcer`, который отвечает за аварийный сброс и генерацию ключей. В качестве базы данных рекомендуется использовать SQLite, т.к. работа с MySQL еще находится в тестовом режиме. Необходимо указать путь к базе данных, пользователя и пароль для данного пользователя. Рекомендовано ввести параметры проверки срока действия ключа и определиться с одним из двух режимов генерации ключей: ручным и автоматическим. Необязательным в данном случае является параметр, указывающий максимальное количество ключей, которое HSM может хранить [2, 3]. Ниже представлен небольшой отрывок из созданного файла конфигурации:

```
<Repository name="Guaprepozit"> <Module>/usr/lib/libCryptoki2.so</Module>
<TokenLabel>Guaprepozit</TokenLabel><PIN>guap757#</PIN>
</Repository>
```

Следующим был настроен файл конфигурации – zonelist.xml. В данном файле прописываются списки зон, и указываются политики, применяемые к данным зонам. Для небольшого числа зон, список файлов зон может быть легко отредактирован вручную. Если число зон велико – например, провайдер, обслуживающий тысячи клиентов, то предполагается, что файл создается с использованием менеджера зон ods-ksmutil.

Файл kasp.xml определяет политику, используемую для подписи зон. Каждая политика состоит из ряда параметров, которые определяют каким методом будет подписана зона. Иерархическая природа XML позволяет группировать параметры на логические блоки. В данном файле политики был выбран и указан номер используемого алгоритма хэш-шифрования. Также указываются основные параметры создания ключей KSK и ZSK, алгоритм шифрования, срок действия ключа, путь расположения хранилища HSM, в котором хранятся ключи, параметры аудита, время действия цифровой подписи до ее следующего обновления [2, 3].

В настройках HSM необходимо указать пути к конфигурационным файлам OpenDNSSEC, указать криптографический интерфейс взаимодействия (PKCS#11) и отключить некоторые функции HSM Luna PCI, которые не будут использованы, например, генератор случайных чисел.

В результате проделанной работы была развернута система OpenDNSSEC, создан вычислительный кластер на основе MPICH2 для оптимизации работы системы с SoftHSM, и интегрирован криптографический модуль Luna PCI.

Библиографический список

1. Журнал «Системный администратор 086-087» 2010 г. стр 72;
2. <http://www.opendnssec.org/>;
3. <http://book.itep.ru/4/4/dnssec.htm>;
4. <http://cluster.linux-ekb.info/ubuntu.php>;
5. <http://rfc2.ru/4033.rfc>;
6. <http://www.rfc-archive.org/getrfc.php?rfc=4034>.
7. http://www.datasec.ru/products/apparatnye_moduli_bezopasnosti/safenet_luna_pci/

УДК 621.372

В. А. Ненашев – магистрант кафедры вычислительных систем и сетей
Л. А. Мироновский (д-р техн. наук, проф.) – научный руководитель

ВЗАИМОСВЯЗЬ СБАЛАНСИРОВАННОГО ПРЕДСТАВЛЕНИЯ С КАНОНИЧЕСКИМИ ФОРМАМИ ЛИНЕЙНЫХ ДИНАМИЧЕСКИХ СИСТЕМ

При компьютерном моделировании динамических систем возникает потребность в выборе конкретной реализации системы в пространстве состояний. Чаще всего для этих целей выбирают одну из известных модальных, сопровождающих или цепных канонических форм (КФ) [1, 2]. Каждая из них обладает своими достоинствами и недостатками, поэтому необходимо иметь алгоритмическое и программное обеспечение для преобразования системных матриц исходной модели к разным каноническим формам, а также для перехода от одной канонической формы к другой.

В таком известном математическом пакете как MATLAB поддерживается лишь несколько канонических форм, в том числе сбалансированное представление, которое строится командой balreal. Это представление было открыто сравнительно недавно (в 1981 г.) и играет важную роль в современной теории управления и моделирования.

Оно обладает хорошей вычислительной устойчивостью, малочувствительно к погрешностям округлений и является эффективным средством для получения редуцированных моделей динамических систем.

В данной работе рассматривается задача установления взаимосвязи сбалансированного представления с другими каноническими представлениями, включая формы Жордана, Фробениуса и Рауса. Каждая из канонических форм связана со своим набором инвариантов (коэффициенты характеристического полинома, полюсы передаточной функции, ганкелевы сингулярные числа (ГСЧ), коэффициенты Рауса и др.) и поставленная задача требует нахождения аналитических соотношений, связывающие эти инварианты. В статье приводятся указанные соотношения, полученные для систем третьего порядка с одним входом и одним выходом.

1. Сбалансированная каноническая форма (СКФ).

Критерием СКФ является диагональность кросс-грамиана W_{cob} . При $n = 3$ матрицы ее представления в пространстве состояний и кросс-грамиан имеют вид:

$$A_b = \begin{bmatrix} -\frac{c_1}{2v_1} & -\frac{c_2}{v_1+v_2} & -\frac{c_3}{v_1+v_3} \\ -\frac{c_1}{v_1+v_2} & -\frac{c_2}{2v_2} & -\frac{c_3}{v_2+v_3} \\ -\frac{c_1}{v_1+v_3} & -\frac{c_2}{v_2+v_3} & -\frac{c_3}{2v_3} \end{bmatrix}, \quad b_b = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}, \quad c_b = [c_1 \quad c_2 \quad c_3], \quad W_{cob} = \begin{bmatrix} v_1 & 0 & 0 \\ 0 & v_2 & 0 \\ 0 & 0 & v_3 \end{bmatrix}.$$

Элементы главной диагонали кросс-грамиана называются ганкелевыми собственными числами системы (v_j). Параметрами сбалансированного представления служат два набора инвариантов: v_j, c_j .

2. Жорданова каноническая форма (ЖКФ).

Каноническая форма Жордана играет важную роль в задачах анализа и синтеза линейных динамических систем. ЖКФ согласована с двумя наборами инвариантов: полюсами и вычетами ПФ λ_j и r_j . В этой канонической форме уравнения системы выглядят особенно простыми. При $n = 3$ матрицы ЖКФ, кросс-грамиана и передаточная функция (ПФ) имеют вид:

$$A_j = \begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{bmatrix}, \quad b_j = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}, \quad c_j = [r_1 \quad r_2 \quad r_3], \quad W_{coj} = \begin{bmatrix} \frac{r_1}{2\lambda_1} & \frac{r_2}{\lambda_1+\lambda_2} & \frac{r_3}{\lambda_1+\lambda_3} \\ -\frac{r_1}{\lambda_1+\lambda_2} & \frac{r_2}{2\lambda_2} & \frac{r_3}{\lambda_2+\lambda_3} \\ \frac{r_1}{\lambda_1+\lambda_3} & \frac{r_2}{\lambda_2+\lambda_3} & \frac{r_3}{2\lambda_3} \end{bmatrix},$$

$$Q_j(p) = \frac{r_1}{p-\lambda_1} + \frac{r_2}{p-\lambda_2} + \frac{r_3}{p-\lambda_3}.$$

В случае комплексных и кратных собственных значений структура ЖКФ усложняется.

Чтобы установить взаимосвязь параметров СКФ и ЖКФ, приравняем коэффициенты характеристического полинома матриц A_b и A_j , а также кросс-грамианов W_{cob} и W_{coj} . В результате получаем систему соотношений:

$$\lambda_1 + \lambda_2 + \lambda_3 = -\frac{1}{2} \left(\frac{c_1}{v_1} + \frac{c_2}{v_2} + \frac{c_3}{v_3} \right),$$

$$\lambda_1 \lambda_2 + \lambda_1 \lambda_3 + \lambda_2 \lambda_3 = \frac{1}{2^2} \cdot \frac{c_1 c_2}{v_1 v_2} \cdot \left(\frac{v_1 - v_2}{v_1 + v_2} \right)^2 + \frac{c_1 c_3}{v_1 v_3} \cdot \left(\frac{v_1 - v_3}{v_1 + v_3} \right)^2 + \frac{c_2 c_3}{v_2 v_3} \cdot \left(\frac{v_2 - v_3}{v_2 + v_3} \right)^2,$$

$$\lambda_1 \lambda_2 \lambda_3 = -\frac{1}{2^3} \cdot \frac{c_1 c_2 c_3}{v_1 v_2 v_3} \cdot \left(\frac{v_1 - v_2}{v_1 + v_2} \right)^2 \cdot \left(\frac{v_1 - v_3}{v_1 + v_3} \right)^2 \cdot \left(\frac{v_2 - v_3}{v_2 + v_3} \right)^2,$$

$$v_1 + v_2 + v_3 = -\frac{1}{2} \left(\frac{r_1}{\lambda_1} + \frac{r_2}{\lambda_2} + \frac{r_3}{\lambda_3} \right),$$

$$v_1 v_2 + v_1 v_3 + v_2 v_3 = \frac{1}{2^2} \cdot \frac{r_1 r_2}{\lambda_1 \lambda_2} \cdot \left(\frac{\lambda_1 - \lambda_2}{\lambda_1 + \lambda_2} \right)^2 + \frac{r_1 r_3}{\lambda_1 \lambda_3} \cdot \left(\frac{\lambda_1 - \lambda_3}{\lambda_1 + \lambda_3} \right)^2 + \frac{r_2 r_3}{\lambda_2 \lambda_3} \cdot \left(\frac{\lambda_2 - \lambda_3}{\lambda_2 + \lambda_3} \right)^2,$$

$$v_1 v_2 v_3 = -\frac{1}{2^3} \cdot \frac{r_1 r_2 r_3}{\lambda_1 \lambda_2 \lambda_3} \cdot \left(\frac{\lambda_1 - \lambda_2}{\lambda_1 + \lambda_2} \right)^2 \cdot \left(\frac{\lambda_1 - \lambda_3}{\lambda_1 + \lambda_3} \right)^2 \cdot \left(\frac{\lambda_2 - \lambda_3}{\lambda_2 + \lambda_3} \right)^2.$$

В эти 6 уравнений входят 12 параметров. Задавая любые шесть из них, можно найти шесть остальных.

3. Фробениусова каноническая форма.

Фробениусова каноническая форма (ФКФ) широко используется в технических приложениях. Коэффициенты этой формы совпадают, с точностью до знака, с коэффициентами числителя и знаменателя ПФ. Это обстоятельство облегчает переход от описания системы с помощью дифференциальных уравнений или ПФ к уравнениям в пространстве состояний. ФКФ согласована с двумя наборами инвариантов, которыми служат коэффициенты числителя и знаменателя ПФ ($b_0 \ b_1 \ b_2$) и ($a_0 \ a_1 \ a_2$). При $n = 3$ матрицы ФКФ и ПФ имеют вид:

$$A_f = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -a_0 & -a_1 & -a_2 \end{bmatrix}, \quad b_f = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \quad c_f = [b_0 \ b_1 \ b_2], \quad Q_f(p) = \frac{b_2 p^2 + b_1 p + b_0}{p^3 + a_2 p^2 + a_1 p + a_0}.$$

Чтобы установить взаимосвязь параметров СКФ и ФКФ, удобно использовать формулу для получения ПФ $Q(p) = c(pE - A)^{-1} b$ и, приравнявая коэффициенты числителя и знаменателя ПФ соответственно, в результате получаем систему уравнений, связывающих параметры СКФ и ФКФ:

$$b_2 = c_1 + c_2 + c_3,$$

$$b_1 = \frac{1}{2} \cdot \frac{c_1 c_2}{v_1 v_2} \cdot \left(\frac{v_1 - v_2}{v_1 + v_2} \right) + \frac{c_1 c_3}{v_1 v_3} \cdot \left(\frac{v_1 - v_3}{v_1 + v_3} \right) + \frac{c_2 c_3}{v_2 v_3} \cdot \left(\frac{v_2 - v_3}{v_2 + v_3} \right),$$

$$b_0 = \frac{1}{2^2} \cdot \frac{c_1 c_2 c_3}{v_1 v_2 v_3} \cdot \left(\frac{v_1 - v_2}{v_1 + v_2} \right)^2 \cdot \left(\frac{v_1 - v_3}{v_1 + v_3} \right)^2 \cdot \left(\frac{v_2 - v_3}{v_2 + v_3} \right)^2 \cdot (v_1 + v_2 + v_3),$$

$$a_2 = -\frac{1}{2} \left(\frac{c_1}{v_1} + \frac{c_2}{v_2} + \frac{c_3}{v_3} \right),$$

$$a_1 = \frac{1}{2^2} \cdot \left(\frac{c_1 c_2}{v_1 v_2} \cdot \left(\frac{v_1 - v_2}{v_1 + v_2} \right)^2 + \frac{c_1 c_3}{v_1 v_3} \cdot \left(\frac{v_1 - v_3}{v_1 + v_3} \right)^2 + \frac{c_2 c_3}{v_2 v_3} \cdot \left(\frac{v_2 - v_3}{v_2 + v_3} \right)^2 \right),$$

$$a_0 = -\frac{1}{2^3} \cdot \frac{c_1 c_2 c_3}{v_1 v_2 v_3} \cdot \left(\frac{v_1 - v_2}{v_1 + v_2} \right)^2 \cdot \left(\frac{v_1 - v_3}{v_1 + v_3} \right)^2 \cdot \left(\frac{v_2 - v_3}{v_2 + v_3} \right)^2.$$

Как и ранее, задавая часть из них, можно находить остальные.

4. Каноническая форма Рауса (КФР).

КФР основана на разложении ПФ в цепную дробь. При $n = 3$ КФР согласована с двумя наборами инвариантов, которыми служат коэффициенты Рауса $(\alpha_0 \alpha_1 \alpha_2)$ и $(\beta_1 \beta_2 \beta_3)$. Матрицы КФР и ее ПФ имеют вид:

$$A_r = \begin{bmatrix} -a_0 & \sqrt{\alpha_1} & 0 \\ -\sqrt{\alpha_1} & 0 & \sqrt{\alpha_2} \\ 0 & -\sqrt{\alpha_2} & 0 \end{bmatrix},$$

$$b_r = \begin{bmatrix} \sqrt{\alpha_1} \\ 0 \\ 0 \end{bmatrix}, \quad c_r = [\beta_1 \quad \beta_2 \quad \beta_3],$$

$$Q_r(p) = \frac{\sqrt{\alpha_1} \beta_0 p^2 - \alpha_1 \beta_1 p + (\alpha_1 \sqrt{\alpha_2} \beta_2 + \sqrt{\alpha_1} \alpha_2 \beta_0)}{p^3 + \alpha_0 p^2 + (\alpha_1 + \alpha_2) p + \alpha_0 \alpha_2}.$$

Чтобы установить взаимосвязь параметров СКФ и ФКФ, используем формулу $Q(p) = c(pE - A)^{-1}b$, в результате получаем систему уравнений, связывающих параметры СКФ и КФР:

$$\sqrt{\alpha_1} \beta_0 = c_1 + c_2 + c_3,$$

$$\alpha_1 \beta_1 = -\frac{1}{2} \cdot \left(\frac{c_1 c_2}{v_1 v_2} \cdot \left(\frac{v_1 - v_2}{v_1 + v_2} \right) + \frac{c_1 c_3}{v_1 v_3} \cdot \left(\frac{v_1 - v_3}{v_1 + v_3} \right) + \frac{c_2 c_3}{v_2 v_3} \cdot \left(\frac{v_2 - v_3}{v_2 + v_3} \right) \right),$$

$$\alpha_1 \sqrt{\alpha_2} \beta_2 + \sqrt{\alpha_1} \alpha_2 \beta_0 = \frac{1}{2^2} \cdot \frac{c_1 c_2 c_3}{v_1 v_2 v_3} \cdot \left(\frac{v_1 - v_2}{v_1 + v_2} \right)^2 \cdot \left(\frac{v_1 - v_3}{v_1 + v_3} \right)^2 \cdot \left(\frac{v_2 - v_3}{v_2 + v_3} \right)^2 \cdot (v_1 + v_2 + v_3),$$

$$\alpha_0 = -\frac{1}{2} \left(\frac{c_1}{v_1} + \frac{c_2}{v_2} + \frac{c_3}{v_3} \right),$$

$$\alpha_1 + \alpha_2 = \frac{1}{2^2} \cdot \left(\frac{c_1 c_2}{v_1 v_2} \cdot \left(\frac{v_1 - v_2}{v_1 + v_2} \right)^2 + \frac{c_1 c_3}{v_1 v_3} \cdot \left(\frac{v_1 - v_3}{v_1 + v_3} \right)^2 + \frac{c_2 c_3}{v_2 v_3} \cdot \left(\frac{v_2 - v_3}{v_2 + v_3} \right)^2 \right),$$

$$\alpha_0 \alpha_2 = -\frac{1}{2^3} \cdot \frac{c_1 c_2 c_3}{v_1 v_2 v_3} \cdot \left(\frac{v_1 - v_2}{v_1 + v_2} \right)^2 \cdot \left(\frac{v_1 - v_3}{v_1 + v_3} \right)^2 \cdot \left(\frac{v_2 - v_3}{v_2 + v_3} \right)^2.$$

Полученные системы независимых соотношений, связывающие параметры СКФ и других КФ, могут быть полезны при анализе, синтезе и моделировании линейных систем третьего порядка.

Библиографический список

1. Мироновский Л. А. Функциональное диагностирование динамических систем: Научное издание. М.–СПб.: МГУ-РИФ, 1998. 256 с.
2. Мироновский Л. А. Взаимосвязь параллельной и сбалансированной канонических форм // Электронное моделирование. 1989. № 6. С.8–10.

УДК 004.732

Е. Н. Романенко – магистр кафедры вычислительных систем и сетей
В. В. Балберин (канд. техн. наук, доц.) – научный руководитель

ГРУППОВЫЕ ПОЛИТИКИ: ОБЗОР РЕАЛИЗАЦИЙ

Рассмотрены способы и инструменты реализации групповых политик, а также преимущества их применения.

С увеличением парка рабочих станций в корпоративной сети приходится дополнительно нанимать обслуживающий персонал. Из-за этого существенно возрастают расходы на содержание IT-инфраструктуры. Кроме того, в большой сети очень непросто отслеживать нарушения корпоративных стандартов настройки компьютеров. Таким образом, задача автоматизации администрирования рабочих мест пользователей возникает в любой крупной компании с развитой сетью.

Итак, некое гипотетическое предприятие использует GNU/Linux и MS Windows в качестве серверных ОС. На рабочих станциях также установлены Windows и Linux. Соответственно, администрировать придется компьютеры, работающие под управлением принципиально различных операционных систем. Поскольку основной объем работ системного администратора приходится на обслуживание Windows-компьютеров, то обзор начнем с них.

Групповые политики(Group Policy) Active Directory.

Прежде всего, стоит ввести определение групповой политики, чтобы иметь точное представление, о чем идет речь.

Групповая политика – это набор правил или настроек, в соответствии с которыми производится настройка рабочей среды Windows. Групповые политики создаются в домене и реплицируются в рамках домена. [1]

Компания Microsoft предлагает использовать групповые политики для централизованного управления компьютерами в домене Active Directory. Эта технология специфична для Windows (требуется Windows 2000 или более поздние версии) и ранее не могла быть использована для рабочих станций под управлением GNU/Linux в силу архитектурных различий операционных систем. С помощью групповых политик можно разрешать пользователям запуск только определенных приложений, настраивать минимальную длину пароля и множество других параметров. Все настройки находятся в объектах групповой политики (Group Policy Object, GPO) Active Directory, которые хранятся на контроллере домена и могут быть связаны с сайтом, доменом или OU (Organizational Unit, подразделение или организационная единица). Привязка GPO определяет область его действия. Любой объект содержит два раздела – Computer Configuration и User Configuration. Настройки первого применяются во время загрузки Windows. Второй раздел содержит настройки учетной записи, которые применяются во время входа пользователя в систему. Далее они обновляются каждые полтора часа, с вариацией в 30 минут. Политика действует только на объекты, находящиеся на уровне иерархии каталога (сайт, домен, подразделение), с которым связан GPO и ниже по дереву (если не запрещено наследование). Порядок применения политик таков: локальные политики,

политики уровня сайта, политики уровня домена, политики уровня OU. На всех уровнях GPO содержат одинаковые настройки, и действующим будет примененное последнее значение. Данное правило распространяется на все параметры кроме определенных как «Not Defined». Для последних Windows не предпринимает никаких действий. Единственное исключение – параметры настройки учетных записей и паролей могут быть определены только на уровне домена.[5]

Администрирование осуществляется с помощью утилит группы Administrative Tools (Администрирование) входящих в комплект поставки операционных систем Server, Advanced Server и Datacenter Server. Файл, необходимый для установки этих утилит, называется adminpak.msi и располагается в каталоге i386 на установочном компакт-диске или в подкаталоге WINNT\SYSTEM32\операционной системы. Этот 15-мегабайтный файл устанавливает все необходимые оснастки и регистрирует библиотеки DLL, необходимые для использования средств администрирования. В настоящее время установить утилиты группы Administrative Tools (Администрирование) можно только на платформе Windows 2000.

Работая с групповыми политиками, следует учитывать, что:

- объекты GPO применяются в отношении контейнеров, а не замыкающих объектов;
- один контейнер может быть связан с несколькими объектами GPO;
- объекты GPO, связанные с одним и тем же контейнером, применяются в отношении этого контейнера в том порядке, в котором они были назначены;
- объект GPO включает в себя две составляющие: параметры, относящиеся к компьютеру, и параметры, относящиеся к пользователю;
- обработку любой из этих составляющих можно отключить;
- наследование объектов GPO можно блокировать;
- наследование объектов GPO можно форсировать;
- применение объектов GPO можно фильтровать при помощи списков ACL.

Фильтрация применяемых политик на основе членства объектов в группах безопасности является еще одним методом изменения обычного порядка применения политик в отношении объектов Active Directory. Фильтрация осуществляется при помощи списков ACL (Access Control List). Каждому объекту GPO ставится в соответствие список ACL. Информация из списка ACL объекта GPO анализируется системой безопасности вне зависимости от того, к какому из контейнеров применяется данный GPO. Политика применяется в отношении объекта только в том случае, если объект обладает в отношении соответствующего GPO разрешениями Read (Чтение) и Apply Group Policy (применение групповой политики). Если объект (пользователь или группа) не обладает разрешением Apply Group Policy (применение групповой политики), политика группы в его отношении не применяется. [1]

Рассмотрим реализации групповых политик на GNU/Linux. Данную возможность можно реализовать с помощью Samba4.

Samba 4 развивается параллельно с веткой 3.x и нацелена на реализацию возможности работы в качестве Active Directory контроллера доменов.

Samba 4 разрабатывается с нуля, едва ли есть что-либо общее в основном коде Samba используемое в Samba3 и Samba4. Почему это было сделано?

Samba 3 демонстрирует свой возраст. По существу, это все еще самая первая реализация, которой положил начало Andrew Tridgell более чем десять лет тому назад. Безусловно, она пережила большое количество изменений со времени первой реализации, но внутренняя структура на самом деле не очень изменилась в процессе разработки.

Samba 4 была учреждена с целью устранения этой ошибки, для того чтобы вывести инфраструктуру на должный уровень, без надобности заботиться о совместимости с существующей системой кодирования.

Две большие части, которых не было до Samba 4 в Контроллере Домена, который Windows-машина принимает как полноценный контроллер домена Active Directory, – это поддержка LDAP и Kerberos.[3]

Файловый сервер Samba хранит служебную информацию, в частности, информацию об учётных записях пользователей, в специальной базе данных, passdb backend. Сейчас Samba поддерживает большое количество разнообразных видов хранилищ, начиная от простого текстового файла (smbpasswd) и заканчивая MySQL-сервером.

Одним из распространённых в последнее время типов хранилищ, используемых вместе с Samba, является LDAP-хранилище, то есть фактически, сервер каталога, доступный по протоколу LDAP.

Наиболее важными преимуществами этого хранилища в сравнении с остальными является:

- простота интеграции с другими системами;
- возможность масштабирования.

Простота интеграции с другими системами позволяет добиться того чтобы все (или большинство) систем, используемых в сети, имели доступ и использовали в своей работе одну и ту же учётную информацию. Это позволяет упростить процесс администрирования и избежать создания внутренних противоречий. [2]

В новой версии Samba администратором является не пользователь root, а именно пользователь Administrator. Это сделано для полной совместимости UNIX/Windows.

Среди прочих функций, которые уже реализованы в Samba 4, стоит отметить:

- поддержка входа в систему (logon) в стиле 'Active Directory';
- полная поддержка семантики NTFS;
- встроенный сервер LDAP, поддерживающий семантику AD;
- встроенный сервер Kerberos, включающий поддержку PAC;
- новая RPC-инфраструктура (PIDL);
- гибкая архитектура базы данных (LDB). [4]

Представленные в этой статье компоненты являются частями основы инфраструктуры Samba 4. Сведенные воедино они составляют Контроллер Домена.

Библиографический список

1. http://ru.wikipedia.org/wiki/Групповая_политика
2. http://xgu.ru/wiki/LDAP-репликация_и_Samba
3. <http://www.samba.org.ua/articles/?section=2&articleid=40>
4. http://xgu.ru/wiki/Samba_4
5. <http://its-srv.ru/articles/100-linux-ad-part3>

УДК 004.021

И. А. Сачков – студент факультета информационных систем и защиты информации
В. А. Килимник (канд. техн. наук, ст. науч. сотр.) – научный руководитель

СПОСОБЫ ОПТИМИЗАЦИИ ВЗАИМОДЕЙСТВИЯ ПРОГРАММНЫХ МОДУЛЕЙ КОМПЛЕКСА ОБРАБОТКИ ИНФОРМАЦИИ

Основной задачей настоящего исследования являлось рассмотрение способов оптимизации элементов программного обеспечения аппаратно программного комплекса [1]. Процесс оптимизации структуры приложения, являющийся неотъемлемой частью его разработки, в рассматриваемом нами случае распадается на несколько основных направлений, каждое из которых отражает одну из основных задач, которые выполняет аппаратно программный комплекс в целом, а именно: оптимизацию сбора данных; оптимизацию вывода данных; оптимизацию внутреннего взаимодействия.

Оптимизация сбора данных, в первую очередь, заключается в минимизации структур для хранения получаемой в процессе работы информации, уменьшения количества дополнительных данных и производимых проверок, без вреда для целостности самих получаемых данных.

Первым решением, примененным в процессе разработки структуры комплекса хранения данных, явился отказ от языка программирования C# с последующим переходом к языку C++. В первую очередь это позволило бы избавиться от излишних затрат, связанных с механизмом сбора мусора, являющегося неотъемлемой частью среды .NetFramework. Вторым важным моментом подобного решения является возможность использования дополнительных низкоуровневых оптимизаций, основанных на работе с отдельными байтами элементов данных. Третьим достоинством подобного подхода является возможность подключения дополнительных библиотек без лишних затрат ресурсов, что так же в целом позволяет разгрузить процесс получения данных от устройства.

Несмотря на необходимость максимальной оптимизации по времени работы, создаваемая структура должна позволять обращаться к хранимым данным в требуемый момент времени без дополнительных затрат для других систем, в первую очередь по времени, которые обращаются за данными, полученными от устройства. В рассматриваемой реализации структуры хранения данных для каждой из «отраслей», связанных со специфической информацией, реализована двухступенчатая структура для хранения данных:

1. Массив данных (сет, блок) – реализует основные инструменты для выделения памяти, добавления, удаления, изменения данных, имеющие в своей структуре элементы, необходимые для преобразования и форматирования входных потоков информации, получаемой от устройства, либо из других источников, в том числе от других средств хранения.

2. Обобщенная оболочка хранения данных – структура, оперирующая массивами данных, реализующая механизмы многопоточного доступа, сериализации и первичного анализа (получение различных характеристик хранимых данных, оценка скорости заполнения памяти компьютера для принятия решения о буферизации потока данных на другом носителе).

Подобная структура позволяет производить дальнейшие оптимизации, связанные с конкретной предметной областью (сжатие данных, многопоточное взаимодействие, реализация обмена данными между оперативной памятью и жестким диском). При этом функциональные элементы программы затрагивают элементы только того уровня, который им нужен для непосредственного решения, поставленных перед ними задач, т. е. элемент вывода, производящий отображение данных, соответствующих отдельному информационному каналу не имеет представления о том, к какому «семейству» данный канал относится.

Оптимизация процесса графического вывода данных в реальном масштабе времени с большой частотой является актуальной проблемой, при разработке приложения, предназначенного для работы с внешним устройством. При этом, разработка графических элементов приложения, не связанных непосредственно с выводом информации, связанной непосредственно с исследованием, вовлекает в структуру приложения дополнительные графические компоненты (WPF, QT, MFC и т. д.), что соответственно создает необходимость создания дополнительных программных конструкций для адаптации разрабатываемого высокопроизводительного узко специализированного решения к выбранной среде графического интерфейса приложения.

Ввиду использования набора компонент WindowsPresentationFoundation, а так же операционной системы Windows 7, как целевой среды исполнения, основным графическим программным интерфейсом для реализации быстрого вывода графики был выбран Direct2D. Данный Api позволил, избавившись от введения в структуру приложения дополнительных абстракций рисования двумерной графики, создать набор компонент

вывода графических объектов с использованием аппаратного ускорения и средств обработки изображения.

Разработанное на нем решение удовлетворяло всем поставленным критериям графического отображения (более 40 кадров в секунду на целевой аппаратной конфигурации при максимально возможной нагрузке), но имело один существенный недостаток – необходимость использования средств пространства имен `InteropServices` среды `NetFramework`, что влечет за собой дополнительные расходы на каждый производящийся вызов функции `Direct2D`.

Для решения данной проблемы был использован подход, предложенный Иеремией Морриллом (`JeremiahMorrill`) [2]. Основная его суть заключается в применении механизма разделяемых ресурсов (`sharedresources`), реализуемого между программными абстракциями `Direct3D 9Ex` и `Direct3D 10.1` с последующей передачей отрисованного кадра компоненту, унаследованному от стандартного `D3DImage` из поставки `NetFramework`, выполняющего работу, связанную с выводением конечного изображения на экран. Данный момент особенно важен, ввиду того, что в данный момент времени компонент `D3DImage` может производить отрисовку кадров, полученных от программных компонент `DirectX 9 (Ex)`, но не от компонент `DirectX 10.1`. При этом поддержка `Direct2D` присутствует только в последнем.

Использование подобного рода конструкции позволяет минимизировать число дополнительных «дорогих» вызовов средств `InteropServices`, что в целом положительно сказывается на производительности приложения и на скорости вывода изображения в частности.

Основной проблемой внутреннего взаимодействия является задача минимизации дополнительных операций для получения требуемой промежуточной информации, в том числе получаемой от пользователя (многократный ввод настроечных данных, конфигурация различных элементов приложения, ввод данных исследования), решаемая в частности по средствам структуризации хранимой информации и создания внутренней системы обмена данными исследования между программными компонентами.

Для решения данной проблемы в структуру приложения вводится ряд дополнительных структур – профилей (профили `usb-устройства`, исследуемого объекта, состояния приложения и т. д.) [3], состоящих, по сути, из строчного имени и группы полей, непосредственно хранящих информацию. Ключевым элементом приложения при подобном подходе является объект, взаимодействующий непосредственно с подобными структурами – фабрика – оператор. Данный программный компонент позволяет генерировать необходимые профили по информации, получаемой от пользователя, либо `usb-устройства` и впоследствии использовать их для генерации следующих «поколений», что позволяет минимизировать количество вводимой информации, а так же систематизировать хранящиеся данные, полученные в процессе исследований. При этом создаваемые структуры имеют между собой определенные зависимости: так к примеру конфигурационный профиль устройства является источником построения профиля оболочки хранения данных, а тот в свою очередь – источником для профиля графического вывода потока информации. При этом непосредственное вмешательство пользователя требуется только на этапе первого подключения устройства к компьютеру, а все дальнейшие действия производятся фабрикой – оператором. В частности на втором этапе производится обращение к базе исследований, доступных для данного устройства, а на третьем – сопоставление выбранного исследования и заранее сгенерированной разработчиком карты разметки каналов для данного исследования.

Рассмотренные направления оптимизации внутреннего взаимодействия не только охватывают основные проблемные моменты разработки аппаратно программного комплекса, направленного на сбор и обработку биологической информации, но и определяют ряд направлений для дальнейшего развития. Одним из таких направлений является дальнейшее развитие использования `Direct2DApi`, что потребует создания новых подходов по мере его дальнейшего внедрения в среду `NetFramework`. Задача оптими-

зации хранения данных в свою очередь сильно изменяется в виду возникновения новых требований к приложению, таких как возможность сетевой синхронизации различных устройств аппаратной части комплекса, возможность создания зашифрованных массивов данных для хранения и т. д.

Библиографический список

1. ГОСТ 27878-88. Системы и комплексы медицинские автоматизированные. Термины и определения. ВНИИМП. Издательство стандартов, Москва, 01.07.1989 (19.04.2010), 13 с.
2. <http://jmorill.hjtcentral.com/Home/tabid/428/EntryId/437/Direct3D-10-11-Direct2D-in-WPF.aspx>
3. Макконелл С. Совершенный код, ISBN 5-469-00822-3, 5-7502-0064-7.

УДК 004.021

И. А. Сачков – студент факультета информационных систем и защиты информации
В. А. Килимник (канд. техн. наук, ст. науч. сотр.) – научный руководитель

СТРУКТУРНАЯ ОПТИМИЗАЦИЯ ПРОГРАММНОГО КОМПЛЕКСА ОБРАБОТКИ ИНФОРМАЦИИ ПОСРЕДСТВОМ ПРИМЕНЕНИЯ «СОВМЕЩЕННОЙ» СРЕДЫ ИСПОЛНЕНИЯ

Основной задачей проделанной работы являлось теоретическое определение оптимальной структуры построения аппаратно программно комплекса сбора данных от носительно имеющихся технических компонент.

Целевой операционной системой для разработки программного обеспечения являлась Windows 7, а аппаратной платформой – `rsx86\х64`. Одним из критериев оптимальности системы является ее способность при выбранном способе исполнения выполнять поставленные задачи за минимальное количество времени. При этом так же система не должна быть структурно избыточной, т.к. одним из требуемых критериев является необходимость постепенного расширения по мере развития комплекса, а так же возможность создания узкоспециализированных реализаций приложения для решения специфических задач, возникающих в процессе проведения того или иного исследования.

Для решения поставленной задачи построения приложения были рассмотрены языки C++, C#, а так же расширение языка C++ – C++/CLI.

Перед комплексом стоят три основные задачи: получение и обработка данных, их отображение, хранение данных и управление системой в целом.

Для решения первой из поставленных задач наиболее оптимальным решением является использование языка C++, что в первую очередь обусловлено тем, что предоставляемые разработчиком технического решения программные компоненты (библиотека для работы с драйвером устройства), написаны на C. Помимо этого язык позволяет производить большое число низкоуровневых операций без дополнительных затрат, таких как привлечение средств C# пространства имен `InteropServices`.

Так же, ввиду использования языка C++, становится возможным использование программного интерфейса `Direct2D`, применяющегося для вывода двухмерной графики в Windows 7.

Решение проблемы хранения разнообразных сопровождающих основной поток информации, получаемой при исследовании, вспомогательных данных, заключается в применении языка C# в совокупности с набором программных компонент `.NetFramework 4.0`. В первую очередь необходимо хранить и обрабатывать большое количество настроечных данных, информации об используемом оборудовании, личные данных

исследуемых пациентов, для чего хорошо подходит совокупность формата разметки xml и средств работы с ним, представленных в C#.

Помимо этого ряд компонентов вывода, время работы которых не сказывается на скорости работы системы в целом, так же могут быть написаны на языке C# для сокращения общего времени разработки программы.

Для взаимодействия

двух выбранных языков программирования и сопровождающих их программных компонент в перечень средств разработки было добавлено расширение языка C++ – C++/CLI, позволяющее разрабатывать “управляемые” компоненты для среды .NetFramework с использованием “нативного” кода без применения сторонних средств, требующих подключения дополнительных сборок.

Структурное распределение используемых языков программирования изображено на рис. 1.

Ключевым решением при разработке программного комплекса является применение расширения языка C++ – CLI. Аббревиатура CLI расшифровывается как CommonLanguageInfrastructure (Обобщенная языковая инфраструктура) и вводит в обращение набор типов данных, относящихся к объединенной системе типов (UnifiedTypeSystem,UTS), являющихся базовыми для всех “управляемых” языков программирования, поддерживаемых в Microsoft.NetFramework. При этом к общему арсеналу языка C++ добавляется ряд операторов, позволяющих в частности производить создание объектов в управляемой памяти, использовать языковые конструкции, такие как foreach, а так же наследовать разрабатываемые классы от базовых классов .NetFramework с возможностью задания области видимости. [1, 2]

Применение подобного «совмещенного» подхода к программированию позволяет производить требуемые оптимизации за счет отсутствия необходимости «компенсации» недостатков той или иной среды. Так, при получении информации от драйвера устройства мы минимизируем количество дополнительно выделяемой информации и вспомогательных проверок, за счет применения языка C++, в то же время сокращаем время работы над проектом за счет применения технологий, реализованных на базе языка C#, таких как обработка информации, хранимой в формате xml, работа с удаленными базами данных, работа со сценариями выполнения, ведение учета производимых действий, создание отчетов, а так же их вывод на печать. Полученный в результате набор программных компонент объединяется в единое приложение за счет использования расширения языка C++/CLI.

Таким образом, при разработке программной части аппаратно программного комплекса мы достигаем наиболее оптимального по установленным нами критериям решения, за счет применения различных сочетаний языков программирования и программных средств, что обеспечивает уменьшение общего числа непредвиденных проблем, которые, как правило, оказываются не решенными при непосредственном проектировании программного продукта.

Библиографический список.

1. <http://www.ecma-international.org/publications/standards/Ecma-372.htm>
2. <http://msdn.microsoft.com/ru-ru/library/xey702bw.aspx>



Рис. 1. Распределение использования различных языков программирования при написании программного комплекса

УДК 621.372

И. Г. Ханьков – магистрант кафедры вычислительных систем и сетей

Л. А. Мироновский (д-р техн. наук, проф.) – научный руководитель

ПОСТРОЕНИЕ ГОДОГРАФОВ ДЕФЕКТОВ МОСТОВОЙ СХЕМЫ НА ДИАГРАММЕ ВЫШНЕГРАДСКОГО

Известны различные методы обнаружения и локализации параметрических дефектов динамических систем, например метод дублирования, метод избыточных переменных, тестовые методы [1]. В докладе рассматривается графический способ, использующий построение годографов однократных дефектов на плоскости Вышнеградского. В работах [1,2] рассматривалось диагностирование по годографам дефектов в плоскости диаграммы Найквиста и в плоскости сингулярных чисел. Плоскость Вышнеградского ранее для диагностирования не использовалась.

Диаграмма Вышнеградского представляет собой классический инструмент для исследования устойчивости линейных систем третьего порядка [3]. Характеристическое уравнение таких систем

$$a_3 p^3 + a_2 p^2 + a_1 p + a_0 = 0$$

заменой переменных $p = q \sqrt[3]{\frac{a_0}{a_3}}$ и сокращением на a_0 приводится к виду

$$q^3 + Aq^2 + Bq + 1 = 0, \quad (1)$$

где коэффициенты A, B называются параметрами Вышнеградского:

$$A = \frac{a_2}{a_3^{2/3} \sqrt[3]{a_0}} \quad \text{и} \quad B = \frac{a_1}{\sqrt[3]{a_3 a_0^{2/3}}}.$$

Возможна также альтернативная форма записи характеристического уравнения, получаемая заменой переменных $p = \frac{a_2}{3a_3} q$:

$$q^3 + 3q^2 + Mq + N = 0. \quad (2)$$

Коэффициенты M, N представляют собой модифицированные параметры Вышнеградского:

$$M = \frac{a_1 3^2}{a_2^2} \quad \text{и} \quad N = \frac{a_0 3^3}{a_2^3}.$$

Между коэффициентами A, B и M, N существует следующая взаимосвязь:

$$A = \frac{3}{\sqrt[3]{N}} \quad \text{и} \quad B = \frac{M}{\sqrt[3]{N^2}}, \quad M = \frac{9B}{A^2} \quad \text{и} \quad N = \frac{27}{A^3}.$$

На рис. 1, а и б изображены классическая и модифицированная диаграммы Вышнеградского. Линиям одной диаграммы соответствуют аналогичные линии

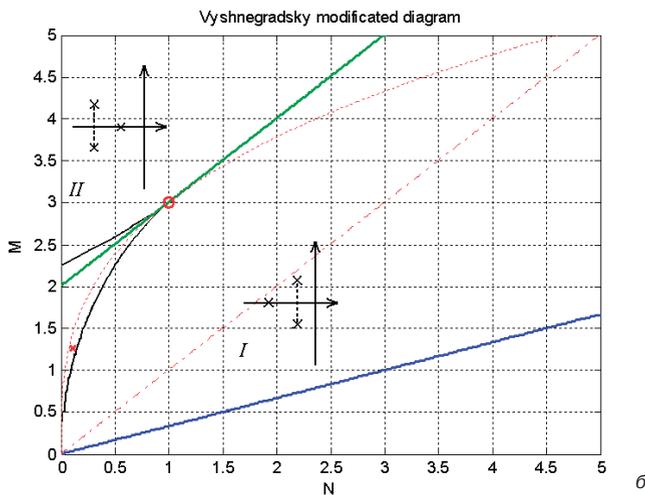
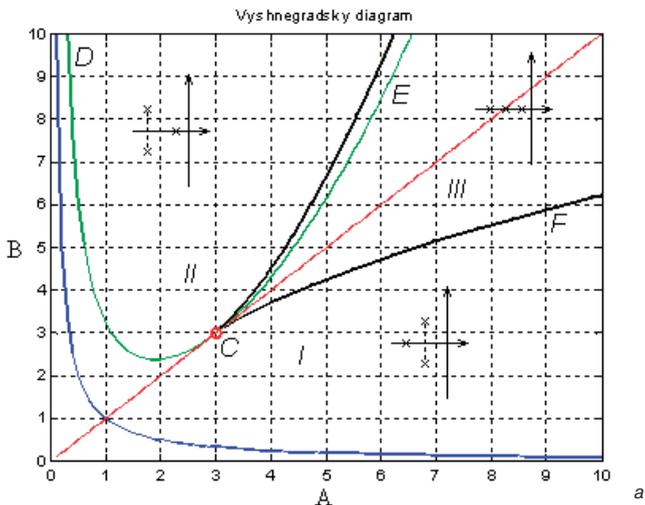


Рис. 1. Диаграммы Вышнеградского: а – классическая; б – модифицированная

на другой диаграмме. Уравнения границы устойчивости для каждой из диаграмм таковы:

$$B = \frac{1}{A}, \quad M = \frac{1}{3}N.$$

Области внутри «клювиков-уголков» отвечают системы с тремя вещественными корнями характеристического уравнения. Уравнения границы этих областей следующие:

$$A^2B^2 - 4(A^3 + B^3) + 18AB - 27 = 0, \quad 27N^2 - 54NM - 9M^2 + 108N + 4M^3 = 0.$$

Линия от точки D до точки C разделяет системы с парой комплексных корней на две группы: комплексная пара расположена левее или правее вещественного корня. Уравнения этой границы таковы:

$$B = \frac{-2A^3 - 27}{-9A}, \quad M = N + 2.$$

Биссектрисе классической диаграммы Вышнеградского соответствует кривая модифицированной диаграммы с уравнением $M = 3\sqrt[3]{N}$.

Далее обе плоскости Вышнеградского будут использованы для построения годографов дефектов.

Каждой паре значений параметров Вышнеградского A, B либо M, N соответствует точка на плоскости (A, B) либо (M, N) . Коэффициенты A, B, M, N зависят от параметров системы. Если изменять значение одного из параметров, то изображающая точка на плоскости будет двигаться по некоторой линии – годографу. Путем поочередного варьирования значений параметров модели строятся пучки годографов, число ветвей которых равно числу параметров системы.

Суть предлагаемого способа диагностирования сводится к следующему. На вход проверяемой системы подается тестовый сигнал и по значению выходного сигнала определяются параметры Вышнеградского. Исправному объекту на плоскости A, B или M, N будет соответствовать некоторая номинальная точка. При изменении одного из физических параметров системы изображающая точка начнет смещаться вдоль некоторой кривой – годографа дефекта. Изменению другого параметра будет соответствовать другой годограф.

Суть предлагаемого способа диагностирования сводится к следующему. На вход проверяемой системы подается тестовый сигнал и по значению выходного сигнала определяются параметры Вышнеградского. Исправному объекту на плоскости A, B или M, N будет соответствовать некоторая номинальная точка. При изменении одного из физических параметров системы изображающая точка начнет смещаться вдоль некоторой кривой – годографа дефекта. Изменению другого параметра будет соответствовать другой годограф.

Определение коэффициентов A, B или M, N может быть выполнено с помощью одного из методов идентификации, например метода Прони. При использовании этого метода на вход системы подается дельта-функция. С выхода снимается отклик (весовая функция). В случае если у системы все корни простые и вещественные, выходной сигнал можно представить в виде суммы экспонент:

$$y = C_1 e^{\lambda_1 t} + C_2 e^{\lambda_2 t} + \dots + C_n e^{\lambda_n t}, \quad (6)$$

где $\lambda_1, \lambda_2, \dots, \lambda_n$ – корни характеристического уравнения, C_1, C_2, \dots, C_n – неизвестные постоянные коэффициенты.

Особенность метода Прони заключается в том, что происходит переход от исследования непрерывной системы к некоторому ее разностному аналогу. Функция (6) удовлетворяет как дифференциальное (7), так и разностное (8) уравнения:

$$y^{(n)} + a_{n-1}y^{(n-1)} + \dots + a_0y = 0, \quad (7)$$

$$y_{n+k} + \beta_{n-1}y_{n+k-1} + \dots + \beta_0y_k = 0. \quad (8)$$

Корни характеристического дифференциального уравнения λ_j и характеристического разностного уравнения μ_j связаны следующими соотношениями:

$$\lambda_j = \frac{\ln \mu_j}{h}, \quad \mu_j = e^{\lambda_j h}, \quad (9)$$

где h – шаг дискретного времени.

Для идентификации методом Прони формируется массив равноотстоящих отсчетов выходного сигнала $y = f(t)$. На основе этого массива составляется система алгебраических уравнений. В матричном виде она выглядит так:

$$A\alpha = b, \quad (10)$$

где α – вектор неизвестных коэффициентов разностного уравнения, A, b – матрица и вектор-столбец, составленные из значений массива отсчетов выходного сигнала системы.

После того как коэффициенты характеристического разностного уравнения определены, находятся его корни, а по ним через формулу пересчета (9) – корни характеристического полинома дифференциального уравнения. Затем по ним восстанавливаются коэффициенты характеристического полинома реальной системы и ее параметры A, B или M, N . Точка с этими координатами наносится на плоскость Вышнеградского.

Если объект диагностирования исправен, то она окажется в месте пересечения всех годографов, т.е. в номинальной точке. Если имеет место однократный дефект, то точка попадет на ветвь годографа, отвечающую неисправному параметру.

В качестве примера объекта диагностирования используется пассивная электрическая мостовая RLC-схема третьего порядка с девятью элементами. Исследуемая схема приведена на рис. 2. Номинальные значения элементов таковы:

$$R_0 = R_2 = R_4 = R_5 = C_1 = C_2 = L_1 = 1, \\ R_1 = R_3 = 3.$$

При этих значениях передаточная функция схемы имеет вид:

$$W = \frac{7p^2 + 3p}{60p^3 + 94p^2 + 44p + 6}.$$

Для нее были построены пучки годографов однократных дефектов на плоскостях Вышнеградского. Они изображены на рис. 3, а и б для всех 9 элементов схемы $R_0, R_1, R_2, R_3, R_4, R_5, C_1, C_2, L_1$. Все годографы пересекаются в номинальных точках

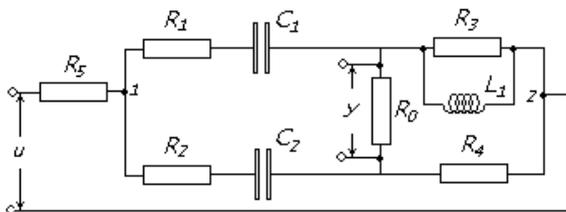


Рис. 2. Исследуемая мостовая схема

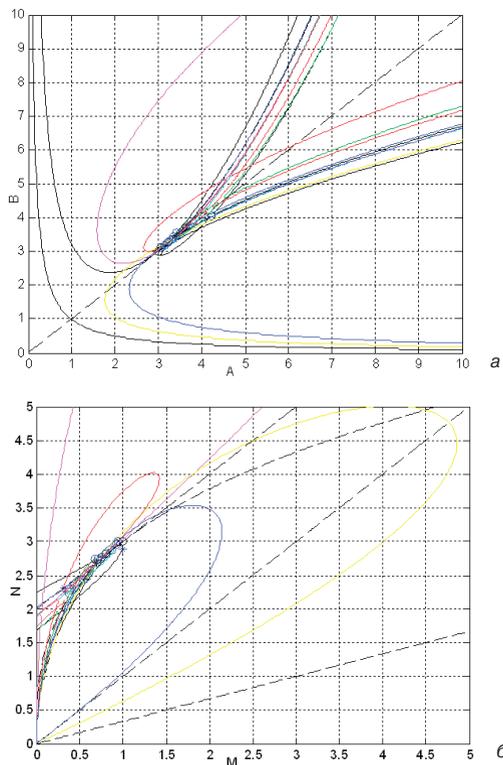


Рис. 3. Пучок годографов мостовой схемы: а – на классической диаграмме Вышнеградского; б – на модифицированной диаграмме Вышнеградского

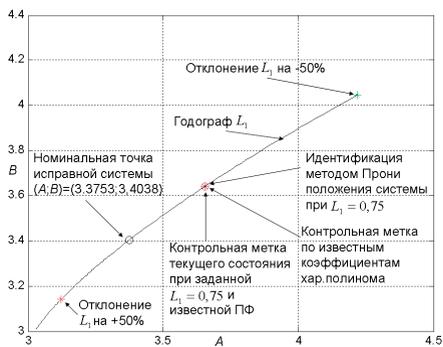


Рис. 4. Годограф L_1

$(A; B) = (3.3753; 3.4038)$, $(M; N) = (2,6895; 0,7023)$. Отклонение одного из параметров от номинала приведет к смещению изображающей точки вдоль соответствующего годографа, что и позволит диагностировать дефект.

Для оценки качества диагностирования был проведен компьютерный эксперимент, в процессе которого в модель схемы вносился однократный дефект. Он состоял в изменении величины индуктивности L_1 на 25% (0.75 вместо 1). В результате идентификации методом Прони был найден характеристический полином неисправной схемы и соответствующие значения коэффициентов Вышнеградского A^* , B^* . Изображающая точка с этими координатами попала на годограф, отвечающий индуктивности L_1 (он показан в укрупненном масштабе на рис. 4, звездочкой помечена изображающая точка). На годографе также помечены номинальная точка исправной системы (кружок) и точки отклонений индуктивности L_1 на $\pm 50\%$ (крестики).

Заключение. В работе показана возможность использования диаграмм Вышнеградского для диагностирования параметрических дефектов.

Для компьютерной реализации предлагаемого метода были написаны следующие программы и функции на языке MATLAB:

- функции построения диаграмм Вышнеградского и нанесения номинальной точки системы на плоскость;
- программа нанесения годографов однократных дефектов на плоскость Вышнеградского;
- программа идентификации коэффициентов характеристического уравнения по отклику системы;
- функция пересчета коэффициентов характеристического уравнения в параметры Вышнеградского A, B и M, N .

Результаты компьютерного моделирования подтвердили работоспособность разработанных программ.

Библиографический список

1. Мироновский Л. А. Функциональное диагностирование динамических систем: Научное издание. М.–СПб.: МГУ-РИФ, 1998. 256 с.
2. Соловьева Т. Н. Диагностика электрических цепей по диаграмме Найквиста // Современные технологии в задачах управления, автоматики и обработки информации: Труды XVII Международного научно-технического семинара. Алушта, сентябрь 2008 г. СПб.: ГУАП, 2008. С. 85–86.
3. Бесекинский В. А, Полов Е. П. Теория систем автоматического управления. Профессия, 2003. 752 с.

УДК 004.946, 004.031.42

А. В Шаблий – магистрант кафедры вычислительных систем и сетей
Н. Н. Решетникова (канд. техн. наук, доц.) – научный руководитель

РЕАЛИЗАЦИЯ ТРЕХМЕРНОЙ ИНТЕРАКТИВНОЙ МОДЕЛИ АВТОМОБИЛЯ С ОСНОВНЫМИ ФИЗИЧЕСКИМИ СВОЙСТВАМИ

Большинство современных водителей, первый раз выезжая на улицы города, не думают о соблюдении правил дорожного движения. Это связано с тем, что каждый водитель в первую очередь старается уберечь свой автомобиль и себя от попадания в дорожно-транспортное происшествие (ДТП). Очень часто инстинкт самосохранения как раз и приводит к ДТП. Современная подготовка водителя, как правило, занимает

2-3 месяца. Практическим занятиям в условиях города отводится 2–3 раза в неделю, примерно по 1–1,5 часа каждое. С уверенностью можно сказать, что этого очень мало. Тренировки в условиях города в присутствии инструктора за такое время не могут дать начинающему водителю тот необходимый уровень навыков для уверенного поведения за рулем. В наши дни на помощь желающим получить водительские права приходят современные мультимедийные технологии. Всевозможные электронные учебники правил дорожного движения (ПДД), симуляторы вождения автомобиля делают обучение чуть более разнообразным и доступным. Но, несмотря на изобилие этих средств, лишь не многие из них могут действительно повысить навык управления автомобилем.

В данной статье рассматривается часть проекта по созданию виртуальной интерактивной обучающей программы изучения ПДД в области управления автомобилем.

Безусловно, управление виртуальной моделью автомобиля не передает всех тех ощущений, которые испытывает водитель, сидя за рулем настоящего. Но, несмотря на это, существует современная платформа, позволяющая с помощью программного аппаратных средств, сделать этот процесс максимально реалистичным -3DVia Virtools. [1] Данная платформа была создана для конечной разработки интерактивных трехмерных сцен. И главной ее особенностью является наличие физического движка Havok. [2]

Рассмотрим подробнее основные этапы реализации трехмерной интерактивной модели автомобиля.

Исходная модель создавалась в трехмерном редакторе Autodesk 3D Studio Max (3Ds Max) [3], так как этот редактор наиболее удобный для работы с высокополигональными объектами. Для достижения максимального внешнего сходства использовались чертежи реально существующего автомобиля.

Следует упомянуть, что в редакторе 3Ds Max используется правосторонняя («right-handed»), а не левосторонняя («left-handed») система координат, как в DirectX [4], а также вверх направлена ось Z, а не Y. Такие особенности вызывают определенные трудности при экспорте готовой модели в среду Virtools. Для решения этой проблемы необходимо осуществлять экспорт через редактор Autodesk MAYA [3] используя расширение OBJ. При экспорте в этот формат программа по желанию разработчика автоматически преобразует систему координат 3Ds Max в систему координат DirectX, как показано на рис. 1.

Эти операции необходимо проделать для того чтобы ориентировать кузов и колеса готовой модели автомобиля так, как показано на рис. 2, что является одним из обязательных условий работы физического движка Virtools.

Как упоминалось ранее, исходная трехмерная модель автомобиля создавалась с использованием чертежей реально существующего автомобиля. Результат моделирования представлен на рис. 3.

Автомобиль, как в жизни, так и в виртуальной среде это объект, состоящий из множества деталей, находящихся в постоянном взаимодействии друг с другом и окружаю-



Рис. 1. Система координат слева 3Ds Max справа DirectX

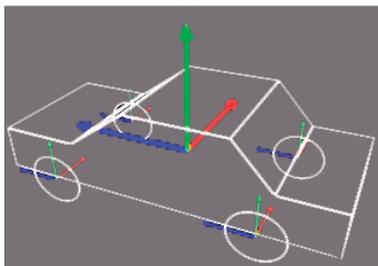


Рис. 2. Ориентация кузова и колес автомобиля в системе координат Virtools



Рис. 3. Полученная модель автомобиля

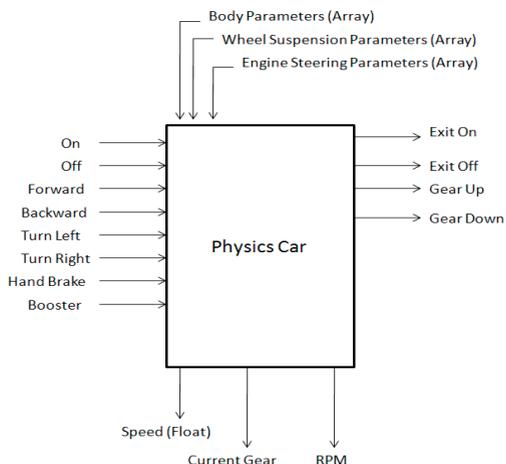


Рис. 4. Блок Physics Car

щей средой. Для моделирования основных взаимодействий автомобиля с окружающей средой в среде Virtools предусмотрен блок Physics Car, с помощью которого из системы объектов (кузова и четырех колес) можно создать физическую модель автомобиля. Блок предусматривает задание таких свойств, как работу подвески автомобиля, силу трения, ускорение, торможение и многие другие.

Рассмотрим подробнее блок Physics Car пакета Virtools (рис. 4).

Основными входными параметрами блока являются таблицы значений:

- Body Parameters – параметры кузова автомобиля (масса, сила трения, центр тяжести и др.)
- Wheel – Suspension Parameters – параметры колес и подвески (масса колес, сила трения, упругость подвески, клиренс и др.)
- Engine – Steering Parameters – параметры динамики и управляемости (мощность двигателя, минимальные и максимальные обороты, максимальная скорость и др.)

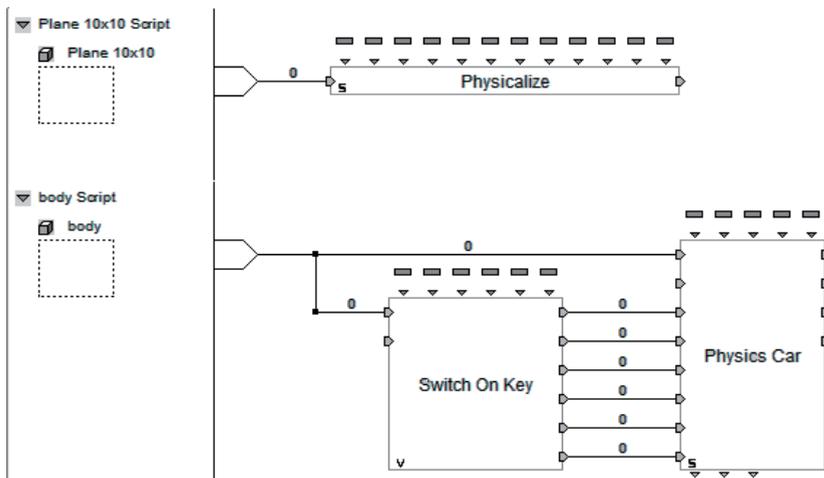


Рис. 5. Сценарий движения автомобиля с использованием блока Physics Car

Входы On, Off, Forward, Backward, Turn Left, Turn Right, Hand Brake, Booster используются для подачи сигналов с органов управления – Включение системы, Выключение системы, Импульс вперед, Импульс назад, Поворот налево, Поворот направо, Ручной тормоз (блокировка задних колес) и Ускорение соответственно.

Конечный сценарий движения автомобиля с использованием блока Physics Car приведен на рис. 5.

На рис. 5 наряду со сценарием движения автомобиля представлен сценарий Physicalize, который необходимо применять ко всем предметам виртуального пространства, для того чтобы автомобиль мог вступать с ними в физическое взаимодействие. Основными параметрами блока Physicalize являются масса тела, коэффициент трения, коэффициент упругости и центр тяжести объекта к которому он применяется.

Предполагалось, что физические параметры модели будут соответствовать реальным параметрам. На практике оказалось, что не все параметры, используемые в блоке Physics Car, однозначно соответствуют реальным. Многие из них весьма условны. Одним из ярких примеров является условие, при котором вес всех четырех колес должен быть равен весу кузова автомобиля. В связи с этим параметры интерактивной модели подбирались с учетом ограничений для достижения максимально возможной реалистичности поведения. Не смотря на это, возможностей платформы Virtools оказалось достаточно для достижения поставленной цели.

Платформа Virtools обладает широкими возможностями для создания интерактивных трехмерных приложений. Визуальное программирование (использование «строительных» блоков) существенно облегчает работу со сценариями.

Библиографический список

1. Официальный сайт производителя платформы Virtools [электронный ресурс]. Режим доступа: <http://www.3dvia.com/>, свободный.
2. Официальный сайт разработчика движка Havok [электронный ресурс]. Режим доступа: <http://www.havok.com/>, свободный.
3. Официальный российский сайт Autodesk [электронный ресурс]. Режим доступа: <http://www.autodesk.ru>, свободный.

4. Официальный сайт разработчика DirectX [электронный ресурс]. Режим доступа: <http://www.gamesforwindows.com/en-US/directx/>, свободный.
5. Свободная энциклопедия Википедия [электронный ресурс]. Режим доступа: <http://ru.wikipedia.org/>, свободный.
6. Российский сайт разработчиков компьютерных игр [электронный ресурс]. Режим доступа: <http://www.gamedev.ru>, свободный.
7. Сайт разработчиков компьютерных игр [электронный ресурс]. Режим доступа: <http://www.gamedev.net>, свободный.
8. Королев С. Основы Virtools: 2-я ред., 2008.

УДК 534.78:621.39

В. И. Шишко – магистрант кафедры моделирования вычислительных и электронных систем

О. О. Жаринов (канд. техн. наук, доц.) – научный руководитель

РАЗРАБОТКА АЛГОРИТМА РАСПОЗНАВАНИЯ ГОЛОСОВЫХ КОМАНД

Характерной чертой современного этапа в развитии вычислительной техники является ее широкое проникновение в различные сферы человеческой деятельности. Это приводит к тому, что круг лиц, использующих ЭВМ в своей повседневной жизни, непрерывно и очень быстро возрастает. Отсюда возникает одна из важнейших задач современного этапа компьютеризации общественного процесса – максимальное упрощение процесса «общения» человека с современными вычислительными машинами. Создание систем, способных распознавать и понимать человеческую речь, является одним из главных путей ее решения. В настоящее время речевое распознавание находит все новые и новые области применения, начиная от приложений, осуществляющих преобразование речевой информации в текст и заканчивая бортовыми устройствами управления автомобилем. Поэтому распознавание звуковых образов – принадлежит к наиболее перспективным направлениям совершенствования мультимедийных технологий.

Актуальность проведения исследования и разработки нового алгоритма распознавания голосовых команд связана с тем, что представленные на рынке аналоги не всегда соответствуют желаниям пользователя. Во-первых, иной раз нет необходимости в использовании программы, обладающей большим количеством распознаваемых команд, т. е. достаточно более простого варианта и, соответственно, стоимость такой программы будет меньше ей более «мощного» аналога. Во-вторых, большинство программ функционирует с помощью компьютера. Поэтому может быть несовместимость технических платформ пользователя и предлагаемой программы. Программа, требующая меньше вычислительных ресурсов, будет универсальной, применяемой в процессоре любой мощности. Задача заключается в разработке и исследовании алгоритма распознавания голосовых команд.

Предметом обработки и исследования при разработке алгоритма распознавания голосовых команд является речевой сигнал. Речевой сигнал представляет собой электрическое колебание, полученное в результате преобразования акустической речевой волны. Смысловое содержание акустической речевой волны неизвестно и, в общем случае, одному смысловому содержанию в разных экспериментах могут соответствовать различные электрические колебания, отличающиеся временной структурой, длительностью, формой и т.п. поэтому речевой сигнал следует рассматривать как случайный процесс. Случайный процесс как совокупность случайных величин задается вероятностными характеристиками – функциями плотности вероятностей, интегральными

функциями распределения вероятностей, моментными функциями, корреляционной функцией, и функцией спектральной плотности мощности. Речевой сигнал является нестационарным случайным процессом. Поэтому оценивание его вероятностных характеристик, таких как функции плотности вероятностей (ФПВ), спектральной плотности мощности, корреляционной функции и т.п., строго говоря, нельзя выполнять усреднением по времени одной реализации. Такое усреднение можно осуществлять лишь на коротких промежутках времени. На основании многочисленных исследований было доказано, что наибольшую информацию дают спектральные характеристики, в частности функция спектральной плотности мощности (СПМ), и поэтому она была выбрана для построения алгоритма распознавания речевых команд.

Предлагаемая концепция обработки речевого сигнала выглядит следующим образом.

В первую очередь необходимо задать словарь команд, подлежащих распознаванию. В данном случае выбрано множество из 5 команд («один», «два», «три», «четыре», «пять»). На этапе обучения алгоритма требуется вычислить характеристики команд, входящих в словарь. Эти характеристики будут служить эталонами для сравнения. Для того чтобы получить функцию спектральной плотности мощности (СПМ), одна команда записывалась многократно, и каждый раз вычислялся спектр (БПФ) и потом усреднялись спектры в каждой точке. В процессе распознавания сигнала нужно сформировать выборку речевого сигнала, подлежащую анализу. По полученной выборке вычислить множество решающих статистик, каждая из которых характеризует различие функций СПМ проверяемой неизвестной команды и каждого из элементов словаря. Решение принимается в пользу минимального значения решающей статистики.

Характеристики алгоритма распознавания в основном определяются видом используемой решающей статистики. В процессе произведенного исследования мною было рассмотрено несколько вариантов решающих статистик. В литературе[1] рекомендуется к применению следующий вариант:

$$\gamma 1_{i,x} = \sum_{f=f_{\text{нач}}}^{f_{\text{кон}}} \frac{G_x(f)}{G_i(f)} + \ln \frac{G_i(f)}{G_x(f)}, \quad (1)$$

где $G_x(f)$ – выборочная оценка спектральной плотности мощности распознаваемого сигнала x ; $G_i(f)$ – оценка спектральной плотности мощности сигнала i из словаря; $f_{\text{кон}}$ и $f_{\text{нач}}$ – границы полосы анализируемых частот (чтобы уменьшить влияние помех).

Заметим, что выбор оптимальных значений $f_{\text{кон}}$ и $f_{\text{нач}}$ является самостоятельной задачей.

В результате проведенного исследования выяснилось, что данная решающая статистика не обеспечивает необходимой достоверности распознавания.

Если проанализировать причины, то можно увидеть, что логарифм приводит к отрицательным значениям. Чтобы избежать этого явления, я решил немного подкорректировать формулу(1), в результате решающая статистика приобретает следующий вид:

$$\gamma 2_{i,x} = \sum_{f=f_{\text{нач}}}^{f_{\text{кон}}} \frac{G_x(f)}{G_i(f)} + \frac{G_i(f)}{G_x(f)}, \quad (2)$$

Результаты оказались намного лучше.

Эффективность распознавания команд во многом также зависит от способа и деталей процесса получения оценок спектральной плотности мощности. Первая проблема, с которой сталкиваются разработчики алгоритмов обработки речевых сигналов – это то, что процесс не гауссовский. Следуя рекомендациям литературы, чтобы приблизить процесс к гауссовскому, обычно применяют нелинейное преобразование отсчетов про-

цесса. Согласно одной из рекомендаций [2], для речевых сигналов можно применить преобразование вида

$$\psi(x) = \sqrt[3]{x}$$

Чтобы снизить влияние громкости на полученные результаты, было использовано стандартное решение – решение применить нормировку функций спектральной плотности мощности:

$$G_{\text{норм}}(f) = \frac{G_{\text{исх}}(f)}{\sqrt{\sum_{f=0}^{M-1} [G_{\text{исх}}(f)]}}$$

Для иллюстрации рассмотрим пример работы в режиме распознавания. Был создан словарь из 5 команд, то есть им соответствующие 5 эталонов $G_i(f)$. В данном опыте в качестве команд, которые входят в словарь, были выбраны цифры от 1 до 5, записанные на русском языке – «один», «два», «три», «четыре», «пять». Реализация речевого сигнала одной из команд показана на (рис.1).

После применения нелинейного преобразования, высокоамплитудные выбросы (пики) оказываются уменьшенными. Это можно увидеть на (рис. 2).

Кроме всего прочего, при спектральном анализе нестационарного процесса обязательно надо использовать функцию окна, которая позволяет выделить интервалы для анализа, в пределах которого процесс может считаться стационарным. В работе выбрано гауссовское окно:

$$w(t) = e^{-\alpha t^2},$$

где ширина окна α подбирается опытным путем.



Рис. 1



Рис. 2



Рис. 3

Один из фрагментов процесса, получающегося после применения функции окна к сигналу, изображенному на рис. 2 приведен на рис.3.

Для оценивания эффективности алгоритма распознавания предъявлялись 5 команд (по очереди) и формировалась матрица из значений решающей статистики.

$$\gamma_{i,x} = \begin{bmatrix} \gamma_{1,1} & \gamma_{2,1} & \gamma_{3,1} & \gamma_{4,1} & \gamma_{5,1} \\ \gamma_{1,2} & \gamma_{2,2} & \gamma_{3,2} & \gamma_{4,2} & \gamma_{5,2} \\ \gamma_{1,3} & \gamma_{2,3} & \gamma_{3,3} & \gamma_{4,3} & \gamma_{5,3} \\ \gamma_{1,4} & \gamma_{2,4} & \gamma_{3,4} & \gamma_{4,4} & \gamma_{5,4} \\ \gamma_{1,5} & \gamma_{2,5} & \gamma_{3,5} & \gamma_{4,5} & \gamma_{5,5} \end{bmatrix}$$

В соответствии с принятым решающим правилом алгоритма распознавания $x = \arg \min_{\forall i} (\gamma_{i,x})$ то значение $\gamma_{i,x}$, которому соответствует минимальное отклонение, будет соответствовать номеру команды. Следовательно, распознавание считается успешным, если в каждой строке $\gamma_{i,x} = \min$ при $x = i$.

Данное исследование посвящено разработке и исследованию алгоритма распознавания голосовых команд. В исследовании были рассмотрены методы математической обработки речевых сигналов, проанализированы их достоинства и недостатки. На основании проведенного анализа выбран метод, наилучшим образом применяемый для обработки голосовых команд – спектральный метод. Алгоритм распознавания был полностью реализован в среде Mathcad, были подобраны оптимальные значения ключевых параметров алгоритма, и оценена его эффективность при обработке реальных речевых команд.

Библиографический список

1. Бондарев В. Н., Трёстер Г., Чернега В. С. Цифровая обработка сигналов: методы и средства: учеб. пособие для вузов. 2-е изд. Х.: Конус, 2001. 398 с.
2. Сапожков М. А., Михайлов В. Г. Вокодерная связь. М.: Радио и связь, 1983. 248 с.

УДК 004.928

А. А. Цицинова – магистрант кафедры вычислительных систем и сетей
Н. Н. Решетникова (канд. тех. наук, доц.) – научный руководитель

**ИЗУЧЕНИЕ ОТКРЫТЫХ СТАНДАРТОВ ТРЕХМЕРНОЙ ГРАФИКИ
 В СИСТЕМЕ ДИСТАНЦИОННОГО ОБУЧЕНИЯ MOODLE**

Истоки разработок в области внедрения третьего измерения во Всемирную Сеть Интернет берут свое начало с середины 90х годов XX столетия, когда Марком Пеши (Mark Pesce) был предложен «язык моделирования виртуальной реальности» VRML [1]. Это был первый шаг в направлении перевода информации, предоставляемой в Сети на качественно новый уровень. Области применения трехмерного Интернета и по сей день остаются довольно обширными – это и интернет-коммерция, и медицина, и образование, и область развлечений.

Открытые стандарты позволяют создавать пользователю собственные трехмерные миры и просматривать их в окне web-браузера. В данной работе рассматриваются

следующие открытые стандарты: Java3D и X3D. Java 3D является стандартным расширением языка программирования Java, а X3D – это наследник языка VRML, тесно интегрированный с расширяемым языком разметки XML. Поход к описанию трехмерных сцен у этих языков во многом схож: файлы имеют древовидную структуру, включающую в себя узлы и группы узлов с трехмерными объектами, описанием внешнего вида этих объектов, добавлением анимации и интерактивности. Ниже (табл. 1) приведен краткий обзор возможностей рассматриваемых стандартов.

Таблица 1

Возможности открытых стандартов

	Java3D	X3D
Набор стандартных примитивов	+	+
Возможность создания сложных геометрических фигур и поверхностей	+	+
Создание текста	+	+
Задание материалов объектов	+	+
Наложения текстур	+	+
Создание и настройка освещения трехмерной сцены	+	+
Создание анимации	+	+
Настройка окружения	+	+
Использование системы частиц	+	-
Подгрузка трехмерных сцен из внешних файлов	.OBJ, .VRML и т. п.	.X3D
Просмотр в Web-браузере	В качестве апплетов	С помощью специальных плееров (BS Contact Player, Vivaty Player, Octaga Player и др.)

Несмотря на доступность этих стандартов, возникает трудность в изучении их синтаксиса, ввиду отсутствия учебных пособий на русском языке.

Так возникла идея создания не просто учебного пособия, а целого курса, который позволил бы начать знакомство с данными открытыми стандартами «с нуля», а также демонстрирующего наряду с теорией и практическое применение изложенного материала. Очевидно, что для реализации такого пособия теоретический материал должен быть представлен именно в той среде, в которой возможна визуализация готовых трехмерных сцен, а именно – в WEB.

В качестве основы для создания учебного пособия был выбран сайт разработчика языка X3D Дона Бруцмана (Don Brutzman) [2], автора книги «X3D: Extensible 3D Graphics for Web Author». На сайте автор представил главы из книги с описанием возможностей стандарта, примеры кода трехмерных моделей, а также непосредственно и сами трехмерные модели.

Для начинающих изучать Java 3D, помимо официального сайта разработчика, с представленной на нем документацией [3], существует обучающий сайт www.java3d.org, на котором также можно посмотреть примеры апплетов, созданных при помощи данного стандарта. Но приведенных трехмерных сцен на данном сайте немного, поэтому требуется дополнительная разработка примеров для полноты восприятия возможностей языка Java 3D.

Существуют различные программные средства, позволяющие быстро и удобно разработать электронные пособия. Учебная среда изначально несет в себе понятие

не только полноценно предоставить пользователю изучаемый материал, но и предоставить ему возможность обратной связи с системой обучения, контроля знаний и регистрации успеваемости. Здесь нельзя не отметить интенсивно развивающиеся в настоящее время системы дистанционного обучения. Реализация такой формы обучения возможна с помощью систем управления обучением (Learning Management System), которые позволяют разрабатывать, управлять и распространять учебные материалы, обеспечивая при этом совместный доступ. Ниже приведена сравнительная таблица систем дистанционного обучения (табл. 2).

Таблица 2

Сравнительная таблица систем дистанционного обучения

	MOODLE	LAMS	Sakai	ATutor	Claroline	Dokeos	OLAT	OpenACS	ILIAS
SCORM	+	-	+	+	+	+	+	-	+
IMS	+	-	+	+	+	+	+	-	-
Языки-приложения	PHP	Java	Java	PHP	PHP	PHP	Java		PHP
СУБД	MySQL	MySQL	MySQL, Oracle, hsqldb	MySQL	MySQL	MySQL	MySQL, PostgreSQL	Oracle, PostgreSQL	MySQL
Лицензии	GNU/GPL	GNU/GPL	GNU/GPL	GNU/GPL	GNU/GPL	GNU/GPL	GNU/GPL	GNU/GPL	GNU/GPL
Русский язык	+	-	+	+	+	+	+	+	+
Другие языки	>54	20	28	>50	36	38	34	35	43
Система проверки знаний	Тесты, Задания, Семинары, Активность на форумах	тесты	Тесты, Задания, Активность на форумах	тесты	Тесты, упражнения	Тесты	Тесты, задания	Тесты	Тесты
Демонстрационный сервер	-	+	-	+	+	+	+	-	-

На сегодняшний день одной из самых популярных систем дистанционного обучения является Moodle [4].

Moodle – это бесплатно распространяемый программный комплекс с открытыми исходными кодами. По своим функциональным возможностям, простоте освоения и удобству использования он удовлетворяет большинству требований, предъявляемых

пользователями к системам электронного обучения [5]. Именно эта система дистанционного обучения была выбрана для реализации учебного пособия. Moodle позволяет создавать виртуальную образовательную среду, курс которой можно наполнить лекциями с трехмерными моделями, системой контроля знаний, форумами, чатами, опросами, wiki и т.п. По ходу создания теоретического материала трехмерные модели добавлялись на html-страницы при помощи следующих тегов:

Для Java-апплетов:

```
<applet code="File_Name" width="400" height="400"></applet>
```

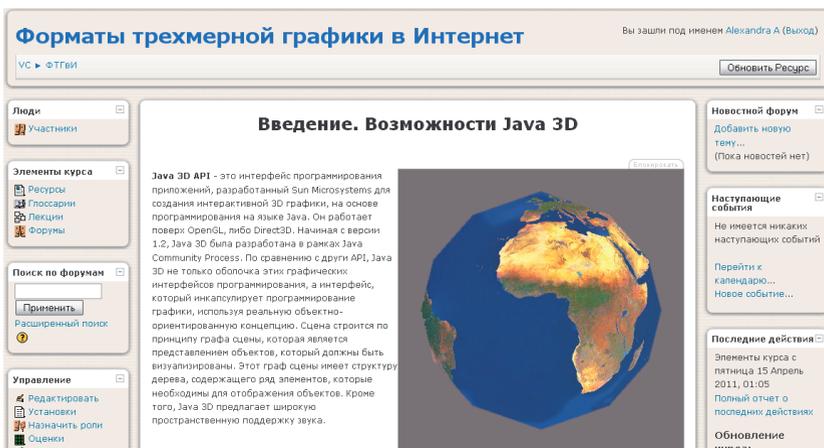


Рис. 1. Фрагмент учебного пособия 1

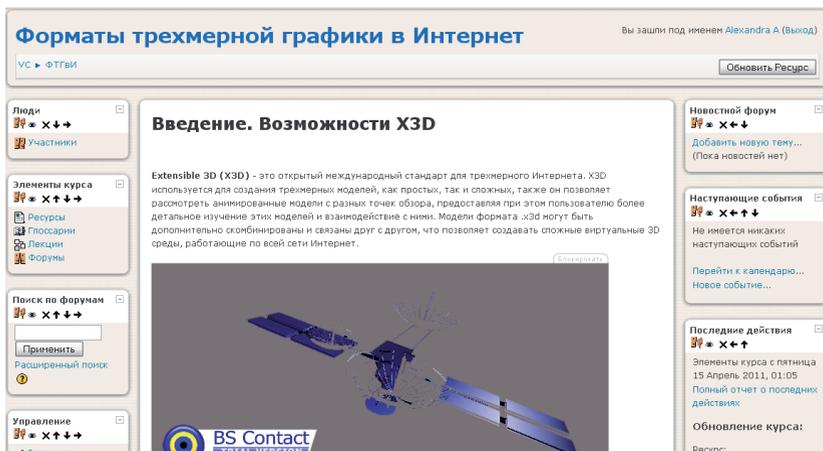


Рис. 2. Фрагмент учебного пособия 2

Для X3D-файлов:

```
<object height="240" width="360" data="http://moodle/mod/resource/File_Name.x3d" type="model/x3d+xml">&nbspnbspnbsp; </object>
```

Поскольку Moodle функционирует на основе виртуального сервера (в данном случае использовался denwer), то для корректного отображения трехмерных моделей, исходные файлы необходимо поместить в каталоге местоположения самого курса в Moodle.

Для создания и просмотра трехмерных сцен на Java 3D было установлено специальное API: JDK 6 и Java 3D. Для воспроизведения X3D-файлов дополнительно был установлен BS Contact Player.

На рис. 1 и 2 приведены фрагменты учебного пособия, созданного в системе дистанционного обучения Moodle с практическими примерами.

В заключение следует отметить, что созданное учебное пособие соответствует поставленным задачам: 1) методический материал изложен полноценно и последовательно в соответствие с иностранными аналогами документации и описания языков Java 3D^[3] и X3D^[2], примеры кодов программ сопровождаются их графической трехмерной реализацией, учебное пособие реализовано в системе дистанционного обучения Moodle; 2) Организация учебного процесса в среде Moodle ориентирована на взаимодействие между преподавателем и учениками, что позволяет добавить в уже созданные учебные материалы систему контроля знаний, обратную связь, форумы для обсуждения, разграничение доступа и т.д., что составит полноценный учебный курс по изучению открытых стандартов.

Библиографический список

1. Web3D Consortium – VRML Archives. – URL: <http://www.web3d.org/x3d/vrml/>
2. X3D: Extensible 3D Graphics for Web Authors. – URL: <http://x3dgraphics.com/>
3. Java 3D API Tutorial. – URL: <http://java.sun.com/developer/onlineTraining/java3d/index.html>
4. Moodle.org: open-source community-based tools for learning. – URL: <http://moodle.org/>
5. Анисимов А. М. Работа в системе дистанционного обучения Moodle: учебн. пособие, 2-е изд., испр. и доп. Харьков: ЦНИТ ХНАГХ, 2009. 292 с.

УДК 612.821

Д. Д. Эхер – студент кафедры моделирования вычислительных и электронных систем
В. И. Исаков (канд. техн. наук, доц.) – научный руководитель

ПРОГРАММНЫЙ КОМПЛЕКС ОБРАБОТКИ БИОЛОГИЧЕСКИХ СИГНАЛОВ

В настоящее время наиболее распространенным неинвазивным методом исследования головного мозга является электроэнцефалография (ЭЭГ). ЭЭГ – метод регистрации электрической активности (биопотенциалов) мозга через неповрежденные покровы головы, позволяющий судить о его физиологической зрелости, функциональном состоянии, наличии очаговых поражений, общемозговых расстройств и их характере. Она широко используется для изучения физиологических процессов головного мозга, и для диагностики неврологических заболеваний. Электроэнцефалограмма отражает электрическую активность головного мозга и является одним из самых непредсказуемых биологических процессов. В основе колебаний потенциалов, регистрируемых с поверхности головы в виде ЭЭГ, лежат изменения внутриклеточных мембранных потенциалов (МП) корковых пирамидных нейронов. При изменении внутриклеточного МП нейрона во внеклеточном пространстве, где расположены глиальные клетки, возникает разность потенциалов – фокальный потенциал. Потенциалы, возникающие во

внеклеточном пространстве в популяции нейронов, представляя собой сумму таких отдельных фокальных потенциалов. Суммарные фокальные потенциалы могут быть зарегистрированы с помощью электропроводных датчиков от разных структур мозга, от поверхности коры или с поверхности черепа. Напряжение токов головного мозга составляет порядка 10^{-5} В. ЭЭГ представляет собой запись суммарной электрической активности клеток полушарий мозга [1, 2].

Для получения экспериментальных данных использован аппаратно-программный электроэнцефалографический комплекс «Мицар-ЭЭГ 202», соответствующий следующим требованиям: количество электродов ЭЭГ не менее 18, наличие кардиоканала, наличие полиграфических каналов, наличие фотостимулятора, возможность видеомониторинга, возможность многостороннего математического анализа [3, 4].

Вследствие нестационарности сигналов ЭЭГ имеют место определенные ограничения при использовании для их обработки традиционного математического аппарата и метода Фурье-анализа. Весьма перспективным из существующих на данный момент способов анализа, очистки и сжатия сигналов электроэнцефалограмм является вейвлет-преобразование. Теория вейвлетов в настоящее время представляет собой бурно развивающуюся область математики, которая стремительно завоевывает популярность в таких различных областях, как физика, биология, медицина, компьютерная графика, телекоммуникации.

Вейвлетное преобразование информации представляет собою обобщение методов спектрального анализа. Термин «вейвлет» (wavelet) в переводе с английского означает «маленькая (короткая) волна». Вейвлеты – это обобщенное название семейств математических функций определенной формы, которые локальны по времени и частоте, и в которых все функции получаются из одной базовой (порождающей) посредством ее сдвигов и растяжений по оси времени. Вейвлет-преобразования рассматривают анализируемые временные функции в терминах колебаний, локализованных по времени и частоте [5].

При исследовании интеллектуальной деятельности человека и диагностировании состояния головного мозга важной задачей является автоматизация процесса вейвлетной обработки сигналов энцефалограмм.

В настоящее время специальные пакеты расширений по вейвлетному анализу присутствуют в основных системах компьютерной математики (Matlab, Mathematica, Mathcad, и др.). Наиболее широко используется пакет расширения Wavelet Toolbox системы Matlab [6, 7]. Он включает в себя средства для изучения, создания и применения вейвлетов и применения вейвлет-преобразований как в командном режиме, так и с помощью специальных средств графического интерфейса пользователя GUI. Пакет Wavelet Toolbox предоставляет в распоряжение пользователя:

- инструментальные средства для непрерывного и дискретного вейвлет-анализа и синтеза сигналов и изображений;

- встроенные вейвлеты различных типов, в том числе пакетные;
- возможность задания собственных типов вейвлетов;
- средства обработки сигналов;
- средства очистки сигналов и изображений от шумов;
- средства специальной обработки и компрессии данных;
- средства визуализации вейвлетов и всех операций с ними.

Для решения задачи автоматизации вейвлетного анализа электроэнцефалограмм разработан программный комплекс, имеющий широкие возможности по обработке сигналов ЭЭГ с помощью вейвлетного преобразования. Входными данными является временная реализация сигнала ЭЭГ, представляющая собой набор одномерных рядов данных, каждый из которых соответствует определенной участку коры головного мозга. В результате работы вычисляются количественные параметры, характеризующие ЭЭГ человека.

Схема программного комплекса представлена на рис.1.

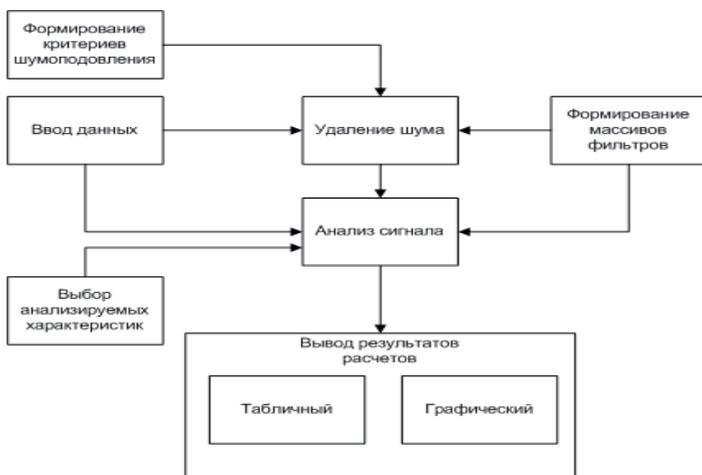


Рис. 1. Схема программного комплекса

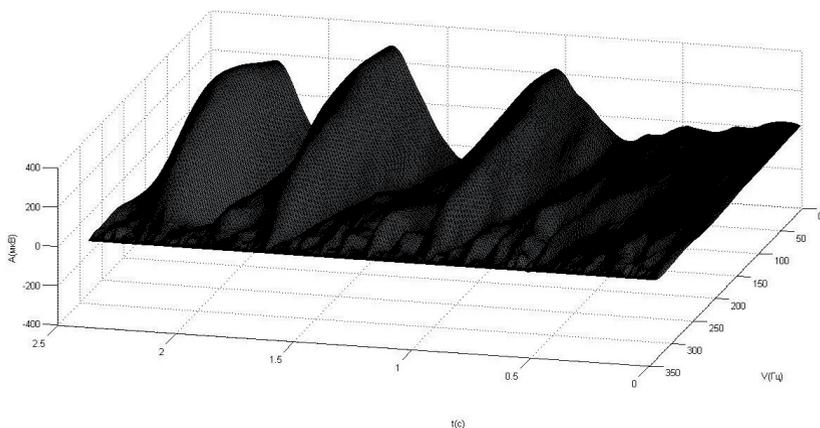


Рис. 2. Трехмерный график матрицы вейвлет-коэффициентов

Программный комплекс позволяет выполнять следующие функции:

- вейвлетное разложение сигналов на аппроксимирующие и детализирующие составляющие,
- расчет статистических характеристик вейвлет-коэффициентов,
- построение энергетических спектров компонент сигналов,
- осуществление непрерывного вейвлет-преобразования сигналов,
- построение спектрограмм сигналов в различных масштабах,
- построение трехмерных графиков матриц вейвлет-коэффициентов,

- осуществление вейвлетной фильтрации сигнала,
- построение графика локальных экстремумов(скелетона) интегрального вейвлет – преобразования сигнала,
- спектральных интегралов вейвлет-преобразования сигнала в различных частотных диапазонах,
- шумоподавление и обработка сигнала.

Программный комплекс дает возможность использовать для анализа данных вейвлеты различных типов.

Результаты расчетов представлены на рис. 2.

Применение вейвлет-преобразования позволяет получить принципиально новую информацию, которая недоступна при использовании стандартных методов анализа ЭЭГ. Разработанный программный комплекс позволяет проводить анализ частотных ритмов сигнала и оценивать переходные процессы нервной системы человека.

Результаты работы могут найти применение в системах медицинской диагностики и при анализе интеллектуальной деятельности человека.

Библиографический список

1. Клиническая электроэнцефалография. / Под ред. Русинова В. С. – М.: Медицина, 1973. 340 с.
2. Сахаров В. Л., Андреев А. С. Методы математической обработки электроэнцефалограмм. Таганрог: Антон, 2000.
3. ООО «Мицар»: электроэнцефалографы «Мицар-ЭЭГ-202». [электронный ресурс] – режим доступа: http://www.mitsar.spb.ru/eeg_202.htm
4. ООО «Мицар»: расширенные возможности ПО «Мицар-ЭЭГ». [электронный ресурс] – режим доступа http://www.mitsar.spb.ru/eeg_ext.htm
5. Блаттер К. Вейвлет-анализ. Основы теории. М.: Техносфера, 2006. 280 с.
6. Смоленцев Н. К. Основы теории вейвлетов. Вейвлеты в Matlab. М.: ДМК Пресс, 2005. 301 с.
7. Дьяконов В. Matlab. Обработка сигналов и изображений. Специальный справочник СПб.: Питер, 2002. 608 с.

УДК 004.732

С. В. Ярошук – студент кафедры вычислительных систем и сетей

А. В. Гордеев (д-р техн. наук, проф.) – научный руководитель

ТЕХНОЛОГИЯ VPN

Виртуальная частная сеть или просто VPN (Virtual Private Network) – это технология, при которой происходит обмен информацией с удаленной локальной сетью по виртуальному каналу через сеть общего пользования с имитацией частного подключения «точка-точка» или «сервер-клиент». Передаваемая по этому виртуальному каналу информация шифруется.

Существует большое количество вариантов программного обеспечения, позволяющего создавать виртуальные частные сети.

Рассмотрим один из вариантов — пакет OpenVPN. «OpenVPN» – свободно распространяемая программа для создания виртуальных частных сетей (VPN). Виртуальная частная сеть, построенная с использованием OpenVPN, представляет собой зашифрованный канал связи типа «точка-точка» или несколько зашифрованных каналов связи между сервером и несколькими клиентами, называемый «сервер-клиент» [2,3].

Основные преимущества «OpenVPN»:

- поддержка разнообразных операционных систем (Linux, FreeBSD, Windows, Solaris, Mac OS X, NetBSD и т.п.). Существует версия, портированная на Windows Mobile.

- работа по произвольному TCP/UDP порту, а также через HTTP прокси-сервер. Полезно при наличии ограничений по протоколам и портам, но вносит небольшое количество избыточной информации за счет заголовков TCP или UDP. В отличие от протокола PPTP, используется только одно соединение для передачи данных и управляющих команд. Рекомендуется использовать протокол UDP, если это возможно.

- поддержка разных режимов защиты канала связи (по общему ключу – поддерживает только каналы точка-точка, проще в настройке, или по сертификату – поддерживает подключение нескольких клиентов к серверу, сложнее в настройке), а также алгоритмов шифрования. В режиме точка-точка шифрование может отключаться.

Пример сетей, в которых может быть использована технология «OpenVPN».

В рассматриваемом примере технология VPN используется для удаленного доступа к ресурсам внутри сети, а также к Интернет-подключению. Она используется для решения трех задач:

Доступ к файлам и другим ресурсам внутри локальной сети, к которой подключен VPN-сервер.

- Подключение к Интернету через локальную сеть провайдера (выгодно, если трафик локальной сети не тарифицируется или стоит дешевле, чем Интернет-трафик). Требуется, чтобы сервер был подключен к локальной сети того же провайдера, что и клиент. Может использоваться с различными технологиями подключения, включая Ethernet, Wi-Fi и GPRS/EDGE/3G.

- Помимо этого, можно найти большое число применений для технологии VPN, включая доступ к Интернету на стороне провайдера, игра по Интернету в игры, рассчитанные на игру по локальной сети, замена внешнего IP-адреса (включая получение внешнего IP-адреса в том случае, если провайдер его не дает), обход ограничений, установленных в локальной сети, объединение удаленных локальных сетей в одну сеть и т. д [4].

Что касается настройки непосредственно самого соединения, то с соединением «точка-точка» никаких проблем не возникает, а соединение «сервер-клиент» вызывает несколько вопросов.

Основной и пожалуй самый важный вопрос, это вопрос безопасности и шифрования данных. Так как в этом режиме необходимо, вместо ключей для аутентификации и шифрования использовать сертификаты. А что такое сертификаты и зачем их надо использовать при данном виде подключения VPN?

Сертификат – файл, хранящий информацию об источнике информации и его открытый ключ. Для установления того, что он получен от того источника, за которого себя выдает, сертификат подписывается ключом сертификационного центра. Таким образом, для проверки действительности сертификата необходимо проверить его подпись с использованием сертификата центра сертификации (а также дополнительные условия: срок действия, отсутствие сертификата в списке недействительных). В ответственных случаях центрами сертификации являются компании, подписывающие сертификаты и отвечающие за правильность представленной в сертификате информации.

При работе с сертификатами используются алгоритмы асимметричного шифрования, на которых и основывается метод электронной цифровой подписи – ЭЦП; она позволяет установить отсутствие искажения информации в электронном документе с момента формирования ЭЦП и проверить принадлежность подписи владельцу сертификата ключа ЭЦП. Значение реквизита получается в результате криптографического преобразования информации с использованием закрытого ключа ЭЦП. С помощью такого алгоритма можно установить, что информация получена от доверенного источника. Для этого источник формирует документ, получает от него контрольную

хеш-функцию и шифрует хеш с помощью закрытого (секретного) ключа. В этом случае говорят, что источник подписывает документ. Ключ расшифровки делается открытым (публичным). Для проверки того, что документ был получен от доверенного источника и не был изменен, получатель расшифровывает хеш с помощью открытого ключа и вычисляет хеш-функцию документа. Если хеши совпадают, значит документу можно доверять. Получить закрытый ключ на основе открытого ключа практически невозможно [5, 6].

Основной проблемой при создании соединения «сервер-клиент» является как раз выше описанная система сертификатов. Эту задачу и надо решить, для упрощения настройки данного соединения при помощи программного обеспечения «OpenVPN».

Для решения данной проблемы, было проведено сравнение (Таблица) различных способов создания и применения сертификатов для подписания программного обеспечения.

После сравнения способов создания сертификатов, было решено использовать создание собственного центра сертификации.

Потому что данный способ имеет несколько неоспоримых преимуществ:

- можно генерировать ключи самостоятельно,
- можно подписывать любое ПО, где используется серверная и клиентская часть,
- можно создавать персональные сертификаты.

Для создания собственного центра сертификации, можно воспользоваться пакетом OpenSSL [7]. С его помощью можно создавать ключи шифрования, сертификаты, а также подписывать сертификаты. Что касается самого центра сертификации, в который мы и будем добавлять созданные сертификаты и ключи, то его необходимо создать самостоятельно (табл. 1).

Таблица 1

Сравнение способов создания ключей и сертификатов

	Заказ сертификата у «Удостоверяющего центра»	Использование Easy-rsa	Создание своих сертификатов
Применение ключей и сертификатов	Можно подписывать любое ПО, где используется серверная и клиентская часть	Используется только для подписания OpenVPN	Можно генерировать ключи и подписывать любое ПО, где используется серверная и клиентская часть
Безопасность	Ключи генерируются в зависимости от желаний заказчика	Генерируется ключ длиной до 1024 бит	Возможность генерации ключей до 2048 бит. (Параметр указывается самостоятельно)
Возможность развития организации	Заказчик заказывает ограниченное количество сертификатов	Есть возможность создания новых сертификатов при увеличении количества клиентов	Есть возможность создания новых сертификатов при увеличении количества клиентов, а также возможно создание персональных сертификатов сотрудников

Таблица 1 (окончание)

	Заказ сертификата у «Удостоверяющего центра»	Использование Easy-rsa	Создание своих сертификатов
Распространение	Заказывается и покупается у «Удостоверяющего центра».	Пакет бесплатный.	Пакет бесплатный.
Основные недостатки	Высокая стоимость данной услуги	Узкая направленность (только для OpenVPN)	Требует большой работы от администратора

В результате проведенных сравнений, было решено, что самым лучшим и эффективным является метод создания собственных сертификатов.

Библиографический список

1. Norbert Graf, Beginning OpenVPN 2.0.9, техническая литература, изд. Packt, 2009, стр. 356.
2. Markus Feilner, OpenVPN: Building and Integrating Virtual Private Networks, техническая литература, изд. Packt Publishing, 2006, стр. 223.
3. <http://openvpn.net>;
4. <http://ru.wikipedia.org/wiki/OpenVPN>;
5. <https://help.ubuntu.com/community/OpenVPN>;
6. <http://wiki.kryukov.biz/wiki/Openvpn>;
7. <http://ru.wikipedia.org/wiki/OpenSSL>.