

Секция

КОМПЬЮТЕРНЫЕ СИСТЕМЫ И СЕТИ

Компьютерные вибродиагностические системы	430
П.Ю. Бранцевич, С.Ф. Костюк, Е.Н. Базылев, В.Э. Базаревский	
Гибридный алгоритм детектирования дыма и открытого пламени для систем видеонаблюдения	432
Р.П. Богуш, Д.А. Тычко	
Актуальные проблемы применения нейрорегуляторов в системах автоматического управления	434
А.С. Михайлов	
Использование физически неклонированных функций для решения задачи генерирования действительно случайных чисел и идентификации	436
С.С. Заливако, А.А. Иванюк	
Тестовая система в архитектуре клиент/сервер	438
А.М. Седун, М.Н. Садовская, В.С. Оскерко	
Обеспечение надежности программных средств при их модификациях	440
Н.В. Успенская, В.В. Бахтизин	
Базовые примитивы схемной обфускации цифровых устройств	442
В.В. Сергейчик, А.А. Иванюк	
Применение статического анализа исходных кодов в оценке надёжности web-приложений	444
Д.Е. Оношко, В.В. Бахтизин	
Выбор параметров проектируемой системы фазовой синхронизации	446
Д.Л. Шилин, В.В. Пучинец, Л.Ю. Шилин	
Принципы математического описания систем фазовой синхронизации	448
М.П. Батура, А.П. Кузнецов, Л.Ю. Шилин, Д.П. Кукин, Д.Л. Шилин	
Создание учебно-научного портала ядерных знаний	450
И.Я. Дубовская, Т.А. Савицкая, А.С. Лобко, С.Н. Сытова, С.В. Черепица	
Построение трехмерных моделей по изображениям с различных камер	452
И.Н. Губчик	
Анализ требований на основе UML-моделей и динамических HTML-прототипов	454
Н.Б. Киреев	
Comparison of nehalem and IVY bridge TLB architectures	456
M. Askari, N.N. Ivanov	
Кластерный анализ содержимого сайта вопросов-ответов STACKEXCHANGE.COM	458
К.Ю. Слисенко, С.И. Сиротко	
Технология тестирования web-приложений на основе вероятностных сетей	460
А.А. Быков, И.И. Пилецкий	
Программно-аппаратная платформа для построения видеосистем	462
С.А. Байрак, Е.В. Калабухов, М.М. Татур	
Архитектура аналитической платформы для встраиваемых интеллектуальных подсистем	464
В.С. Абатуров, А.Ю. Дорогов, О.В. Забродин, И.В. Раков	

Защита исходного кода веб-приложений методом лексической обфускации JAVASCRIPT-кода	466
М.А. Бартошик, В.Н. Ярмолик	
Принципы улучшения качества спецификации требований программных средств	468
А.Ю. Чиркова, В.В. Бахтизин	
Система централизованного управления учетными записями и правами доступа SATORI	470
М.В. Стержанов	
Масштабируемые веб-приложения	472
А.П. Шкор	
Имитация управленческой деятельности с использованием элементов рефлексии	473
Е.В. Рулько	
Верификация проектов СБИС: проблемы и решения	475
Л.А. Золоторевич	
Особенности нелинейных фильтров в прикладных задачах.....	477
О.П. Соловей, Н.Н. Иванов	
О приводимости некоторых динамических систем с хаотическим поведением к автомодельному уравнению Курамото – Сивашинского.....	479
В.В. Цегельник	
Межканальная спектральная корреляция в задачах сжатия гиперспектральных изображений.....	481
Д.Ю. Перцев, А.А. Дудкин	
К моделированию поведения вирусов в компьютерной сети.....	483
А.В. Борзенков, О.Л. Коновалов	
Применение XSLT преобразований в аналитической СУБД.....	484
Д.Д. Черкасов, О.В. Забродин	
APPS: Adaptive Priority Based Packet Scheduling methode in IP	486
Seyed Enayatallah Alavi, Marjan Naderan-Tahan, Mohamad Aminian	
Программный комплекс для разработки и тестирования систем массового обслуживания.....	488
А.И. Клименко, В.С. Бурчик	
Параллельная обработка запросов систем управления идентификационными данными.....	490
П.П. Пинюта, В.В. Бахтизин	
Управление качеством мобильных приложений.....	492
А.А. Шелкович, В.В. Бахтизин	
Рекурсивный алгоритм сравнения деревьев.....	494
А.В. Лычковский	
Метод классификации текстов на русском языке.....	496
Ф.И. Третьяков, Л.В. Серебряная	
Алгоритм текстурного анализа изображений.....	498
М.М. Лукашевич	
Логико-вероятностные подходы в автоматизации моделирования структурно-сложных систем.....	500
Н.В. Лапицкая	
Тестирование производительности веб-ориентированных приложений.....	502
Ю.П. Курмаз, С.С. Куликов	

КОМПЬЮТЕРНЫЕ ВИБРОДИАГНОСТИЧЕСКИЕ СИСТЕМЫ

П.Ю. БРАНЦЕВИЧ, С.Ф. КОСТЮК, Е.Н. БАЗЫЛЕВ, В.Э. БАЗАРЕВСКИЙ

*Белорусский государственный университет информатики и радиоэлектроники
ул. П. Бровки, 6, г. Минск, 220013, Республика Беларусь
branc@bsuir.edu.by*

Рассмотрены вопросы организации систем вибрационного контроля, мониторинга, диагностики, построенных по принципу компьютерных приборов. Представлены разработки лаборатории вибродиагностических систем БГУИР, решающие данные задачи. Предложена организация системы распределенного сбора и централизованной или распределенной обработки вибрационных данных на базе интернет-ресурса.

Ключевые слова: вибрация, мониторинг, технология, компьютер, интернет.

Правила эксплуатации сложных механизмов и агрегатов на предприятиях энергетики, нефтехимии, транспортировки газа требуют их оснащения штатными системами вибрационного контроля, мониторинга, защиты. А современные технологии эксплуатации такого оборудования предполагают планирование работ по техническому обслуживанию и ремонту в соответствии с их реальным техническим состоянием, что требует решение задач технической диагностики.

Научно-исследовательская лаборатория вибродиагностических систем БГУИР более двадцати лет занимается разработкой систем, комплексов, приборов, программного обеспечения и методик их применения в данной области.

В лаборатории разработан и производится многоканальный измерительно-вычислительный комплекс (ИВК) «Лукомль-2001», который эксплуатируется более пятнадцати лет на большинстве крупных турбоагрегатов Беларуси [1,2]. Структурно ИВК представляет собой универсальную ПЭВМ с типизированным модулем АЦП, подключаемым к ее стандартному интерфейсу (ISA, PCI, USB), блока аналоговой обработки сигналов, принимаемых от первичных виброизмерительных каналов, и блока управления сигнализацией и защитным отключением. При таком построении основная функциональность комплекса обеспечивается алгоритмическим и программным обеспечением [3].

Отличительная особенность комплекса «Лукомль» – реализация алгоритмов защиты технических объектов по вибрационным параметрам не только по стандартизованным критериям, но и с учетом расширенного числа показателей вибрации, индивидуальных особенностей конкретного объекта и обобщенной оценки ситуации на объекте. При анализе вибрационного состояния защищаемого объекта учитываются факторы низкочастотной вибрации, высокочастотной вибрации, оборотной составляющей вибрации, изменение вектора оборотной составляющей. Значения конкретных уровней срабатывания предупредительной сигнализации и защиты по каждому фактору устанавливаются индивидуально для конкретного агрегата. При принятии решения о защитном отключении учитываются показания, полученные в нескольких точках контроля, и динамика изменения этих показателей во времени [4]. В настоящее время эксплуатируется пять таких систем автоматической защиты по вибрации.

Однако решение задач вибрационной диагностики до настоящего времени остается весьма проблематичным, так как вывод о техническом состоянии объекта только на основе количественных значений вибрационных параметров во многих случаях дос-

таточно неоднозначен. Для более достоверных заключений представляется целесообразным проведение анализа динамики изменения непрерывных вибрационных сигналов, отражающих техническое состояние объекта на достаточно длительном временном интервале (минуты, часы и даже сутки). Для регистрации таких вибрационных сигналов используется измерительно-вычислительный комплекс «Гембр» на базе мобильного компьютера, модуля АЦП с USB интерфейсом, виброизмерительных каналов с первичными пьезоэлектрическими преобразователями и проблемно-ориентированного программного обеспечения [5–7]. При решении задач вибродиагностики применяются разнообразные способы исследования вибрационных сигналов [8].

Расширение пропускной способности каналов передачи данных и развитие компьютерных интернет-технологий позволяют создавать системы поддержки принятия решений по оценке технического состояния сложных механизмов роторного типа на основе распределенного сбора больших объемов виброметрических данных и программных средств как традиционного, так и браузерного исполнения, доступ к которым производится через интернет-ресурс [9].

Разработан прототип подобной системы. Для регистрации длительных реализаций вибрационных сигналов применяется комплекс «Гембр-М» [10]. Обработка длинных реализаций вибрационных осуществляется программным средством, написанном на языке Java, которое может выполняться в браузерах, что позволяет использовать для обработки мобильные платформы и выполнять достаточно сложные цифровые преобразования и анализ данных в любом месте, где имеется мобильная связь [11].

Список литературы

1. *Бранцевич П.Ю.* // Энергетика и ТЭК. 2008, № 12(69). С.19-21.
2. *Бранцевич П.Ю., Костюк С.Ф.* Устройство для измерения параметров вибрации и защиты механизмов с вращательным движением / Патент на полезную модель РБ № 8654.
3. *Бранцевич П.Ю., Кульков Э.И., Костюк С.Ф. и др.* // Наука энергетике 1999-2000. Сборник научных трудов. Минск, 2001. С. 72-81.
4. *Brancevich P., Miao X., Li Y.* // 20th International Congress on Sound and Vibration. Bangkok, 7-11 July 2013. P.n. 528. PP. 1-8.
5. *Бранцевич П.Ю., Жук М.М., Костюк С.Ф. и др.* // Чрезвычайные ситуации: предупреждение и ликвидация: Сборник тезисов докладов 4 Международной научно-практической конференции. Т.1. Минск, 2007. С. 183-186.
6. *Бранцевич П.Ю., Бобрук Е.В.* // Чрезвычайные ситуации: предупреждение и ликвидация. 2012, № 1 (31). С. 5-8.
7. *Бранцевич П.Ю., Костюк С.Ф., Носко Д.В. и др.* Устройство для обработки вибрационных сигналов при динамических испытаниях конструкций и диагностике механизмов с вращательным движением / Патент на полезную модель РБ № 4050.
8. *Бранцевич П.Ю., Гузов В.А.* // Проблемы вибрации, виброналадки, вибромониторинга и диагностики оборудования электрических станций. Сб. докл. Москва, 22-26 октября 2007. С. 58–66.
9. *Базаревский В.Э.* // Доклады БГУИР. 2013, № 1 (71). С.51-57.
10. *Бранцевич П.Ю.* // Topical areas of fundamental and applied research: Proceedings of the Conference. North Charleston, 2013. Vol. 1, pp. 111-116.
11. *Бранцевич П.Ю., Костюк С.Ф., Базылев Е.Н. и др.* // Международный конгресс по информатике: информационные системы и технологии: материалы международн. научн. конгресса. Минск, 4-7 нояб. 2013 г. С. 286-290.

ГИБРИДНЫЙ АЛГОРИТМ ДЕТЕКТИРОВАНИЯ ДЫМА И ОТКРЫТОГО ПЛАМЕНИ ДЛЯ СИСТЕМ ВИДЕОНАБЛЮДЕНИЯ

Р.П. БОГУШ, Д.А. ТЫЧКО

*Полоцкий государственный университет
ул. Блохина, 29, г. Новополоцк, 211440, Республика Беларусь
bogushr@mail.ru, dtychko@gmail.com*

Одной из актуальных прикладных задач, решаемых в последнее время с помощью систем технического зрения и видеонаблюдения, является автоматическое обнаружение пожаров. Существующие алгоритмы обнаружения основных факторов пожаров направлены, как правило, на выявление одного из них. В данной работе предлагается алгоритм позволяющий обнаруживать дым, открытое пламя и их совокупность на динамических изображениях.

Ключевые слова: анализ движения, цветовая сегментация, вейвлет-преобразование.

В связи с интенсивным развитием цифровых телекамер, резким снижением их стоимости и совершенствованием алгоритмического обеспечения обработки динамических изображений, системы видеонаблюдения стали доступны широкому классу пользователей и позволяют эффективно решать многие задачи в практической деятельности человека [1,2]. Развитие таких систем обеспечивается не только улучшением технических характеристик видеокамер, но и автоматизацией выявления интересующих событий, происходящих в зоне наблюдения, которая обеспечивается благодаря развитию методов и алгоритмов обработки последовательностей изображений (динамических изображений), получаемых с видеокамеры. Одной из актуальных прикладных задач, решаемых в последнее время с помощью систем технического зрения и видеонаблюдения, является автоматическое обнаружение пожаров, в том числе и на открытых пространствах. В данном случае методы обработки динамических изображений направлены на автоматическое обнаружение основных факторов пожара, которые можно зафиксировать с помощью видеокамеры – дым и открытое пламя [1]. В связи с высокой актуальностью и сложностью задачи существует ряд алгоритмов обработки динамических изображений для видеодетекторов пожаров [1,3]. Однако они направлены, как правило, на выявление одного из признаков пожара: либо пламени, либо дыма. Это объясняется, в первую очередь, различными характеристиками указанных объектов. В тоже время, анализ развития интеллектуальных видеосистем и соответствующих направлений научных исследований также показывает, что необходимо создание алгоритмов обработки динамических изображений с расширенными функциональными возможностями, которые бы позволяли обнаруживать оба признака пожаров.

Для решения указанной задачи разработан гибридный алгоритм, позволяющий обнаруживать либо дым, либо открытое пламя, либо их совокупность на динамических изображениях. Предложенный алгоритм включает следующие основные операции: цветовая сегментация, предобработка, межкадровое вычисление разности, обновление фона, построение переднего плана, морфологическое открытие и закрытие, контурный анализ, вычисление оптического потока, оценка хаотичности движения, анализ контраста, пространственный и временной вейвлет-анализ.

Движение характерно как для дыма, так и для пламени, поэтому рассматривается алгоритмом, как общий признак, который может использоваться для их обнаружения. Извлечение движущихся пикселей выполняется на основе адаптивного метода вы-

читания фонового кадра. Дальнейшее объединение пикселей в связные области осуществляется с использованием операций математической морфологии и контурного анализа. Отличительной особенностью алгоритма, направленной на снижение вероятности ложного обнаружения, является анализ хаотичности движения в обнаруженных областях на основе векторов движения, определяемых блочным методом вычисления оптического потока. При этом оценивается сонаправленность векторов движения для объекта. С учетом различия яркостно-цветовых свойств признаков пожара, применяется цветовая сегментация для пламени и анализ контраста для дыма. Сегментация для пламени выполняется в цветовом пространстве YCbCr с учетом глобального анализа кадра и локального анализа области-кандидата. Отличительной особенностью алгоритма является также использование на этапе классификации пространственного и временного вейвлет-анализа, как для пламени, так и для дыма. Применение единых приемов обработки динамических изображений для выявления дыма и открытого пламени упрощает структуру алгоритма, снижает сложность реализации и позволяет уменьшить вычислительные затраты. Алгоритм реализован на языке C++ с использованием среды разработки Microsoft Visual 2010 и библиотеки компьютерного зрения Open CV2.2. Эксперименты проведены на видеопоследовательностях, полученных в реальных условиях с использованием видеокамеры Panasonic SDR-S50, а также размещенных на сайтах <http://signal.ee.bilkent.edu.tr/VisiFire/Demo/SampleClips.html>, <http://www.openvisor.org>, <http://cvpr.kmu.ac.kr>. Примеры обнаружения представлены на рис. 1.



Рис. 1. Примеры детектирования дыма и пламени гибридным алгоритмом

Результаты исследований свидетельствуют, что алгоритм обеспечивает достаточно стабильное обнаружение дыма, пламени, либо дыма и пламени в совокупности, в том числе на сложном динамическом фоне и при наличии движущихся объектов, а также объектов со схожими яркостно-цветовыми характеристиками.

Список литературы

1. Лукьяница А.А., Шшикин А.Г. Цифровая обработка видеоизображений. М., 2009.
2. Федоров А.В., Членов А.И., Лукьянченко А.А. и др. Системы и технические средства раннего обнаружения пожара. М., 2009.
3. Li M., Xu W., Xu K. et. al. Review of fire detection technologies based on video image/ M. Li, Fan, D. Hou //Journal of Theoretical and Applied Information Technology. 2013. Vol. 49. P. 700-707.

АКТУАЛЬНЫЕ ПРОБЛЕМЫ ПРИМЕНЕНИЯ НЕЙРОРЕГУЛЯТОРОВ В СИСТЕМАХ АВТОМАТИЧЕСКОГО УПРАВЛЕНИЯ

А.С. МИХАЙЛОВ

*ФГБОУ ВПО «Костромской государственный технологический университет»
ул. Дзержинского, 17, г. Кострома, 156005, Российская Федерация
amt@kstu.edu.ru*

В статье приводится краткий обзор существующих на сегодняшний день методов нейрорууправления, отмечаются их главные недостатки. В качестве универсального метода нейрорууправления предлагается концепция нейросетевого динамического регулятора состояния.

Ключевые слова: нейрорууправление, система автоматического управления, технологический объект управления, динамический регулятор состояния.

В настоящее время в системах автоматического управления (САУ) технологическими объектами (ТОУ) все более широкое применение находят элементы искусственного интеллекта и в частности искусственные нейронные сети (ИНС). На сегодняшний день уже разработано несколько методов нейрорууправления, классификация которых на основе анализа работ [1, 2] приводится на рис. 1.

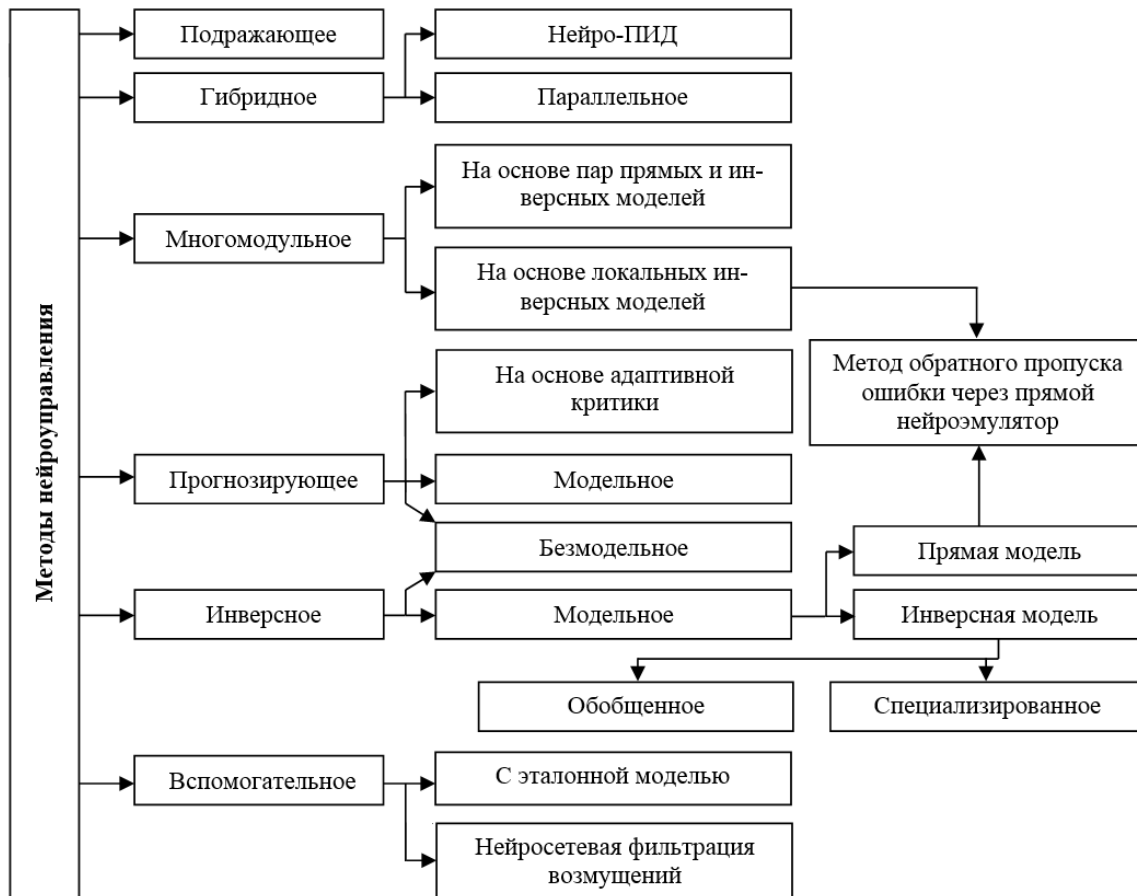


Рис. 1. Классификация методов нейрорууправления

Однако ни один из перечисленных на рис. 1 методов нельзя считать универсальным. В табл. 1 приведены основные недостатки некоторых базовых методов нейрорегулирования. Общие недостатки существующих методов нейрорегулирования также рассмотрены в работе [3].

Табл. 1. Недостатки некоторых базовых методов нейрорегулирования

Метод	Недостатки
Подражающее нейрорегулирование	Для обучения необходим регулятор-учитель Нейрорегулятор в принципе не может обеспечить качество управления лучше, чем регулятор-учитель
Инверсное нейрорегулирование	Сложность реализации инверсных моделей ТОУ Трудности в выборе случайного процесса на входе САУ в процессе идентификации Громоздкость полученной САУ вследствие применения нескольких ИНС
Прогнозирующее нейрорегулирование	Неприменимо в цифровых САУ с малым периодом квантования из-за недостаточного быстродействия алгоритмов оптимизации
Многомодульное нейрорегулирование	Большой объем обучающей выборки Необходимость обучения прямых и инверсных моделей разных модулей Громоздкость полученной САУ вследствие применения нескольких ИНС
Гибридное нейрорегулирование	Сложность методов исследования устойчивости Необходимость в точной математической модели ТОУ

Несмотря на достаточно большое число публикаций по нейрорегулированию и нейрорегуляторам проблемы синтеза регуляторов состояния (РС) на основе ИНС в настоящее время практически не исследуются. А именно данный метод управления, основанный на формировании цепей обратных связей, придающих замкнутой системе заранее выбранное распределение корней, является весьма перспективным. РС по сравнению с другими регуляторами имеют ряд неоспоримых достоинств, среди которых:

- простая структура, поскольку РС представляет собой сумму жестких обратных связей по внутренним координатам ТОУ;
- возможность обеспечить теоретически любой заданный вид переходного процесса;
- высокая робастность системы с РС.

Научная новизна предлагаемой концепции состоит в дополнении стандартного динамического РС модулями нейросетевой идентификации и нейросетевой настройки параметров. При этом модуль нейросетевой идентификации на основе некоторых исходных данных должен формировать структуру и определять параметры модели ТОУ в пространстве состояний, а модуль нейросетевой настройки – определять коэффициенты РС и коэффициенты подстройки наблюдателя состояния.

Список литературы

1. Вороновский Г.К., Махотило К.В., Сергеев С.А. // Проблемы общей энергетики. 2007. №16. С. 54-67.
2. Чернодуб А.Н., Дзюба Д.А. // Проблемы программирования. 2011. №2. С. 79-94.
3. Михайлов А.С., Староверов Б.А. // Вестник ИГЭУ. 2013. №3. С. 64-68.

ИСПОЛЬЗОВАНИЕ ФИЗИЧЕСКИ НЕКЛОНИРУЕМЫХ ФУНКЦИЙ ДЛЯ РЕШЕНИЯ ЗАДАЧИ ГЕНЕРИРОВАНИЯ ДЕЙСТВИТЕЛЬНО СЛУЧАЙНЫХ ЧИСЕЛ И ИДЕНТИФИКАЦИИ

С.С. ЗАЛИВАКО¹, А.А. ИВАНЮК²

¹*Nanyang Technological University
50 Nanyang Avenue, Singapore 639798
zali0001@e.ntu.edu.sg*

²*Белорусский государственный университет информатики и радиоэлектроники
ул. П. Бровки, 6, г. Минск, 220013, Республика Беларусь
ivaniuk@bsuir.by*

Исследована возможность применения физически неклонированной функции на основе статического ОЗУ для решения задачи генерирования последовательности действительно случайных чисел, а также идентификации цифровых устройств. Разработанная схемная реализация в силу своей универсальности может одновременно решать две задачи, используя два различных механизма обработки генерируемых значений.

Ключевые слова: физически неклонированная функция на основе статического ОЗУ, генерирование действительно случайных числовых последовательностей, идентификация цифровых устройств.

В настоящее время цифровые устройства (ЦУ) и их проектные описания нередко подвергаются нелегальному копированию, клонированию и перепроизводству. Одним из способов защиты прав производителей является уникальная идентификация ЦУ и их проектных описаний. Данную задачу производители решают различными способами: физическое нанесение серийных номеров; сохранение идентификатора в регистрах, предназначенных только для чтения; идентификация на уровне функционирования; активное измерение и др. В данной работе предлагается для решения этой задачи использовать физически неклонированные функции (ФНФ) [1]. ФНФ, по Туилсу [1], – это физические системы (устройства), неотъемлемым свойством которых является неклонированность (неповторяемость) некоторых их функций, свойств, характеристик либо параметров.

Для решения задачи идентификации цифровых устройств предложена комбинированная ФНФ [2], которая в зависимости от управляющего сигнала *Challenge* может работать в двух режимах: RO-PUF (Ring Oscillator PUF), SRAM PUF (Static Random Access Memory PUF) [3]. В данной работе мы рассмотрим режим SRAM PUF. В этом режиме значение сигнала *Challenge* = 1, что позволяет эмулировать поведения ячейки памяти, «хранящей» один бит информации. При этом ячейка памяти может постоянно принимать значение логического нуля (единицы) или же изменять свое значение от запуска к запуску. Описанные выше свойства SRAM PUF и будут использованы для получения уникального идентификатора цифрового устройства.

Предлагаемое устройство, построенное на основе ФНФ имеет структуру, представленную на рис. 1. В общем случае устройство состоит из трех структурных блоков: N ячеек ФНФ, схемы сжатия (для генерирования последовательности действительно случайных чисел) или схемы коррекции (для идентификации) и регистра, который хранит либо случайное число, либо идентификатор цифрового устройства.

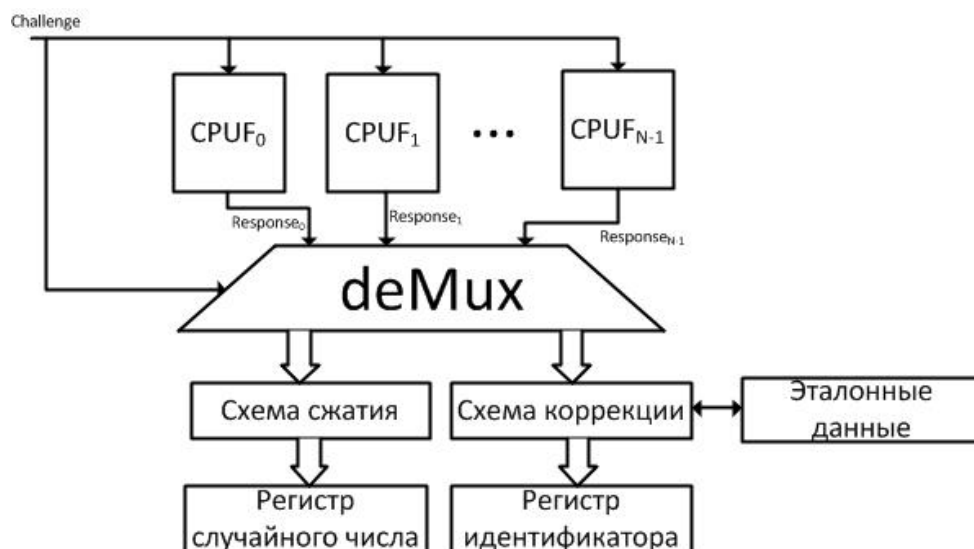


Рис. 1. Структура предлагаемого устройства

В качестве схемы сжатия был использован адаптивный сигнатурный анализатор [4], а для схемы коррекции предлагается использовать коды коррекции ошибок (например, код Хэмминга). Для проверки гипотезы исследования был проведен эксперимент (на двух идентичных ПЛИС Xilinx Spartan 3E-500 FG320, входивших в состав двух плат B_0 и B_1), который состоял в том, что данное устройство включалось и выключалось 10^6 раз и каждый раз на вход подавалось 64-битное значение запроса. В результате чего получался 64-битный отклик, который и являлся уникальным идентификатором цифрового устройства. Поскольку работа ФНФ является нестабильной, то идентификатор для каждого устройства получался по правилу максимального правдоподобия (если в результате n (не менее 100) экспериментов вероятность появления единицы больше чем нуля в определенной ячейке, то за бит идентификатора берется значения единицы, иначе – нуля). Эксперименты показали, что для 64-битных идентификаторов расстояние Хэмминга между устройствами составляет порядка 24.

На основе нестабильности идентификаторов также был построен генератор действительно случайных числовых последовательностей (ГДСЧП). На основе множества идентификаторов с применением адаптивного сигнатурного анализатора были сгенерированы последовательности 8-битных чисел, объемом $6 \cdot 10^7$ бит. Тестирование этих последовательностей пакетами NIST и Diehard показало, что они соответствуют критериям действительной случайности. Также на уровне значимости $\alpha = 0,05$ было подтверждено, что последовательности, сгенерированные на различных платах B^0 и B^1 , не обладают корреляционной зависимостью. Таким образом, две абсолютно противоположные задачи (идентификации и ГДСЧП) могут быть успешно решены с помощью одной и той же схемы комбинированной ФНФ. Метрики расстояний показывают хорошую степень идентификации, а прохождение тестов NIST и Diehard – качество последовательности случайных чисел.

Список литературы

1. *Toils, P.* Security with Noisy Data. London, 2007.
2. *Заливако, С.С., Иванюк, А.А.* // Доклады БГУИР. 2013. № 7 (77). С. 37-43.
3. *Иванюк, А.А.* Проектирование встраиваемых цифровых устройств и систем: монография. Минск, 2012.
4. *Иванюк, А.А., Ярмолик, В.Н.* Проектирование контролепригодных цифровых устройств. Минск, 2006.

ТЕСТОВАЯ СИСТЕМА В АРХИТЕКТУРЕ КЛИЕНТ/СЕРВЕР

А.М. СЕДУН¹, М.Н. САДОВСКАЯ², В.С. ОСКЕРКО³

*Белорусский государственный экономический университет
пр-т Партизанский, 26, г. Минск, 220070, Республика Беларусь
¹sedun@bseu.by, ²sadmanik@rambler.ru, ³oskerkov@mail.ru*

Излагается структура и функциональные возможности тестовой системы, работающей в архитектуре клиент/сервер и используемой в БГЭУ для компьютерного тестирования студентов-заочников с целью контроля их учебы в межсессионный период и допуска к зачетам/экзаменам. Описывается организация и эффективность такого тестирования.

Ключевые слова: компьютерное тестирование, тестовая система, модуль, база тестовых вопросов и заданий.

В Белорусском государственном экономическом университете компьютерное тестирование используется более 15 лет. Поначалу применялись разработки преподавателей простых тестовых программ для эпизодического тестирования по отдельным дисциплинам. Массовый характер компьютерное тестирование приобрело с внедрением в учебный процесс системы контроля знаний, которая была встроенным компонентом визуальной среды разработки интерактивных мультимедийных обучающих приложений Authoware Star for Windows компании Macromedia. Эта система давала возможность промежуточного тестирования по темам дисциплин и итогового тестирования по завершении их изучения. Она устанавливалась на компьютеры, работавшие автономно.

С созданием и развитием в университете локальной компьютерной сети встал вопрос о разработке собственной тестовой системы. Для ее реализации была выбрана архитектура клиент/сервер. Такая тестовая система была разработана в 2002 году и поначалу использовалась для проведения преподавателями в группах промежуточного и итогового тестирования студентов непосредственно на занятиях.

В 2006 году руководством университета было принято решение о проведении компьютерного тестирования студентов-заочников (несколько тысяч человек ежегодно) взамен контрольных работ без участия преподавателей-предметников. Причины тому: часто несамостоятельное выполнение заочниками контрольных работ, желание заставить учиться заочника в межсессионный период и повысить качество обучения.

В связи с этим тестовая система была существенным образом переработана и постоянно совершенствуется в плане расширения функционального наполнения. Особое внимание уделяется вопросам защиты баз тестовых вопросов и заданий по дисциплинам. В настоящее время тестовая система состоит из шести модулей. Модули «Редактор тестов», «Центр управления», «Статистика», «Результаты» устанавливаются на кафедральные компьютеры. «Редактор тестов» дает возможность преподавателям создавать и редактировать базу тестовых вопросов и заданий по дисциплинам на сервере сети. Доступ к этим базам осуществлялся по паролю. В базы допускается включение вопросов и заданий, требующих: выбора одного или нескольких правильных ответов из предложенного перечня; ввода ответа строкой символов с клавиатуры; установки соответствия между элементами двух списков. Формулировки тестовых вопросов и заданий могут содержать объекты разных типов – таблицы, формулы, схемы, графические изображения.

С помощью «Центра управления» преподаватели генерируют тесты – задают их параметры: имена тестов; количество вопросов в тестах; время тестирования; пороги сдачи тестов и др. Имеется возможность редактирования параметров тестов.

«Статистика» позволяет многоаспектный анализ: выявить трудные и легкие вопросы и задания, просмотреть ход обновления баз тестовых вопросов и заданий в указанном периоде, определить количество попыток и процент сдачи тестов и др.

Модуль «Результаты» позволяет лаборантам кафедр сформировать и распечатать для преподавателей перед контрольным мероприятием (зачетом, экзаменом) ведомость сдачи тестов по дисциплинам в учебных группах.

В компьютерных лабораториях, где осуществляется тестирование, на компьютерах студентов устанавливается модуль «Консоль студента», а на компьютере лаборанта – модуль «Лаборант».

Модуль «Лаборант» позволяет: лаборанту зарегистрировать тестируемых по данным зачетной книжки или студенческого билета, отслеживать информацию о количестве сделанных попыток сдачи теста студентами и, если оно не превышает трех допустимых бесплатных, то на компьютерах студентов активировать модуль «Консоль студента». В противном случае студент должен предъявить квитанцию об оплате сдачи теста, и ее номер вводится лаборантом.

В модуле «Консоль студента» студенту генерируется случайным образом набор тестовых вопросов и заданий из базы на сервере по конкретному предмету, заявленному им, и он активируется. Тестируемый может отвечать на вопросы в произвольном порядке. На экране компьютера отображается счетчик времени, оставшегося до завершения тестирования, и при наличии такового можно откорректировать свои ответы.

По окончании тестирования студент имеет возможность ознакомиться с результатом тестирования – количеством правильно отвеченных вопросов в процентном отношении и просмотреть правильно, неправильно и частично правильно отвеченные вопросы.

Компьютерное тестирование потребовало:

- осуществления ряда организационных мероприятий: назначение ответственных преподавателей за разработку и ведение баз тестовых вопросов и заданий, выделение единого по университету времени на их обновление, а также компьютерных лабораторий для тестирования и др.;
- решения методических задач: разработки стандарта СМК БГЭУ «Компьютерное тестирование в учебном процессе», регламентирующего данный процесс; определение тем дисциплин, выносимых на тестирование; параметров тестов и др.;
- пополнения сайта университета методическими рекомендациями по подготовке к тестированию, информацией об именах тестов, компьютерных лабораториях для тестирования и предоставления студентам-заочникам возможности в режиме on-line просматривать результаты сдачи тестов по дисциплинам.

Следствием замены контрольных работ компьютерным тестированием студентов-заочников, без непосредственного участия в его ходе преподавателей-предметников, явилось снижение объема учебной нагрузки в масштабе университета, увеличение доли активной нагрузки преподавателей.

Компьютерное тестирование делает неизбежной усиленную подготовку и систематизацию знаний у студентов, что положительно сказывается на их успеваемости.

Результаты проведенного авторами анализа эффективности компьютерного тестирования в университете дают основание утверждать, что это действительно инновационная форма контроля знаний, способствующая повышению качества обучения, облегчающая труд преподавателей, позволяющая снизить затраты на обучение.

ОБЕСПЕЧЕНИЕ НАДЕЖНОСТИ ПРОГРАММНЫХ СРЕДСТВ ПРИ ИХ МОДИФИКАЦИЯХ

Н.В. УСПЕНСКАЯ¹, В.В. БАХТИЗИН²

¹Белорусский государственный университет информатики и радиоэлектроники
ул. П. Бровки, 6, г. Минск, 220013, Республика Беларусь
nvuspenskaya@gmail.com

²Белорусский государственный университет информатики и радиоэлектроники
ул. П. Бровки, 6, г. Минск, 220013, Республика Беларусь
bww@bsuir.by

Использование программных средств для моделирования сложных систем в последнее время пользуется все большей популярностью. В этой связи большую важность представляют методы обеспечения надежности программных средств для моделирования сложных систем. Одним из средств для вышеуказанной цели является проектирование по контракту.

Ключевые слова: проектирование по контракту, предусловия, модифицированные программные средства.

Для обеспечения надежности программных средств (ПС) для программного моделирования сложных систем предлагается использовать проектирование по контракту (Design by Contract). Данный способ устанавливает отношение между модулями программного средства, однозначно устанавливающий права и обязанности сторон. Такой подход обеспечивает точное определение требований и ответственности для каждого модуля, что повышает тестируемость и диагностируемость программных средств, что в свою очередь обеспечивает их надежность [1].

Основная идея контрактного программирования — это модель взаимодействия элементов ПС. Взаимодействие между модулями ПС происходит по схеме “клиент-поставщик” в соответствии с определенным контрактом. Контракт некоторого объекта включает в себя обязательства клиентского модуля (предусловия), обязательства, присутствующие после выполнения метода (постусловия), обязательства по выполнению конкретных свойств — инвариантов, которые должны выполняться при получении поставщиком сообщения, а также при выходе из метода [2].

Любая открытая функция класса – это не просто абстрактный фрагмент кода, предназначенный для выполнения неопределенной задачи. Каждая функция обладает строгими семантическими свойствами, которые отражают, что делает эта функция, независимо от того, как она это делает. Для четкого выражения задачи, выполняемой конкретной функцией, с ней связывают два утверждения – предусловие и постусловие. Предусловие определяет условия, которые должны выполняться всякий раз перед вызовом функции, а постусловие определяет свойства, гарантируемые после завершения ее выполнения. Предусловие и постусловие определяют контракт функции со всеми ее клиентами [2].

Нарушением контракта является отклонение конкретной реализации от заданной спецификации. При включенном мониторинге нарушение утверждения в период выполнения приводит к генерации исключения.

Выбор уровня мониторинга зависит от разработчика. Предлагается использовать максимальный уровень мониторинга как при отладке ПС, так и в модулях, критичных к высокой надежности выполнения операций. Для модулей, в которых критично время

выполнения операций, предлагается рассчитывать уровень мониторинга в зависимости от производительности приложения.

В случае значительного падения показателей производительности ПС, предлагается не отключать мониторинг полностью и обязательно проверять предусловия. Опыт показывает, что это значительно менее критично к ресурсам нежели проверка постусловий и обеспечивает достаточно высокий уровень надежности по сравнению с полным отсутствием мониторинга.

Как показывает практика проектирования ПС, контрактное программирование повышает уровень повторного использования кода и позволяет однозначно определить причину проблемы в повторно используемом коде.

Существуют слабые и сильные предусловия. Понятия «сильный» и «слабый» пришли из логики. Говорят, что условие $P1$ сильнее, чем $P2$, а $P2$ слабее, чем $P1$, если выполнение условия $P1$ влечет за собой выполнение условия $P2$, но они не эквивалентны, поскольку при выполнении условия $P1$ выполняется и условие $P2$, при этом эти условия не эквивалентны [2].

Любое условие, заведомо равное True, является слабейшим, а условие, заведомо равное False, является сильнейшим из всех возможных описаний. Необходимо избегать данных крайних значений. В случае слабейшего условия, контракт будет выполнен в любом случае, что в корне неприемлемо в соответствии с теорией контрактного проектирования. В случае сильнейшего предусловия контракт будет невыполнен вечно, что навредит функциональной эксплуатации ПС.

В описании метода проектирования по контракту не описаны подходы для определения степени «силы» для предусловия. Предлагается реализовывать предусловия выполнения контракта максимально сильными, но не доходящими до крайних значений. Т.е. условия выполнения контракта должны предусматривать проверку корректности всех внешних данных, используемых при выполнении метода. Значения для проверки должны быть минимально удовлетворяющие контракту. Это позволит обеспечить высокую надежность и при этом не повредит работоспособности ПС.

Определить степень того, насколько предусловие сильно и подходит для конкретного случая проверки контракта предлагается исходя из того, что успешность выполнение метода не должно зависеть от состояния данных после проверки предусловия. При выполнении данного условия ожидается, что ответственность за сбой в работе лежит на модуле, который выполнил последнюю операцию, что вносит ясность при расследовании причин неправильной работы ПС.

Для обеспечения целостности данных рекомендуется выполнять проверку контракта методами, реализованными в классе-источнике. Опыт показывает, что стратегия, при которой каждый модуль ПС несет ответственность только за свои действия, приносит результаты при эксплуатации, сопровождении и доработке модифицированных ПС.

Как показывает практика проектирования ПС, контрактное программирование повышает уровень повторного использования кода и позволяет однозначно определить причину проблемы в повторно используемом коде.

Список литературы

1. *Мейер, Б.* Объектно-ориентированное конструирование программных систем. М.: Русская редакция, 2005.
2. *Meyer, B.* Touch of Class. Learning to Program Well with Objects and Contracts. London: 2009.

БАЗОВЫЕ ПРИМИТИВЫ СХЕМНОЙ ОБФУСКАЦИИ ЦИФРОВЫХ УСТРОЙСТВ

В.В. СЕРГЕЙЧИК¹, А.А. ИВАНЮК²

*Белорусский государственный университет информатики и радиоэлектроники
ул. П. Бровки, 6, г. Минск, 220013, Республика Беларусь
vovasq@mail.ru¹, ivaniuk@bsuir.by²*

Рассматриваются особенности обфускации высокоуровневых описаний цифровых устройств. Дается краткий обзор разновидностей обфускации, изучаются их достоинства и недостатки. Рассматриваются методы схемной обфускации. Приводятся базовые примитивы схемной обфускации, исследуется их применение.

Ключевые слова: обфускация VHDL, лексическая обфускация, схемная обфускация.

Введение

В настоящее время ущерб от несанкционированного производства и использования цифровых устройств составляет около 1 миллиарда долларов в день. Развиваются и другие виды угроз: атаки на аппаратные реализации криптографических алгоритмов (т.н. Side-Channel Attacks), вредоносные изменения схем (аппаратные трояны). В связи с этим решающее значение приобретает разработка методов защиты от подобных угроз. Обфускация – один из таких подходов. Она служит для запутывания понимания структуры и функционирования схемы с целью защиты от обратного проектирования [1]. Кроме того, обфускация применяется для сокрытия авторских водяных знаков и пользовательских отпечатков пальцев.

Схемная обфускация

В случае языка VHDL выделяют две разновидности методов обфускации: лексические и функциональные (схемные). Лексическая обфускация в случае языка VHDL затрагивает лишь уровень исходного описания. Отсюда происходит её главный недостаток: результат синтеза (схема) до и после обфускации остаётся неизменным. Именно поэтому простейшим способом атаки на лексические методы является логический (RTL) синтез. Суть схемной обфускации в получении более сложной для понимания схемы, имеющей эквивалентную функциональность:

$$\begin{aligned} V; V^* = O(V); DD(V) = S; DD(V^*) = S^*; S \neq S^*; Func(S) \equiv Func(S^*); \\ C(\{V^*, S^*\}) > C(\{V, S\}), \end{aligned} \quad (1)$$

где V – исходное HDL-описание, V^* – обфусцированное описание, O – результат обфускации, DD – результат синтеза описания, S – схема, $Func$ – функциональность схемы, C – сложность.

Интересным способом схемной обфускации является внедрение генераторов констант (ГК). Они представляют собой разновидность непрозрачных предикатов, значения которых известны на этапе обфускации, но должны быть вычислены при анализе. Способ предполагает замену выводов «0» и «1» схемами (примитивами), генерирующими соответствующие логические значения постоянно:

$$V_{\{0,1\}}; DD(V_{\{0,1\}}) = S_{0,1} \notin \{V_{DD}, GND\}; Func\{S_{0,1}\} \equiv Func\{V_{DD}, GND\}, \quad (2)$$

где $V_{\{0,1\}}$ – исходное описание примитива; V_{DD} , GND – сигнальные источники логической «1» и «0» соответственно.

Сложность непрозрачного предиката будет определяться сложностью анализа схемного примитива.

Важно добиться того, чтобы схема генератора не оказалась распознана и минимизирована средством синтеза. Различные схемы на основе только комбинационной логики (переключательных функций) не позволили достичь такого результата. Перспективными и эффективными с точки зрения числа используемых ресурсов выглядят последовательностные схемы и схемы, сочетающие последовательностную и комбинационную логику. В качестве дополнительных ухищрений может рассматриваться использование сигналов с хорошо определённым поведением (сигналы сброса, синхронизации и т.д.) на входах ГК.

Примеры ГК приведены на рис. 1. Они синтезированы с помощью XST I.24 Release 8.1i, синтезатор не смог распознать константы и минимизировать эти схемы. Схема *a* обладает двумя недостатками: синтезатор выдаёт предупреждение о комбинационном цикле, что даёт атакующему повод для анализа; если при включении на вход *src* подана «1», то до изменения значения входного сигнала на выходе *q* возможна «1», после изменения всегда будет «0». Устранить второй недостаток можно, подавая на вход сигнал системного сброса. Схема *z*, сочетающая последовательностную и комбинационную логику, лишена указанных недостатков, но требует больше аппаратных ресурсов.

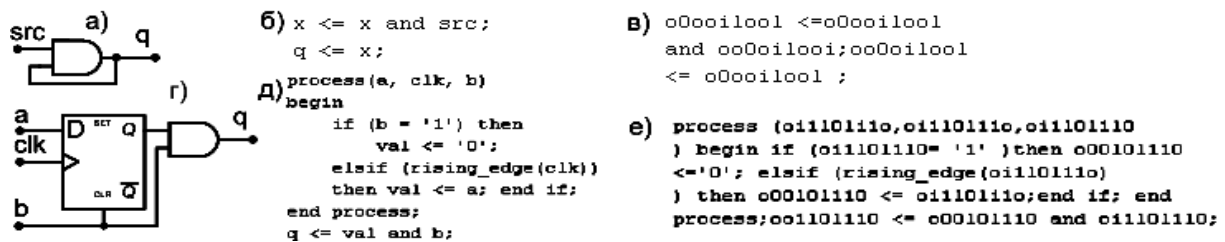


Рис. 1. Генераторы констант: *a*, *z* – схемы примитивов; *б*, *д* – исходные VHDL-описания; *в*, *е* – результаты лексической обфускации

Достоинством схемной обфускации является усложнение для понимания не только высокоуровневого описания, но и результата синтеза. Приведённые примеры демонстрируют недостатки схемной обфускации: в некоторых случаях возможно увеличение аппаратных затрат и снижение скорости работы схемы.

Процесс внедрения примитивов осуществляется на уровне HDL-описаний.

Заключение

Использование схемной обфускации позволит скрыть константы – ценные источники информации для атакующего. Исследование зависимости входов примитивов от сигналов исходной схемы приведёт к дополнительному увеличению времени и сложности анализа обфусцированной схемы. Для увеличения защиты на всех уровнях абстракции имеет смысл использовать лексическую и функциональную обфускацию совместно.

Список литературы

1. *Chakraborty R.S.*, Hardware Security Through Design Obfuscation: Dis., Ph.D. Cleveland, 2010.

ПРИМЕНЕНИЕ СТАТИЧЕСКОГО АНАЛИЗА ИСХОДНЫХ КОДОВ В ОЦЕНКЕ НАДЁЖНОСТИ WEB-ПРИЛОЖЕНИЙ

Д.Е. ОНОШКО¹, В.В. БАХТИЗИН²

Белорусский государственный университет информатики и радиоэлектроники

ул. П. Бровки, 6, г. Минск, 220013, Республика Беларусь

¹dimonsoft@sa-sec.org, ²bww@bsuir.by

Исследованы существующие подходы к оценке надёжности и безопасности web-приложений с точки зрения их устойчивости к SQL-инъекциям. Предложен способ применения статического анализа исходных кодов web-приложений для обнаружения в них уязвимостей к SQL-инъекциям. Предложены числовые характеристики, основанные на данных, собранных в результате статического анализа, которые могут использоваться для оценки надёжности web-приложений.

Ключевые слова: надёжность, безопасность, web-приложение, SQL-инъекция, статический анализ исходных кодов, фильтрация данных.

Одной из основных тенденций в области разработки программных средств (ПС) в настоящее время является тенденция к переходу от классических desktop-приложений к web-приложениям. Между тем, размещение таких приложений на общедоступных серверах повышает требования к качеству этих приложений и, в частности, к корректности обработки поступающих от пользователей данных.

С точки зрения выполняемых функций непосредственно web-приложение представляет собой программный код, обеспечивающий преобразование запросов пользователей, как правило, по протоколу HTTP, в запросы к базе данных на языке запросов, в роли которого чаще всего выступает SQL. Анализ угроз безопасности, проведённый в рамках Открытого проекта обеспечения безопасности web-приложений (OWASP) в 2013 году показал, что наиболее распространённой угрозой для различных типов приложений и в том числе web-приложений являются SQL-инъекции [1].

Причиной уязвимости web-приложения к SQL-инъекции является наличие в нём ошибок, вследствие которых полученные от пользователя данные полностью или частично подставляются в текст запроса к системе управления базами данных (СУБД) без необходимой фильтрации. Это может быть использовано атакующим для внедрения таких специально выбранных последовательностей символов, что сгенерированный web-приложением запрос будет содержать необходимые атакующему команды SQL. Таким образом, недостаточная обработка (фильтрация) пользовательских данных открывает возможность выполнения произвольных действий на сервере приложений, а впоследствии и на компьютерах других пользователей web-приложения.

Для упрощения разработки web-приложений, не уязвимых к SQL-инъекциям, были предложены такие технические решения, как хранимые процедуры и подготовленные выражения. Между тем, они, хотя и имеют высокую эффективность, не являются достаточными, поскольку лишь обеспечивают безопасный механизм взаимодействия с СУБД, тогда как разработчик по тем или иным причинам может отказаться от их применения.

Несмотря на высокую популярность методов обнаружения ошибок посредством динамического анализа поведения web-приложения в условиях эксплуатации, их реальная эффективность остаётся крайне низкой, поскольку отсутствие проблем с точки зрения систем или компонентов, реализующих данные методы, не означает фактического

отсутствия уязвимостей в web-приложении. Единственным способом гарантированного обнаружения всех подобных ошибок, способных открыть возможность проведения SQL-инъекции, остаётся анализ исходных кодов, который является рутинной процедурой, трудоёмкость которой очевидно выше трудоёмкости разработки анализируемого web-приложения. Поэтому целесообразна её автоматизация с помощью ПС контроля качества кода, ориентированных на обнаружение потенциальных уязвимостей.

Для разработки подобных ПС предлагается подход, основанный на анализе потоков данных. Для этого анализируемое web-приложение рассматривается как множество $P = \{P_1, P_2, \dots, P_N\}$ процедур, вызывающих друг друга с некоторыми параметрами. Формальным параметрам процедур назначаются оценки, в простейшей реализации имеющие бинарный характер: «опасный» — данные, соответствующие параметру, будут корректно обработаны даже при отсутствии должной фильтрации, «безопасный» — данные должны пройти предварительную фильтрацию. Аналогичным образом назначаются оценки для возвращаемых процедурой значений: «опасный» — возвращаемые данные должны пройти фильтрацию перед подстановкой в текст запроса, «безопасный» — возвращаемые данные могут подставляться в запрос без дополнительной обработки.

Первоначально ПС известны оценки только для операторов языка программирования (также рассматриваются как процедуры) и некоторых стандартных процедур для взаимодействия с СУБД. Пусть на i -м шаге известны оценки для параметров и возвращаемых значений процедур $P'(i) = \{P_1, P_2, \dots, P_{C(i)}\}$, где $C(i)$ — количество таких процедур на i -м шаге, а процедурой $P_{C(i)+1}$ используются только процедуры из $P'(i)$. Тогда, анализируя операторы $P_{C(i)+1}$, можно получить необходимые оценки для её параметров и возвращаемых значений: оценка фактического параметра совпадает с оценкой формального параметра. Последней анализируемой процедурой является процедура P_N , представляющая основную программу (главный блок `begin...end`, функцию `main()` и т.п.). Вычисленные оценки формальных параметров P_N (им соответствуют поступающие от пользователя данные) должны иметь значение «опасный». Параметры, для которых это не выполняется, являются потенциально уязвимыми. Анализируя путь, по которому была получена оценка, можно выявить причину возникновения уязвимости и предложить способы её устранения.

Пусть A — количество потенциально уязвимых параметров P_N , B — их общее количество. Тогда величина $X = 1 - \frac{A}{B}$ может использоваться в качестве оценки, характеризующей устойчивость web-приложения к SQL-инъекциям. Данная характеристика может отслеживаться на протяжении процесс разработки web-приложения с момента появления прототипа и использоваться для принятия проектных решений.

Список литературы

1. OWASP Top 10-2013. The Ten Most Critical Web Application Security Risks [Электронный ресурс]. — Режим доступа: <http://owasptop10.googlecode.com/files/OWASP%20Top%2010%20-%202013.pdf>. — Дата доступа: 31.10.2013.

ВЫБОР ПАРАМЕТРОВ ПРОЕКТИРУЕМОЙ СИСТЕМЫ ФАЗОВОЙ СИНХРОНИЗАЦИИ

Д.Л. ШИЛИН, В.В. ПУЧИНЕЦ¹, Л.Ю. ШИЛИН

*Белорусский государственный университет информатики и радиоэлектроники
ул. П. Бровки, 6, г. Минск, 220013, Республика Беларусь
¹puchinets.viktor@yandex.ru*

Системы фазовой синхронизации (СФС) широко применяются в радиотехнике, телекоммуникациях и компьютерной технике, что обусловлено их высокой точностью, скоростью, простотой технической реализации, работой в широком диапазоне частот.

Ключевые слова: система фазовой синхронизации, области устойчивости, области качества, критерий выбора.

Ранее авторами предложен метод проектирования СФС [1], основывающийся на построении многомерных областей устойчивости и качества. При этом появляется необходимость выбора параметров проектируемого устройства, при которых рабочие характеристики СФС соответствуют заданным требованиям. Однако вследствие большого количества возможных значений параметров передаточных функций и наличия нескольких факторов влияющих на их выбор, данный процесс весьма затруднителен.

Для упрощения процедуры выбора необходимых значений используются различные критерии определения наилучших значений параметров в зависимости от требований, предъявляемых к характеристикам системы [2].

Авторами предлагается критерий выбора наилучших значений параметров передаточных функций в зависимости от взаимного соотношения требований, предъявляемых к быстродействию и уровню шумов СФС:

$$\Theta = K_1 \frac{t_{\text{пп}}}{t_{\text{пп min}}} - K_2 \frac{SNR}{SNR_{\text{max}}} \quad (1)$$

где $t_{\text{пп}}$ – время переходного процесса;
 SNR – отношение сигнал/шум;
 K_1, K_2 – весовые коэффициенты быстродействия и уровня шума соответственно.

Согласно данному критерию наилучшими являются такие параметры СФС, при которых Θ принимает минимально возможное значение.

На основании ранее полученных областей качества по времени переходных процессов и уровню шумов, по предложенному критерию строятся области значений параметра Θ . По полученным областям выбираются наилучшие значения параметров передаточных функций проектируемой системы.

В зависимости от области применения системы, при ее проектировании различное значение придается быстродействию и уровню шумов. Наиболее часто встречаются три варианта устройств фазовой синхронизации:

1. Быстродействующие СФС с приемлемым уровнем шумов. В этом случае весовые коэффициенты принимают следующие значения:

– $K_1 = 2 \div 10$, так как основным качеством системы является высокое быстродействие;

– $K_2 = 1$, так как полное пренебрежение отношением сигнал/шум может привести к невозможности распознавания полезного сигнала на выходе устройства.

2. Малошумящие СФС с низким быстродействием. Для данных систем весовые коэффициенты принимают значения:

– $K_1 = 1$, так как необходимо обеспечить приемлемое быстродействие СФС.

– $K_2 = 2 \div 10$, так как определяющим параметром системы является уровень шумов.

3. СФС обладающие высоким быстродействием и низким уровнем шумов. Данные устройства наиболее распространены. Так как в этом случае быстродействие и уровень шумов оказывают равное влияние на выбор параметров системы, следовательно, весовые коэффициенты принимают одинаковые значения: $K_1 = 1$, $K_2 = 1$.

В предложенной методике проектирования СФС, описанной в [1], при построении областей устойчивости областей качества использовались линеаризованные модели СФС, допускались некоторые приближения и упрощения при математических преобразованиях выражений, необходимых для определения устойчивости и оценки быстродействия и уровня шумов, для уменьшения трудоемкости данных операций. Вследствие этого границы построенных областей могут быть искажены, т. е. быть больше реальных областей с полученными параметрами. Поэтому перед разработкой электрической схемы устройства необходимо проверить соответствие системы с параметрами из выбранной области, заданным требованиям цифровым моделированием на имитационной модели.

При моделировании могут использоваться различные пакеты программ. Единственным условием является возможность работы с моделью СФС, выбранной для проверки характеристик спроектированного устройства [3, 4].

После получения характеристик спроектированной СФС выполняется проверка, соответствуют ли полученные параметры работы системы требованиям технического. В случае выполнения данного условия осуществляется переход к этапу представления полученных передаточных функций системы электрическими цепями и звеньями.

Если обнаружено не соответствие характеристик проектируемой СФС требованиям технического задания, необходимо выбрать другие значения параметров передаточных функций из нужной области и повторить процесс моделирования.

Список литературы

1. Шилин Д.Л., Пучинец В.В., Шилин Л.Ю. Проектирование систем фазовой синхронизации. Информационные технологии и системы 2011 (ИТС 2011): материалы международной научной конференции. Стр. 21-22. БГУИР, Минск, Беларусь, 26 октября 2011 г.

2. Романов С.К., Тихомиров Н.М., Ленъшин А.В. Системы импульсно-фазовой автоподстройки в устройствах синтеза и стабилизации частот. – М.: Радио и связь, 2010. – 328 с.

3. *Giovanni Bianchi*. Phase Locked Loop Synthesizer Simulation. – McGraw-Hill, 2005.

4. *Dean Banerjee*. PLL Performance, Simulation, and Design. Fourth Edition. – Dog Ear Publishing. LLC, 2006.

ПРИНЦИПЫ МАТЕМАТИЧЕСКОГО ОПИСАНИЯ СИСТЕМ ФАЗОВОЙ СИНХРОНИЗАЦИИ

М.П. БАТУРА, А.П. КУЗНЕЦОВ, Л.Ю. ШИЛИН, Д.П. КУКИН, Д.Л. ШИЛИН

*Белорусский государственный университет информатики и радиоэлектроники
ул. П. Бровки, 6, г. Минск, 220013, Республика Беларусь
goroch@bsuir.by*

Приведены результаты исследования различных вариантов реализации аппаратных и программных систем фазовой синхронизации, предложены возможные варианты построения обобщенных математических моделей многокольцевых систем, а также устройств с циклическим прерыванием режима автоподстройки, предложен один из возможных алгоритмов функционирования программной системы фазовой синхронизации.

Ключевые слова: система фазовой синхронизации, многокольцевые, системы с циклическим прерыванием режима автоподстройки, программные системы.

Развитие современной радиотехники, устройств связи и управления, радиолокации и навигации, радио и информационно-измерительных комплексов невозможно представить без всестороннего применения систем фазовой синхронизации (СФС). В современной технике все большее применение находят СФС с элементами дискретизации [1], что связано с общими тенденциями развития радиоэлектроники, совершенствованием элементной базы микроэлектроники, что позволяет использовать дискретные устройства в высокочастотных системах. Многокольцевые импульсные системы фазовой синхронизации и многокольцевые импульсно-цифровые системы частотно-фазовой автоподстройки получили большую популярность в технике частотного синтеза. Добавление дополнительных связей между кольцами в многокольцевых СФС позволяет повышать быстродействие, расширять области устойчивости и синхронизма. На современном этапе развития науки и техники все большее внимание привлекает идея разработки СФС с циклическим прерыванием режима автоподстройки [2]. Такие устройства образуют класс цилиндрических дискретных систем с разрывным временем.

Многокольцевые СФС (МКСФС) находят все большее применение в современных синтезаторах частоты [3]. Введение дополнительных колец и перекрестных связей между ними позволяет улучшить качественные параметры устройства, прежде всего точность оценки отслеживаемого параметра по сравнению с однокольцевыми, а также повысить быстродействие, расширить область устойчивости и диапазон синтезируемых частот. МКСФС образуют класс устройств тороидального типа [4], особенностью которых является наличие нескольких периодов дискретизации. Получена следующая математическая модель МКСФС:

$$\begin{cases} U_{\phi d1 m+1} = U_{\phi d1 m} + \alpha \cdot [\{\theta\}_1 - F1(U_{\phi d1 m}) - K_{c1} \cdot \beta \cdot F2(U_{\phi d2 m})] \\ U_{\phi d2 m+1} = U_{\phi d2 m} + \beta \cdot [\{\theta\}_2 - F2(U_{\phi d2 m}) - K_{c2} \cdot \alpha \cdot F1(U_{\phi d1 m})] \end{cases}$$

где, $\alpha = S1 \cdot \frac{k_{1m}}{k_{\varphi}}$, $\beta = S2 \cdot \frac{k_{2m}}{k_{\varphi}}$, k_{1m} и k_{2m} - максимальные кодовые последовательности на выходах первого и второго детекторов, k_{φ} - код соответствующий разности фаз равной π , θ_1 и θ_2 - нормированные частотные расстройки первого и второго колец.

Дискретные системы СФС с циклическим прерыванием режима автоподстройки (ДСФСЦП) находят все более широкое применение в радиотехнике из-за особенностей,

связанных с применением режима прерывания, позволяющих одновременно генерировать несколько высокостабильных частот, а также строить устройства с пониженным энергопотреблением [5]. Наиболее часто подобные системы используются в устройствах, допускающих паузы в передаваемом сообщении. Для режима подстройки получена следующая система уравнений, описывающих поведение системы:

$$\begin{cases} U_{OC, t+1} = U_{OC, t} - \alpha \cdot F(U_{OC, t}) + U_{m, t}[z] \\ U_{m, t+1}[z] = d \cdot U_{m, t}[z] - \beta \cdot F(U_{OC, t}) + g' \\ 0 \leq t < k. \end{cases}$$

В тоже время в режиме модуляции уравнения будут иметь следующий вид:

$$\begin{cases} U_{OC, t+1} = U_{OC, t} - \alpha \cdot F(U_{OC, k-1}) + U_{m, t}[z] \\ U_{m, t+1}[z] = d \cdot U_{m, t}[z] - \beta \cdot F(U_{OC, k-1}) + g' \\ k \leq t < k+l. \end{cases}$$

Где k – длительность режима подстройки частоты, l – длительность режима модуляции, m – номер итерации, которые в данном случае обозначают циклы функционирования устройства, $U_{m, t}[z]$ – частотная расстройка, возникающая из-за дискретности системы, g – обобщенный параметр.

В последние годы все большей популярностью пользуется идея реализации СФС средствами программного обеспечения [6]. Очевидно, что данный вариант реализации является актуальным для случая, когда необходимые алгоритмы функционирования выполняются достаточно быстро на аппаратной платформе, используемой для запуска программы. Программные СФС (ПСФС) обеспечивают значительно больший спектр возможностей, чем аппаратные варианты реализации, благодаря тому, что программными методами можно имитировать как аналоговые, так и цифровые системы, а также реализовывать функции и алгоритмы недоступные для аппаратных устройств [7]. Любую аналоговую, цифровую, или аналогово-цифровую СФС можно реализовать программными методами, поэтому количество разнообразных вариантов реализации и алгоритмов функционирования ПСФС чрезвычайно велико.

Список литературы

1. Батура, М.П. Дискретные системы с фазовым управлением / М. П. Батура ; под общ. ред. А.П. Кузнецова; Бел. гос. ун-т информатики и радиоэлектроники. – Минск : Ин-т техн. кибернетики, 2002. – 173, [1] с.
2. Giovanni Bianchi. Phase-Locked Loop Synthesizer Simulation. // McGraw-Hill Professional, 2005. – 304 с.
3. Паушкина Т.К. Динамические свойства синтезатора частот на основе двух взаимосвязанных колец ФАПЧ // Теоретическая электротехника. Республ. межвед. научн. техн. сб. Львов.: Львовский гос. ун-т. – 1989. – Вып. 47. – С. 122–128.
4. Федосова Т.С. Анализ систем фазовой синхронизации с двумя периодическими нелинейностями // Радиотехника. 1986. № 6. – С.46–48.
5. Казаков Л.Н. "Разработка и исследование быстродействующих широкополосных синтезаторов частоты" Дис. канд. тех. наук./ Моск. инст-т радиотехн. электрон. и автомат. – М.: 1988. – 172 с.
6. Keliu Shu. CMOS PLL Synthesizers: Analysis and Design. // Springer, 2005 – 216 с.
7. Roland E. Best. Phase-Locked Loops: Design, Simulation, and Applications // McGraw-Hill (Tx). 2007 – 408 с.

СОЗДАНИЕ УЧЕБНО-НАУЧНОГО ПОРТАЛА ЯДЕРНЫХ ЗНАНИЙ

И.Я. ДУБОВСКАЯ¹, Т.А. САВИЦКАЯ¹,
А.С. ЛОБКО², С.Н. СЫТОВА², С.В. ЧЕРЕПИЦА²

¹Белорусский государственный университет
пр-т Независимости, 4, г. Минск, 220030, Республика Беларусь
dubovskaya51@mail.ru, savitskayaTA@bsu.by

²Институт ядерных проблем Белгосуниверситета
ул. Бобруйская, 11, г. Минск, 220030, Республика Беларусь
lobko@inp.bsu.by, sytova@inp.bsu.by, svcharapitsa@tut.by

Предлагается концепция и принципы создания учебно-научного портала ядерных знаний.

Ключевые слова: портал, система электронного документооборота, ядерные знания.

С начала 2000-х годов Международное агентство по атомной энергии (МАГАТЭ) уделяет огромное внимание управлению ядерными знаниями, в том числе разработке специализированных порталов ядерных знаний на различных уровнях. В результате этой работы в мире постепенно складывается единое информационное пространство в области ядерных знаний. Развитие атомной отрасли Беларуси ставит задачи подготовки квалифицированных специалистов в соответствующих областях науки и технологий, в том числе ядерной, нейтронной и реакторной физики, физики ионизирующих излучений, применении ядерно-физических методов в различных областях науки и техники, ядерной медицины и т. д. В связи с этим актуальным является создание в стране портала управления ядерными знаниями. Первые предложения по созданию учебно-научного портала ядерных знаний, который бы в перспективе мог перерасти в общенациональный портал, содержатся в [1, 2].

Социально-экономическая значимость и необходимость создания такого портала заключается в обеспечении быстрого доступа к необходимой информации, накоплении, сохранении и преумножении знаний на уровне, обеспечивающем безопасное, устойчивое и эффективное развитие ядерной энергетики и промышленности страны, а также популяризации ядерных знаний с целью привлечения в эту область самых способных молодых людей и создания позитивного имиджа ядерной энергетики.

В будущем на основе предлагаемого проекта возможно изменение его тематики и создание учебно-научных порталов с системой дистанционного обучения различных профилей. Новизна предлагаемой разработки заключается в разработке электронного учебно-научного портала ядерных знаний с учетом специфики и условий Беларуси, формирования собственной базы ядерных знаний и системы дистанционного обучения на основе свободного программного обеспечения. Готового свободного программного обеспечения, реализующего заявленные цели и задачи, в настоящее время нет.

Портал будет являться системой, объединяющей информационные ресурсы (приложения, базы данных и др.) и позволяющей создателям и потребителям знаний взаимодействовать друг с другом с предоставлением им единого защищенного доступа к информации и виртуальных каналов коммуникаций для совместной работы над документами из географически разнесенных мест через единый web-интерфейс в режиме коллективной работы со строгой персонализацией (разграничением права доступа к ресурсам: данным, сервисам, приложениям, документам).

Таким образом, портал ядерных знаний будет одновременно являться 1) вертикальным порталом (порталом-нишей), имеющим тематическую направленность и ориентированным на полный охват заявленной тематики; 2) публичным порталом, открытым для широкой интернет-публики, интересующейся ядерной тематикой и 3) корпоративным порталом совместной работы. Основное отличие такого портала от обычного веб-сайта – наличие интерактивных сервисов (почта, новости, форумы, инструменты для совместной и индивидуальной работы пользователей и т.д.), в том числе средств дистанционного обучения. Важные подсистемы портала: интерфейс с удобным и быстрым поиском и навигацией; система электронного документооборота с соблюдением правил и стандартов разработки и сопровождения документов, средствами интеллектуального поиска, категоризации информации; системы управления проектами, учебными материалами и курсами с системой дистанционного обучения студентов и возможностью создания и совершенствования курсов; электронная библиотека; форумы по основным направлениям деятельности; каналы новостей и другие важные приложения, интегрированные в портал.

Система дистанционного обучения, которая будет доступна в рамках on-line работы портала, должна содержать видеолекции и анимированные уроки (возможно, разбитые на короткие модули), электронные учебники, интерактивные тесты и др. материалы, разработанные лучшими преподавателями республики. Такие системы активно развиваются во всем мире последние 20 лет. Система дистанционного обучения в рамках портала ядерных знаний будет способствовать повышению престижа и качества образования в области ядерной науки и технологий. В совокупности материалы электронной библиотеки, учебные курсы, базы данных, электронные документы ресурсов (фото, видео и т.д.), другой контент портала составят базу ядерных знаний.

Основными этапами создания портала являются следующие шесть этапов:

- 1) разработка концепции, структуры и таксономии портала ядерных знаний;
- 2) проведение анализа требований и условий функционирования портала;
- 3) разработка информационной модели и архитектуры портала;
- 4) разработка методов и собственных программных средств для реализации портала;
- 5) программная реализация портала с формированием базы знаний и системы дистанционного обучения;
- 6) апробация и верификация портала, наполнение портала знаниями.

Четвертый и пятый этапы включают в себя следующие работы: реализация структуры и таксономии портала в виде базы данных в СУБД Firebird; разработка системы аутентификации пользователя; разработка интерфейса пользователя; сборка всех компонент системы; написание документации; тестирование отдельных компонент системы; системное тестирование и отладка.

Наполнение портала информацией и знаниями, разработка материалов для системы дистанционного обучения является практически бесконечным творческим процессом, к которому должны быть подключены все заинтересованные стороны, особенно преподаватели и научные сотрудники ведущих вузов и научных учреждений страны. Все работы по созданию должны проводиться под контролем и с непосредственным участием МАГАТЭ.

Список литературы

1. *С.Н. Сытова, С.В. Черепица, А.С. Лобко.* // Сб. научных тр. IV конгресса физиков Бе-ларуси (Минск, 24-26 апреля 2013 г.). С. 419–420.
2. *С.Н. Сытова, С.В. Черепица, А.С. Лобко.* // Тр. Межд. конгресса CSIST'2013. Минск, 4-7 ноября 2013 г. С.254–259.

ПОСТРОЕНИЕ ТРЕХМЕРНЫХ МОДЕЛЕЙ ПО ИЗОБРАЖЕНИЯМ С РАЗЛИЧНЫХ КАМЕР

И.Н. ГУБЧИК

*Белорусский государственный университет информатики и радиоэлектроники
ул. П. Бровки, 6, г. Минск, 220013, Республика Беларусь
gubchikin@gmail.com*

В сообщении описана модель реконструкции объектов городских и природных ландшафтов по изображениям, полученным при разных условиях съемки и с использованием различных некалиброванных фотокамер.

Ключевые слова: трехмерная сцена, калибровка камеры, стереоизображение, реконструкция объекта.

В текущее время в связи с бурным развитием картографических сервисов существует задача построения трехмерных моделей объектов как городских, так и природных ландшафтов. Изображения со спутников предоставляют необходимую информацию для этого, однако возможность съемки для многих объектов ограничена. Ставится задача создания программного обеспечения для построения точных трехмерных моделей как для картографических сервисов, так и под специализированные задачи.

Одним из распространенных подходов в построении трехмерных объектов ландшафта является использование 3D-сканеров, однако недостатком данных устройств является их дороговизна. Альтернативой данному методу выступает построение трехмерных объектов по группе снимков.

Предлагаемая модель построения трехмерных объектов акцентирует внимание на предположении, что изображения одного объекта поступают с различной фотоаппаратуры в отличие от предлагаемых методик получения изображений с одной фотокамеры [1] при существовании ограничений по съемке [2]. В модели присутствует модуль анализа полученных снимков, их сортировки по качеству и информационной насыщенности (наличие достаточного количества точек интереса), а также модуль обнаружения брака среди изображений: размытые снимки, некорректное освещение, наличие шумов и т.п. Для построения трехмерной сцены по изображениям с разных камер учитываются условия съемки, искажения различных объективов, место съемки, а также снимаемую часть объекта. Считается, что для построения детализированной трехмерной модели объекта необходимо большее число двумерных снимков, поэтому описываемая модель реконструкции использует пирамидальную математическую модель построения трехмерных сцен [3] для последующего уточнения отдельных деталей сцены.

Поскольку модель работает с изображениями с различных камер, обязательна калибровка каждой отдельной камеры. В качестве калибровочных объектов могут выступать как непосредственные изображения объекта (самокалибровка) [4], так и существующие априорные данные об искажениях, вносимых камерой и объективом. Калибровка по изображениям сцены возможна для корректировки дисторсии объектива, если на изображениях присутствуют объекты известных геометрических форм: линии, окружности. В качестве калибровочной информации выступают известные геометрические размеры объектов на изображении, калибровка выполняется итерационным методом с использованием нескольких изображений, которые содержат объекты потенциально геометрических форм: здания, автомобили, дороги и т.п.. В случае отсутствия

объектов геометрических форм, например, при съемке природных ландшафтов, необходимо заранее знать параметры искажений объектива и камеры, которые могут быть предоставлены пользователем или выбраны из существующей базы данных.

Одной из проблем построения трехмерных сцен является наличие на изображениях динамических объектов. Предлагаемая модель включает в себя метод сегментации динамических объектов при обработке группы снимков. Данный метод определяет соответствие множеству точек, принадлежащих отдельному динамическому объекту, множеству на других снимках.

При построении трехмерной сцены определяются параметры геометрических (пространственных) преобразований (масштаб, вращение, смещение) между двумя скорректированными изображениями, содержащими одинаковый объект ландшафта. Отдельным множествам точек на одном изображении, которые относятся к разным объектам или разным частям объекта, разнесенным в пространстве, определяется однозначное соответствие другому множеству точек на втором изображении. Для каждой пары множества точек на двух снимках определяются параметры геометрических преобразований. По найденной совокупности параметров пространственных преобразований вычисляются параметры общего преобразования между двумя снимками.

Описанная модель построения трехмерных моделей позволяет использовать изображения с различных камер. Применяемые изображения для реконструкции объектов могут быть сделаны разными людьми и разнесены по времени, что подразумевает не целенаправленную съемку ландшафтов, а сбор изображений, предоставляемых случайными пользователями, например, фотографирование исторических зданий посетителями. Модель использует итерационный метод калибровки по предоставляемым изображениям с одной камеры, обходя необходимость использования калибровочных объектов.

Список литературы

1. *M. Vergauwen, L.V. Gool* Web-based 3D Reconstruction // *Machine Vision and Applications*, Volume 17, Issue 6 - December 2006. – pp. 411–426.
2. *A. Koutsoudisa, B. Vidmarb, G. Ioannakisa and others.* Multi-image 3D reconstruction data evaluation // *Journal of cultural heritage* – 17 January 2013 – pp. 7.
3. *N. Indhumadhi, G. PadMavathi* Enhanced Image Fusion Algorithm using Laplacian Pyramid and Spatial frequency based Wavelet Algorithm // *International Journal of Soft Computing and Engineering (IJSCE)*, Volume 1, Issue 5 – November 2011 – pp. 298–302.
4. *P. Armand, A. Kiselev, O. Marcotte and others.* Self Calibration of a Pinhole Camera // *Mathematics-in-Industry Case Studies Journal*, Volume 1 – 2009 – pp. 81–89.

АНАЛИЗ ТРЕБОВАНИЙ НА ОСНОВЕ UML-МОДЕЛЕЙ И ДИНАМИЧЕСКИХ HTML-ПРОТОТИПОВ

Н.Б. КИРЕЕВ

Институт Информационных Технологий

Белорусского государственного университета информатики и радиоэлектроники (ИИТ БГУИР)

ул. Козлова, 28, г. Минск, 220013, Республика Беларусь

info@webmax.by; www.webmax.by

При разработке информационных программных систем функциональные требования являются наиболее сложными с точки зрения их выявления, детализации и управления, но именно они определяют архитектуру и влияют на процесс разработки программы. Предлагаемая итерационная методика позволяет выявлять и детализировать функции системы посредством построения визуальных UML-моделей с размещением последних в среде Axure Pro и генерацией динамических HTML-прототипов. Данная методика может быть использована в рамках стандартного унифицированного процесса разработки программного обеспечения, а также при неопределенных и изменяющихся требованиях к системе в условиях «гибкой» методологии «SCRUM».

Ключевые слова: модель UML, HTML-прототип, диаграмма вариантов использования.

В основе любой программной системы, а также процесса её разработки лежат требования к ПО. Карл Вигерс в [1] предложил их следующую классификацию (рис. 1).

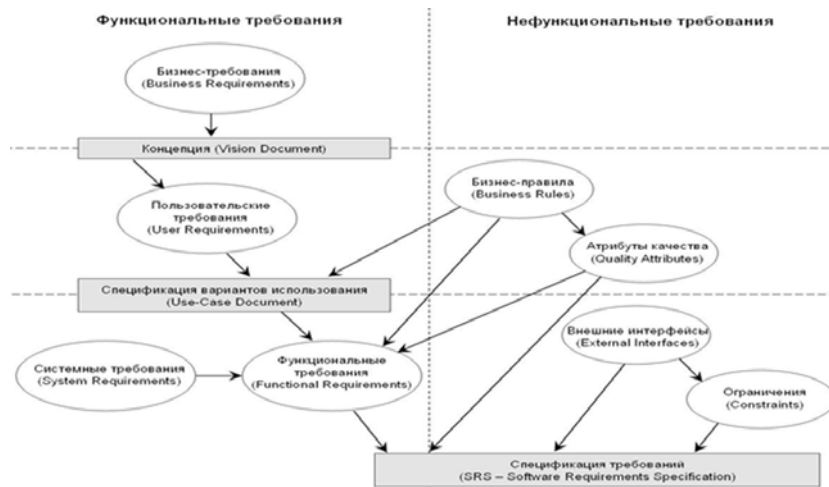


Рис. 1. Классификация требований к ПО, предложенная К. Вигерсом

При этом к группе функциональных требований относятся:

Бизнес-требования (Business Requirements) – определяют высокоуровневые цели организации/клиента (потребителя, заказчика разрабатываемого ПО) или его бизнеса;

Пользовательские требования (User Requirements) – описывают цели/задачи пользователей системы, которые должны достигаться/выполняться пользователями в рамках поддерживаемого бизнес-процесса.

Функциональные требования (Functional Requirements) – определяют функциональность (поведение) программной системы, которая должна обеспечить пользователям исполнение их обязанностей в контексте пользовательских требований.

В приведенной выше классификации, модель вариантов использования создается на основе пользовательских требований, соответственно, на их основе производится разработка архитектуры системы. Данный подход является классическим, но обладает существенным недостатком. Часто такие варианты использования оказываются слишком абстрактными и с трудом могут быть использованы для дальнейшего проектирования архитектуры системы.

Другой подход – создавать модель вариантов использования на основе функциональных сервисов (функций бизнес-логики), которые иницируются через графический интерфейс пользователя (GUI). Проектировать архитектуру, реализующую конкретную функциональность предпочтительнее, чем для более абстрактных пользовательских требований. Однако, до внедрения методик и программ, генерирующих динамические прототипы GUI, сложность такого подхода состояла в том, что этот функционал разрабатывался и тестировался на этапе кодирования. Использование программных продуктов, таких как Axure Pro 6.5 или Prototyper 5.6, позволяет уже на этапе анализа требований детализировать и согласовать с заказчиком функции бизнес-логики, иницируемые GUI разрабатываемой системы, и на их основе проектировать ее архитектуру.

При использовании гибкой модели процесса разработки Scrum прототипы экранных форм могут быть разработаны в самом начале процесса и использоваться для составления product backlog'a – некоего аналога списка требований.

Возможности моделирования функциональности, структур данных и процессов, одновременно с разработкой прототипов экранных форм, позволяют разработчикам выявлять скрытую функциональность, детально прорабатывать и согласовывать с пользователями потоки событий в сценариях взаимодействия с программной системой.

Иерархическая браузерная организационная структура проекта, создаваемого в программном средстве Axure Pro 6.5, позволяет вводить пользовательские виджеты, соответствующие нотации языка визуального моделирования UML и строить диаграммы на отдельных страницах проекта, которые не включаются в HTML-прототип. Кроме того, благодаря простому механизму UML-диаграммы, созданные в других приложениях (Magic Draw UML, Visio, Rational Rose и т.д.), можно легко интегрировать непосредственно в HTML-прототип, например, для пояснения сценария взаимодействия пользователя и программной системы.

Таким образом, визуальное моделирование с одновременным созданием динамических HTML-прототипов, является мощным средством анализа программных систем, позволяющим не только определить функционал, но и детализировать его, а также согласовать с пользователями сценарии взаимодействия.

Список литературы

1. Карл И. Вигерс Разработка требований к программному обеспечению «Русская Редакция». М., 2004.

COMPARISON OF NEHALEM AND IVY BRIDGE TLB ARCHITECTURES

M. ASKARI¹, N.N. IVANOV²

*Belarusian State University of Informatics and Radioelectronics
6, P. Brovki Str., Minsk, Republic of Belarus
¹mahask2004@yahoo.com; ²ivanovnn@gmail.com*

The speed gap between processor and main memory is reduced using a cache. Cache role in multi-core parallel systems is detected here. Cache memories without Translation Lookaside Buffer (TLB), will not show their capabilities and their impacts on performance will be lower. The performance comparison of these buffers in the different architectures helps to discover impact of architectures change on their work. In this paper, two different types of Intel microarchitecture have been compared by miss ration of their TLBs. Nehalem and Ivy-Bridge. For this purpose, the statistical techniques used to compare the measured values by Intel-Vtune2013 on SPEC CPU2000 benchmarks.

Keywords: Nehalem, Ivy-Bridge, Miss Rate, Translation Lookaside Buffer, TLB, Spec CPU2000, Vtune2013.

The developers of the system are trying to increase system performance. To support this goal, the Performance Monitoring Unit (PMU) offers a wide range of events to help investigate the processor components. Observe the counted events require special software and understanding presented numbers need statistical analysis. In addition, it is necessary to run specific and standard applications in order to compare the two systems together. For the calculation of Miss Rates, counted event is divided by the total of events which is called `INST_RETIRED.ANY`. Event containing misses of data TLB (DTLB) is `DTLB_MISSES.ANY` and event containing hits of Second level of TLB (STLB) is `DTLB_LOAD_MISSES.STLB_HIT`. Miss rate in this case is obtained by subtracting the Hit rate of value 1.

SPEC company, has introduced some program that can be used as standard applications to calculating performance and miss ratios of TLB, caches and etc. In this paper, it has been used all 12 benchmarks of CINT2000 package included: BZIP2, CRAFTY, EON, GAP, GCC, GZIP, MCF, PARSER, PERLBMK, TWOLF and VORTEX. Each benchmark is executed 50 times in 3 phases. A total of 1800 times randomly run on each system. All experiments have performed on 2 user mode and a copy of the each benchmark is executed in parallel on each core.

The first system under test with architecture Nehalem is Intel Core i5-460M with 2 cores. It has a two-level TLB for each core. On the first level it is an Instruction TLB (ITLB) and a Data TLB (DTLB). On the second level it is a TLB (STLB) unified Instruction and Data. Posted Virtual Address (VA) by core, according to its type is converted to a Physical Address (PA) in the ITLB or DTLB. If a match not happened in level one, it is required to search in level two of TLB and if the address is not found here, it is need to refer main memory that it has a significant time to update. If at any stage it is happened a match, it will be referred to a hierarchy cache and main memory to obtain data relating to physical address conversion. This system has three levels of caches which are called: L1, L2 and L3. The first level (L1) has two parts for Instruction and Data included: L1I and L1D. Each TLB in each level have some parts related to page size included 4Kbyte size, Large and Huge size. In this table, not paying attention to these cases and other cases including prefetching and nature of bench-

marks for discussion and only it want compare the results of two factors together: DTLB and STLB miss rates. ITLB is not counted by event counter in these experiments.

The second system under test with architecture Ivy-Bridge is Intel Core i5-3317U with 2 cores. Assuming the lack of detail, the two systems will have similar model for study their TLBs. The results have been statistically analyzed. Techniques used for the analysis is T-test method. A null hypothesis assumes that there is no significant relationship between the values of experiments. Then the result of statistical analysis is compared with a value considered. If statistical analysis was performed, have a result less than that assumed value, the null hypothesis is rejected. Usually, the assumed value of 0.05 is chosen. In case of rejection of the null hypothesis, say 95% chance of a significant relationship is between the values.

Table 1 shows the results of the analysis for DTLB and STLB by Regression. “R2” is acronym for Standard deviation, “Ttest” for T-test coefficient and “Bnch” for Benchmark.

Table 1. Result of analysis for DTLB and STLB in 460M and 3317U.

	R2	Ttest	R2	Ttest	R2	Ttest	R2	Ttest
Bnch	BZIP2		CRAFTY		EON		GAP	
460M	0.32718	3.28e-05	0.38111	3.80e-06	0.00274	0.72366	0.04302	0.15719
3317U	0.00698	0.58072	0.20229	0.00134	0.15856	0.00556	0.08589	0.04321
Bnch	GCC		GZIP		MCF		PARSER	
460M	0.35563	1.22e-05	0.00669	0.58875	0.31324	3.53e-05	0.19757	0.00196
3317U	0.14059	0.00940	0.04669	0.13588	0.07384	0.05892	0.81317	9.67e-19
Bnch	PERLBMK		TWOLF		VORTEX		VPR	
460M	0.00222	0.74772	3.05e-05	0.97024	0.13154	0.01324	0.24352	0.00042
3317U	0.11244	0.01981	0.58495	6.10e-10	0.28944	8.02e-05	0.76267	5.71e-16

Considering that, each miss on DTLB is an input for STLB so it is assumed that STLB miss is dependent to miss on DTLB. In this regard, the null hypothesis is stated as STLB miss is not dependent to miss on DTLB. Looking at Table 1, where the value in “Ttest” column is less than 0.05 then null hypotheses in this benchmark is rejected and assumption is acceptable with 95 %.

In relation to the i5-460M, the null hypothesis for EON, GAP, GZIP, PERLBMK and TWOLF is acceptable and is not significant correlation between the results. In relation to the i5-3317U, the null hypothesis for BZIP2, GZIP and MCF is acceptable and is not significant correlation between the results. Therefore, according to the architectural details of these two CPUs that was ignored and prefetching behavior and nature of benchmarks like multithreading that were not considered, so it is probability that the algorithms related to above benchmarks be affected by above behavior. As future work, the real effectiveness of the above behavior on these benchmarks can be studied.

References

1. *Tanenbaum, A. Modern Operating Systems / A. Tanenbaum. // New Jersey: Prentice Hall, 2001.*
2. *Hennessy, J. Computer Architecture / J. Hennessy, D. Patterson // Philadelphia: Elsevier, 2011.*
3. *Kandiraju, B. K. Characterizing the dTLB Behavior of SPEC CPU2000 Benchmarks. / B. K. Kandiraju, A. Sivasubramaniam. // international conference on Measurement and modeling of computer systems. –2002. –Vol.30, p. 129–139.*

КЛАСТЕРНЫЙ АНАЛИЗ СОДЕРЖИМОГО САЙТА ВОПРОСОВ-ОТВЕТОВ STACKEXCHANGE.COM

К.Ю. СЛИСЕНКО¹, С.И. СИРОТКО²

*Белорусский государственный университет информатики и радиоэлектроники
ул. П. Бровки, 6, г. Минск, 220013, Республика Беларусь
¹kslisenko@gmail.com; ²sergeyis@tut.by*

Рассматривается процесс кластеризации данных сайта вопросов и ответов, имеющего большое количество неструктурированного содержимого. В виду предполагаемого большого объёма обрабатываемых данных рассматривается и обосновывается применение специализированных технических средств анализа данных – фреймворков Apache Hadoop и Mahout [1]. Рассматривается применение алгоритма кластеризации K-средних для больших объёмов данных. Дается интерпретация и пояснение полученных результатов.

Ключевые слова: машинное обучение, анализ данных, кластеризация, векторизация, Apache Hadoop, Apache Mahout.

Многочисленные форумы, сайты с вопросами и ответами и иные интернет-ресурсы могут содержать актуальные, свежие и новые знания. Например, популярный ресурс stackexchange.com содержит 8 миллионов вопросов и 14,6 миллионов ответов [2]. Однако проблема заключается в неструктурированности и большой зашумлённости имеющейся информации. Поэтому актуальной задачей является исследование данной информации с целью выявления закономерностей и групп схожих вопросов и ответов, а так же структурирование и последующее визуальное представление общей картины.

Исследуемые данные имеют объём около 15 Гб, поэтому необходимо использовать специализированные технологии и алгоритмы для обработки больших объёмов данных [1]. Для работы был выбран стек технологий на основе фреймворка с открытым исходным кодом Apache Hadoop [3]. Выбор обоснован возможностью масштабирования обработки данных на кластере из множества машин, а так же автоматическим обеспечением отказоустойчивости вычислений [1]. Для осуществления поставленной задачи необходимо прибегнуть к алгоритмам машинного обучения. Для анализа был выбран алгоритм кластеризации K-средних, поскольку он не требует задания ожидаемых групп вопросов, и может быть масштабирован на кластере из множества машин [4]. Данный алгоритм реализован в составе библиотеки Apache Mahout, работающей поверх Hadoop [5].

Первым этапом анализа является подготовка исходных данных в формате XML для последующего применения алгоритма кластеризации. Алгоритм подготовки данных выглядит следующим образом:

1. Разбор XML, извлечение текста вопросов;
2. Обработка текста: разбивка на слова, удаление нетекстовых слов, приведение к нижнему регистру, отсеечение слов по минимальной и максимальной длине, удаление частоупотребимых слов, обработка алгоритмом Портера для приведения слов в начальную форму и удаления окончаний и суффиксов, формирование словаря;
3. Векторизация с учётом частоты слов относительно всех текстовых документов (применение алгоритма TF-IDF).

На вход алгоритму кластеризации приходят сформированные векторы вопросов и ответов. Алгоритм K-средних требует установки количества кластеров, которое зада-

ётся во входном параметре K. Данный алгоритм при выполнении на множестве машин выглядит следующим образом:

1. Выбирается k случайных точек как центры кластеров;
2. На фазе Map каждая точка ассоциируется с ближайшим центром кластера;
3. На фазе Reduce пересчитываются центры кластеров;
4. Производится расчёт величины смещения кластеров.

Для измерения близости между точками и кластерами была использована метрика косинусного расстояния, поскольку она наиболее оптимальна для текстовых данных. Идентификаторы вопросов и ответов сортируются и объединяются с изначальным текстом. На рис. 1 изображены результаты визуализации найденных кластеров.

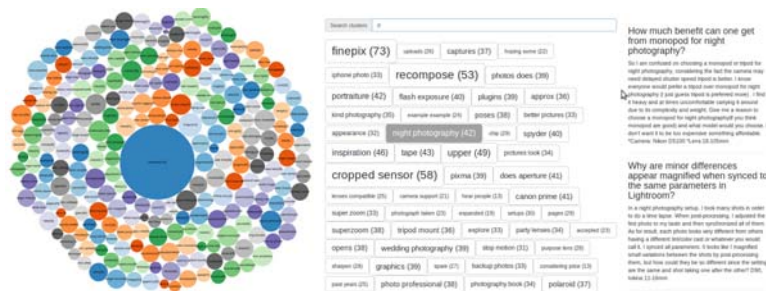


Рис. 1. Результаты визуализации найденных кластеров

На изображении слева каждая окружность соответствует кластеру, величина окружности пропорциональна количеству вопросов. На изображении справа виден список кластеров с возможностью просмотра вопросов, входящих в каждый кластер. Выполненная визуализация помогает понять, насколько широко обсуждается та или иная тематика посетителями сайта.

В рамках исследования был успешно проведен кластерный анализ содержимого web-сайта. При этом было выявлено, что наибольшим образом на качество кластеризации влияет предобработка текстовых данных. В дальнейшем планируется разработать критерии оценки качества кластеризации и на основе них сравнить различные алгоритмов и метрики. Также планируется применить алгоритм для оценки первоначального числа кластеров, чтобы упростить в дальнейшем автоматизацию процесса обработки.

Список литературы

1. Слисенко, К. Ю., Сиротко, С.И., Кириченко А.Ю. // Сб. тез. докл. междунар. на-уч. конф. “Информационные технологии и системы 2013”. Минск, 23 октября 2013 г. С. 326–327.
2. Stackexchange: сайт вопросов и ответов. [Электронный ресурс]. – Режим досту-па: <http://stackexchange.com>. – Дата доступа: 14.01.2014.
3. White, T. Hadoop the definitive guide, Third edition. O’Reilly Media, 2012.
4. Gillick, D. // MapReduce: Distributed Computing for Machine Learning // Berke-ley University, CS262A. 2006.
5. Owen S., Anil R. Mahout in Action. Manning Publications, 2012.

ТЕХНОЛОГИЯ ТЕСТИРОВАНИЯ WEB-ПРИЛОЖЕНИЙ НА ОСНОВЕ ВЕРОЯТНОСТНЫХ СЕТЕЙ

А.А. БЫКОВ, И.И. ПИЛЕЦКИЙ

*Белорусский государственный университет информатики и радиоэлектроники
ул. П. Бровки, 6, г. Минск, 220013, Республика Беларусь
anton.bukov@gmail.com, ipiletski@iba.by*

В докладе предлагается технология автоматизации функционального, регрессионного и приемочного тестирования Web-интерфейсов на основе сетевой модели. В докладе рассматриваются визуальный редактор, генератор кода и эвристические алгоритмы, основанные на статистике тестирования, которые позволяют упростить саму процедуру тестирования, оценить покрытие Web интерфейса тестами, снизить сложность процесса поддержания автоматизированных тестов в актуальном состоянии.

Ключевые слова: вероятностные сети; тестирование; web интерфейсы; автоматизация.

В настоящее время рост сложности интерфейса веб приложений связан с быстрым с ростом сложности самих приложений и распространением технологии асинхронных вызовов. Общепризнанная методология итеративной разработки программного обеспечения RUP подразумевает выполнение полного тестирования на каждой итерации разработки. Процесс тестирования не только нового, но и написанного на предыдущих итерациях разработки, называется регрессионным тестированием. При выполнении такого вида тестирования для упрощения работы тестировщика целесообразно использовать автоматизированные инструменты.

Одним из направлений снижения сложности процесса автоматизации тестирования является разработка визуального языка тестирования, на котором автоматизируется процесс разработки тестов. Примером такого простого, не сложного, высокоуровневого языка тестирования является графический язык программирования системы Lenny [1]. Цель данной системы – снизить сложность разработки и сопровождения автоматизированных регрессионных функциональных тестов, используя визуальный, компонентно-ориентированный язык автоматизации построения тестов. Этот язык состоит из готовых компонентов и представляет собой части сети, узлами которой являются страницы или состояния приложения, а связями – операции над интерфейсом. К примеру, переход по ссылке или нажатие кнопки, ввод текста и др. Для автоматизации тестирования предлагается использовать сеть, которая имеет следующую структуру:

Сеть первого уровня состоит из двух слоев, которые определяют расположение графических элементов управления на web странице. Узлы верхнего уровня это страницы или состояние страницы тестируемого приложения, например страницы аутентификации пользователей. Узлы нижнего уровня представляют собой шаблоны для идентификации графических элементов. Графические элементы, которые встречаются более чем на одной странице, можно перенести в общий для нескольких страниц блок.

Главная задача сети второго уровня – описать алгоритм работы программы в виде простых, связанных между собой правил, а также описать состояния программы и действия над её интерфейсом. Сеть третьего уровня, описывает тесты и дефекты тестируемой программы. Блоки-тесты описывают цель каждого функционального теста и соединен с узлами-страницами, которые необходимо протестировать. Впоследствии, для выполнения одного или нескольких тестов система найдет предварительные шаги

для выполнения тестирования, используя алгоритм поиска пути в графе, предложенный С. Расселом [2].

С каждым тестом связывается приоритет его запуска. Данная характеристика представляет собой вероятность того, что мы не можем предугадать результат выполнения этого теста. К примеру, мы не можем предугадать воспроизведется ли недавно исправленный дефект или правильно ли работает страница код, которой был доработан. Чем выше неопределенность, тем важнее запустить тест и определить наличие или отсутствие проблемы. Приоритет запуска теста может быть задан тестировщиком вручную, получен статистически или вычислен на основе получаемых результатов для данной итерации функционального регрессионного тестирования.

Система автоматизации Lenny является расширением для широко используемой платформы программирования Eclipse. Система позволяет взаимодействовать с основными функциями этой платформы такими как: редактор для графических языков GEF, системы управления версиями файлов SVN и CVS, менеджер проекта и др. Можно выделить 2 режима для работы с тестами для системы Lenny: режим автоматизации тестирования и режим выполнения тестирования. В режиме автоматизации система Lenny запускает окно браузера со встроенным механизмом записи. Для автоматизации тестирования Lenny использует контекстное меню с набором часто используемых операций для добавления операций над интерфейсом в тест. Тестировщик при открытии начальной страницы web-приложения, может нажимая правой клавишей мыши над графическими элементами добавлять проверку их состояния. Также система будет сохранять все действия над этими графическими элементами, такими как щелчек мышью, ввод текста или выбор значения для списков и переключателей. В режиме выполнения тестирования система Lenny запустит окно браузера со встроенным механизмом управления над окном и начнет поэтапно повторять записанные операции и проверять записанные состояния интерфейса.

Результаты работы:

- разработан новый метод автоматизации регрессионного тестирования с использованием вероятностных сетей.
- разработаны алгоритмы поиска пути для выполнения тестирования согласно различных критериев тестирования.
- разработан инструмент (плагин для платформы Eclipse), поддерживающий автоматизацию регрессионного тестирования согласно описанного метода.

Список литературы

1. *Bykau A. A.*, Probabilistic Networks as a Means of Testing WebBased Applications, SYRCoSE 2013, 30-31, may 2013, Kazan, Russia, pp 58-61
2. *S. Russell, P. Norvig*, Artificial intelligence: a modern approach (AIMA), Williams, Moscow, 2007.

ПРОГРАММНО-АППАРАТНАЯ ПЛАТФОРМА ДЛЯ ПОСТРОЕНИЯ ВИДЕОСИСТЕМ

С.А. БАЙРАК, Е.В. КАЛАБУХОВ, М.М. ТАТУР

*Белорусский государственный университет информатики и радиоэлектроники
ул. П. Бровки, 6, г. Минск, 220013, Республика Беларусь
bairak@bsuir.by, kalabukhov@bsuir.by, tatur@bsuir.by*

Разработан и создан макет программно-аппаратной платформы, предназначенный для построения прототипов телеуправляемых устройств, робототехнических комплексов и видеосистем.

Ключевые слова: видеопроцессор, обработка видеоданных в реальном времени, параллельная программируемая архитектура, FPGA.

Сегодня в связи с развитием компьютерных технологий, алгоритмической базы по машинному зрению, оптических систем и повышением разрешения видеокамер достаточно быстро развиваются системы машинного зрения: системы видеонаблюдения, видеоаналитики и автоматического неразрушающего контроля.

Основная идея представляемой платформы – в совмещении функций преобразования и обработки видеоинформации на достаточно небольших видеопроцессорах, имеющих программируемую архитектуру на базе FPGA [1].

Разработанная платформа состоит из трех частей (см. рис. 1):

- видеопроцессора;
- видеокамеры;
- управляющей ПЭВМ с соответствующим программным обеспечением.

В качестве источника видеосигнала в макете используется плата MT9T031C1STCH с матрицей CMOS MT9T031 от компании Aptina. Основой этой платы является сама матрица, которая в составе с объективом представляет собой видеокамеру. Матрица может быть использована в режиме по умолчанию, либо перепрограммирована пользователем на другие параметры (размер фрейма и т.д.). Режим работы по умолчанию: QXGA (2048 x 1536), 12 fps, ADC – 10 бит.

Для построения видеопроцессора в макете используется плата Xilinx EK-S6-SP605-G с FPGA Spartan-6 (XC6SLX45T) в качестве основного вычислителя. Также на данной плате реализована аппаратура ряда интерфейсов, в том числе и интерфейс для передачи выходных данных – Ethernet. Для подключения к FPGA, кроме стандартных интерфейсов, на данной плате можно использовать разъем стандарта FMC.

Основные функции видеопроцессора: получать изображение с CMOS матрицы видеокамеры, выполнять его обработку на базе микросхемы FPGA Spartan-6, выдавать результаты обработки через набор стандартных интерфейсов (Ethernet, USB) для дальнейшего использования.

Программное обеспечение платформы включает в себя:

- прошивку видеопроцессора;
- программное обеспечение для управления видеопроцессором и отображения результатов обработки.

Под прошивкой видеопроцессора понимается программное обеспечение, выполняющее необходимую коммутацию блоков FPGA для формирования структуры вычис-

лителя по заданному алгоритму, а также реализация вспомогательных интерфейсов управления и передачи данных. Для создания прошивки использовалась программная среда Xilinx ISE 14.

Для эффективной разработки кода прошивки видеопроцессора создана оригинальная программная система моделирования алгоритмов обработки изображений [2], благодаря которой затраты времени на создание прошивки могут быть значительно сокращены.

Программное обеспечение для управления видеопроцессором и отображения результатов обработки создано в упрощенном варианте, предназначенном в основном для тестирования видеопроцессора.

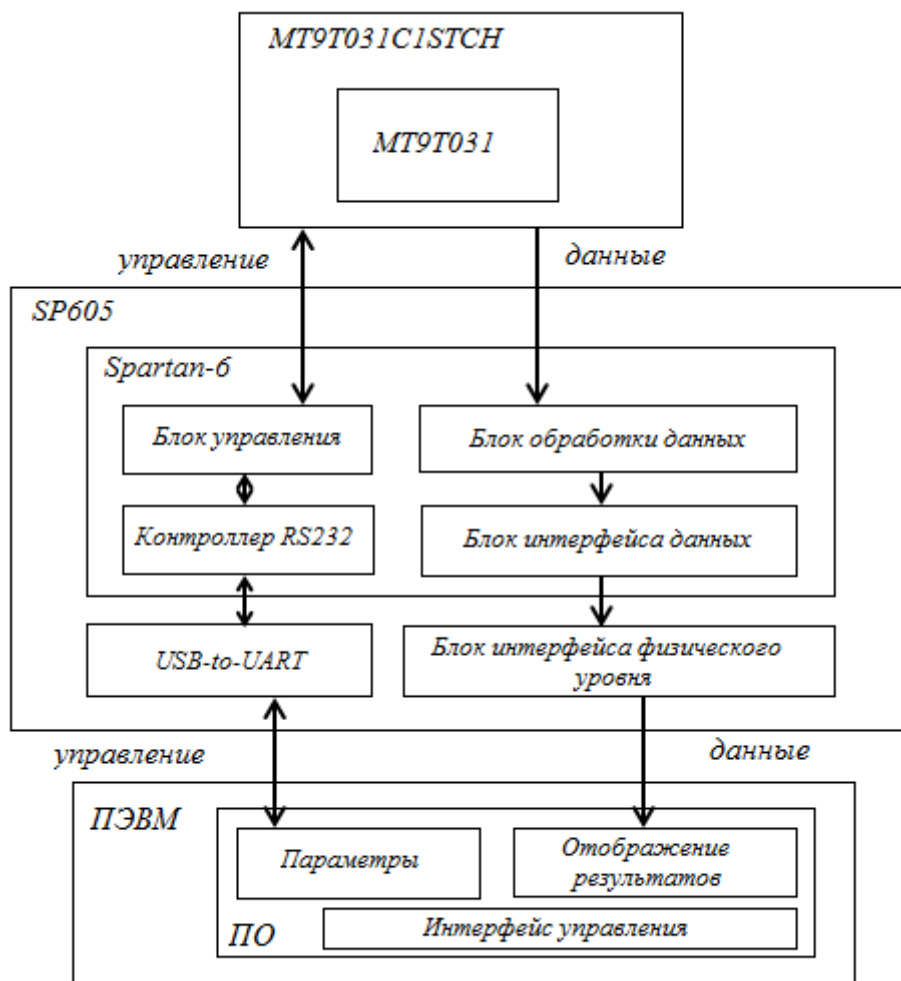


Рис. 1. Структурная схема программно-аппаратной платформы

Список литературы

1. *М.М. Татур* Parallel processor for Pattern Recognition – Science. Innovation. Production. Proceedings of the 2nd Belarus-Korean Forum 2013, 19–20 Nov., p. 75–76.
2. *М.М. Татур, Е.В. Калабухов* Программная система для разработки алгоритмов видеопроцессора // Фотоника: наука в производство, сборник материалов Белорусско-Китайского форума, 12–13 декабря 2013, БНТУ, Минск, 2013, с. 25–31.

АРХИТЕКТУРА АНАЛИТИЧЕСКОЙ ПЛАТФОРМЫ ДЛЯ ВСТРАИВАЕМЫХ ИНТЕЛЛЕКТУАЛЬНЫХ ПОДСИСТЕМ

В.С. АБАТУРОВ¹, А.Ю. ДОРОГОВ², О.В. ЗАБРОДИН, И.В. РАКОВ

*Санкт-Петербургский государственный электротехнический университет «ЛЭТИ» 197376, Россия,
Санкт-Петербург, улица Профессора Попова, дом 5
¹vasilianich@yandex.ru, ²vaksa2006@yandex.ru*

В настоящей работе предлагается вариант архитектурного решения аналитической платформы для встраиваемых интеллектуальных подсистем на основе СУБД PostgreSQL. Архитектура аналитической платформы подчинена требованиям SQL/MM и PMML. Показана схема формирования сценариев основных фаз извлечения знаний: фазы обучения, фазы тестирования и прикладной фазы. Приведены преимущества представленного архитектурного решения.

Ключевые слова: аналитическая платформа, извлечение знаний, унифицированный интерфейс.

Современные системы управления технологическими процессами требуют обработки и анализа больших объемов информации. Данная проблема давно стала критической в областях, непосредственно связанных с аналитической обработкой данных (Data Mining, искусственный интеллект, системы поддержки принятия решений, техническое зрение, мультимедиа технологии, и др.). В настоящее время рынок аналитических систем экспоненциально развивается. В этом процессе принимают участие такие крупные зарубежные компании как: IBM Cognos, MicroStrategy, Oracle, SAS, Microsoft, а также российские фирмы BaseGroup Labs, «Прогноз» [1].

Существующие стандарты Data Mining затрагивают основных аспекты построения аналитических систем: унификация интерфейсов, посредством которых любое приложение может получить доступ к функциональности аналитической платформы; разработка единого соглашения по хранению и передаче моделей Data Mining.

Среди существующего многообразия стандартов (разработанными различными международными организациями и в разное время) согласованными является пара SQL/MM [2] и PMML (Predicted Model Markup Language) [3]. Проведенный анализ показал, что стандарт SQL/MM достаточен для разработки архитектуры аналитической платформы встроеной в реляционную базу данных. Стандарт PMML полностью покрывает представление моделей алгоритмов и, кроме того, последняя версия стандарта допускает использование PMML моделей для информационных целей.

В настоящей работе предлагается вариант архитектурного решения аналитической платформы (рис. 1, а) для встраиваемых интеллектуальных подсистем на основе СУБД PostgreSQL. Взаимодействие аналитической платформы с приложениями осуществляется с помощью трех независимых интерфейсов: SQL/MM – расширение языка SQL для управления процессами извлечения знаний; SOAP – протокол коммуникаций между интернет-приложениями; DDS – открытый стандарт распределенного сервиса для систем реального времени.

Ключевую роль в данной архитектуре выполняет стандарт PMML, который применяется не только для представления и хранения моделей знаний, но и для создания интерфейса сообщений с аналитической платформой через каналы удаленного доступа.

Предложенная архитектура позволяет с помощью хранимых процедур СУБД PostgreSQL выполнять классические этапы извлечения знаний, включающие: обучающую фазу, фазу тестирования и фазу применения (рис. 1, б).

Фаза обучения – этап, на котором строится вычислительная модель интеллектуального анализа данных.

Фаза тестирования – этап, на котором осуществляется проверка качества предсказания на основе построенной модели.

Прикладная фаза – этап, на котором строка оперативных данных оценивается на основе обученной модели.

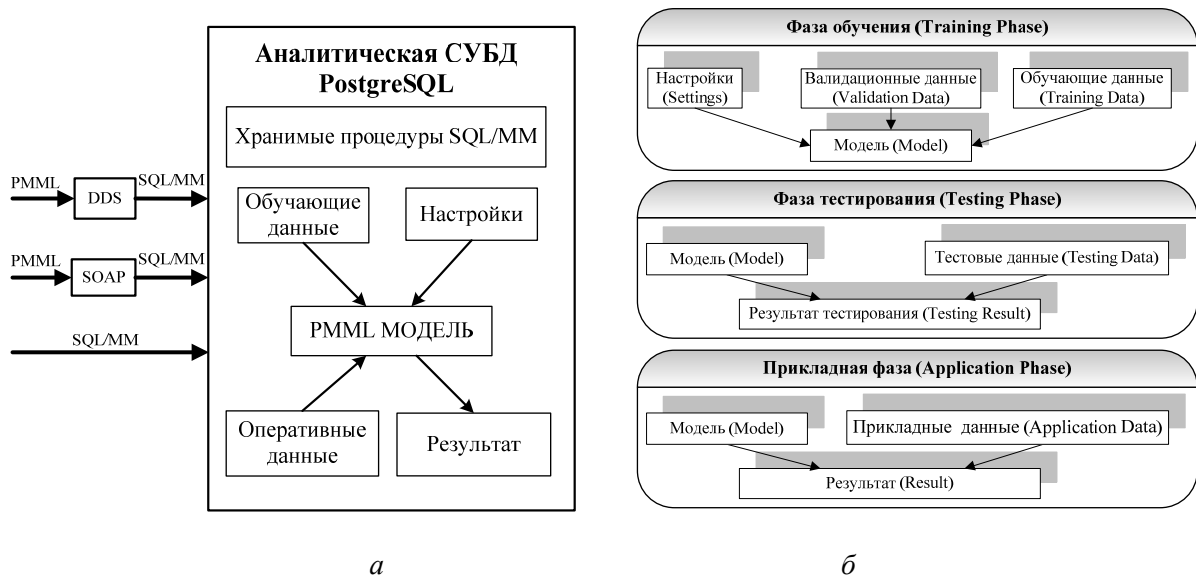


Рис. 1. Аналитическая платформа: а – архитектура аналитической платформы; б – фазы извлечения знаний

Поскольку в качестве аналитической СУБД была выбрана СУБД PostgreSQL, аналитическая платформа, реализованная в соответствии с предложенной архитектурой, будет иметь следующие тактико-технические характеристики:

- обработка масштабных массивов разнородной информации (до 32 TB);
- расширяемость и масштабируемость аналитики;
- многоплатформенность;
- контроль целостности данных;
- интерфейсы доступа к языкам программирования высокого уровня;
- триггеры и правила для управления процессами;
- система управления правами доступа и авторизации;
- параллельная обработка пользовательских сессий;
- шифрование трафика.

Преимуществами предложенной архитектуры аналитической платформы встроенной в базу данных является высокая гибкость применения алгоритмов извлечения знаний, а также простые возможности их расширения и масштабирования.

Список литературы

1. Data Mining Community. Top Resource [Электронный ресурс]. – Режим доступа: <http://www.kdnuggets.com>. Дата доступа: 23.01.14.
2. PMML Version 4.1, 2012, Data Mining Group (DMG) <http://www.dmg.org/>
3. ISO/IEC 13249-6-2006, SQL/MM Part 6.

ЗАЩИТА ИСХОДНОГО КОДА ВЕБ-ПРИЛОЖЕНИЙ МЕТОДОМ ЛЕКСИЧЕСКОЙ ОБФУСКАЦИИ JAVASCRIPT-КОДА

М.А. БАРТОШИК¹, В.Н. ЯРМОЛИК²

*Белорусский государственный университет информатики и радиоэлектроники
ул. П. Бровки, 6, г. Минск, 220013, Республика Беларусь
¹maxbartoshik@gmail.com; ²yarmolik@bsuir.by*

В настоящее время широкое развитие получили технологии создания веб-приложений с использованием HTML5, CSS и JavaScript[1]. Обратное проектирование исходного кода позволяет анализировать и модифицировать механизмы защиты приложений с целью их несанкционированного использования, что указывает на необходимость разработки технических методов защиты веб-приложений[2]. Лексическая обфускация является одним из эффективных методов затруднения процесса обратного проектирования исходного кода.

Ключевые слова: обфускация, JavaScript, защита программного обеспечения, обратное проектирование.

В последние годы возникает большое количество веб-приложений. В качестве исходного кода данных приложений используется язык JavaScript. Так как программа на данном языке выполняется на стороне клиента, любой человек имеет доступ к исходному коду приложения, может разобраться в алгоритме его работы и обойти защитные механизмы.

Для затруднения процесса обратного проектирования следует применять лексическую обфускацию исходного кода. Данный процесс включает в себя изменение имен идентификаторов и строковых литералов[3]. Суть метода заключается в преобразовании логических имен в нечитаемые и сложные для восприятия человеком последовательности символов. Формально такое преобразование можно представить в виде:

$$\{V, S\} \xrightarrow{T_{\text{лекс.}}} \{V', S'\} \quad (1)$$

где V - множество идентификаторов программы, S - множество строковых констант, используемых в программе, V' - множество преобразованных идентификаторов программы, S' - множество преобразованных строковых констант, $T_{\text{лекс.}}$ - множество лексических преобразований.

Множество лексических преобразований представлено следующим образом:

$$T_{\text{лекс.}} \in \{N, E, R, I\}, \quad (2)$$

где N – использование различных систем счисления, E – использование коди-
ровок, R – использование регулярных выражений, I – использование индексаторов.

Суть метода с использованием различных систем счисления основывается на способе представления целых положительных чисел в позиционных системах счисления. Целое число без знака x в b -ричной системе счисления представляется в виде конечной линейной комбинации степеней числа b [4]:

$$x = \sum_{k=0}^{n-1} a_k b^k, \quad (3)$$

где a_k это целые числа, удовлетворяющие неравенству $0 \leq a_k \leq b-1$.

При $b > 10$ в представлении чисел в качестве значений разрядов кроме цифр используются буквы английского алфавита, что позволяет кодировать строки при помощи числовых значений. Например, для получения строки «alert» можно воспользоваться представлением десятичного числа в 33-ричной системе счисления ($b = 33$):

```
a = (12630053).toString(33); // returns "alert"
```

Использование различных кодировок позволяет именовать идентификаторы случайным образом и представлять их различными символами (ASCII, UNICODE, HEX, OCT). Особый интерес представляют незадокументированные символы, поддерживаемые различными веб-браузерами.

Регулярные выражения сложны для восприятия человеком, их использование также увеличивает время, необходимое для изучения исходного кода приложения.

Использование индексов позволяет конструировать строки с использованием встроенных в JavaScript возможностей, например:

```
_ = $=+[],({}+[])[++$]; // returns 'o'
```

Результатом выполнения данного кода является символ “o”, который можно использовать для дальнейшего конструирования строк. Данная техника позволяет создавать код, в котором не используется ни одной цифры или буквы английского языка.

Данный метод позволяет увеличить время, необходимое для обратного проектирования исходного кода приложения, однако он не позволяет навсегда скрыть алгоритм работы необфусцированного кода. Это накладывает ограничения на область использования метода, однако он отлично подходит для сокрытия параметров, которые теряют свою актуальность со временем. Пусть время актуальности скрытых параметров равно $T_{акт.}$, а время взлома данных параметров – $T_{взл.}$. Тогда условие, при котором использование метода будет эффективным, можно представить следующим образом:

$$T_{акт.} \leq T_{взл.} \quad (4)$$

Список литературы

1. Building Cross-Platform Apps with HTML5 [Электронный ресурс]. – Режим доступа: <http://software.intel.com/en-us/articles/building-cross-platform-apps-with-html5>. – Дата доступа: 20.12.2013.
2. Chikofsky E.J., Cross J.H. // IEEE Computer Society. January 1990. P. 13–17.
3. Heiderich M. Web application obfuscation. Syngress, 2007.
4. Фомин С.В. Системы счисления. М., 1987.

ПРИНЦИПЫ УЛУЧШЕНИЯ КАЧЕСТВА СПЕЦИФИКАЦИИ ТРЕБОВАНИЙ ПРОГРАММНЫХ СРЕДСТВ

А.Ю. ЧИРКОВА¹, В.В. БАХТИЗИН²

¹Белорусский государственный университет информатики и радиоэлектроники
ул. Александрова, 2, 27, п. Лесной, Минск. р-н, 223040, Республика Беларусь
aliaksandra.chyrkova@gmail.com

²Белорусский государственный университет информатики и радиоэлектроники
ул. П. Бровки, 6, г. Минск, 220013, Республика Беларусь
bww@bsuir.by

В настоящее время большое количество разрабатываемых программных продуктов являются неудовлетворительными для конечного пользователя или заказчика или даже заканчиваются провалом. Зачастую причиной этого является плохо составленная спецификация требований к программному продукту. В тезисах доклада предложены принципы улучшения качества требований.

Ключевые слова: спецификация требований, качество, управление требованиями, модель сбора требований.

Качественные требования – это требования, которые выражают то, что основные заинтересованные лица действительно хотят получить как результат работы разрабатываемого программного продукта[1]. При создании качественных требований, а не тех, которые могли бы в лучшем случае работать в некоторой степени, но не в полной мере удовлетворять реальные потребности заинтересованных лиц, следует руководствоваться следующими ключевыми принципами подхода к составлению требований.

Принцип выявления наиболее значимых целей и задач продукта. Зачастую требования высокого уровня указаны нечетко и игнорируется командой разработчиков проекта. Основная проблема заключается в том, что требования не выражают достаточно подробно, каких результатов пытается достичь бизнес. В спецификации требований важно описать, какую выгоду, ценность или результат предоставляет проект, а не только функциональные требования и пользовательские истории.

Принцип применения различных техник при сборе требований. В настоящее время существует несколько техник сбора требований, но все они имеют свои достоинства и недостатки. Зачастую аналитик выбирает конкретный метод сбора информации по следующим причинам: аналитик знает только одну технику; аналитик пользуется одной техникой, потому что считает ее подходящей для всех случаев; аналитик использует определенную технику, которая в настоящий момент является самой распространенной; аналитик выбирает только одну эффективную технику. Очевидно, что четвертая причина демонстрирует самый высокий профессиональный уровень аналитика, что приводит к более глубокому пониманию потребностей заинтересованных сторон, и таким образом, к более высокой вероятности того, что в результате разрабатываемое программное средство будет удовлетворять всем потребностям. Однако даже этот факт не обеспечивает максимального эффекта от сбора информации. Совокупность применения нескольких техник позволяет оптимизировать процесс сбора требований.

В целях улучшения процесса сбора требований предлагается использовать единую модель выбора и применения различных техник. При сборе требований выполняется ряд мероприятий, целью каждого из которых является приведение заинтересован-

ных лиц к правильному пониманию требований. Этот ряд мероприятий можно рассматривать как ряд математических функций, $elicit_1, elicit_2, \dots$, каждая из которых создает новые требования, применяя методику сбора информации.

$$elicit_i(R_i, S_i, t_i) \rightarrow R_{i+1}, S_{i+1}, \quad (1)$$

где R_i – текущее состояние требования; R_{i+1} – новое состояние требования;
 S_i – текущее виденье задачи;

t_i – техника, применяемая для сбора информации на шаге i , t_i принадлежит множеству всех известных методов сбора информации.

На каждом шаге следует выбирать технику, наиболее применимую к текущей задаче, чтобы с большей вероятностью выявить отсутствующие требования.

Принцип квантификации требований, или определения количественных мер в требовании. Использование чисел в требованиях является базовым и мощным методом определения качества требования. При помощи квантификации осуществляется переход от неоднозначных слов (таких как «несколько» или «высококачественный») к ясности, позволяющей четко понимать и валидировать реальные требования.

Принцип разделения задач и решений. Требование должно быть составлено и абстрагировано таким образом, чтобы оно четко и однозначно определяло задачу, а не конкретное решение, которое может не описывать все ожидаемые результаты, быть недостаточно эффективным и т.д. Важно отличать частные решения и главные цели задачи проекта.

Принцип валидации требований. Все требования должны быть понятны всем участникам проекта. Требования не должны содержать неоднозначных высказываний. Разработчики должны подтвердить, что они понимают цели бизнеса, инженеры качества должны подтвердить, что они смогут провести приемочные испытания по описанным требованиям, заказчик должен подтвердить, что его ожидания соответствуют описанным требованиям.

Принцип использования методов управления качеством требований. Для снижения влияния субъективного мнения на оценку качества требований к программным средствам предлагается внедрять в процесс разработки спецификации метод управления качеством требований, базирующийся на наборе методологий совершенствования процессов СММИ[2].

Таким образом, предложенные принципы значительно расширят подходы, которые применяются в настоящее время для создания спецификаций требований к разрабатываемым программным средствам. Не только в совокупности, но и по отдельности предложенные принципы улучшат качество спецификаций требований, а, следовательно, и качество самих программных средств.

Список литературы

1. *Gilb T.*, “Competitive Engineering: A Handbook for Systems Engineering, Requirements Engineering, and Software Engineering Using Planguage,” Elsevier Butterworth-Heinemann, Boston, 2005.

2. *Чиркова А.Ю., Бахтизин В.В.* // Управление качеством требований при разработке программных средств дистанционного обучения // Дистанционное обучение – образовательная среда XXI века : материалы VIII междунар. науч.-метод. конф. (Минск, 5–6 декабря 2013 года). – Минск : БГУИР, 2013. – 407 с.

СИСТЕМА ЦЕНТРАЛИЗОВАННОГО УПРАВЛЕНИЯ УЧЕТНЫМИ ЗАПИСЯМИ И ПРАВАМИ ДОСТУПА SATORI

М.В. СТЕРЖАНОВ

*Белорусский государственный университет информатики и радиоэлектроники
ул. П. Бровки, 6, г. Минск, 220013, Республика Беларусь
accept@bk.by*

Система централизованного управления учетными записями и правами доступа – механизм, обеспечивающий синхронизацию, распространение и централизованное управление правами и учетными записями пользователей в гетерогенных информационных системах и приложениях предприятия на основе единого централизованного представления учетных данных. В работе описываются требования, архитектура, технологии реализации предлагаемой системы Satori.

Ключевые слова: централизованное управление доступом, управление учетными записями, корпоративное программное обеспечение.

При создании единого информационного пространства современного предприятия одной из наиболее сложных проблем является управление правами доступа к приложениям и сервисам внутрикорпоративной сети. Для управления доступом к сервисам разных типов приходится использовать несколько разных систем управления и выполнять большое число операций, что создает большую нагрузку на администраторов. Согласно исследованию, проведенному Gartner в 2010 году, управление учетными данными и правами доступа занимает первое место в первой пятерке приоритетов безопасности, в список которых входят межсетевое экранирование, предотвращение вторжений, предотвращение утечки данных и антивирусное обеспечение.

В сетях крупных организаций, с большим количеством сервисов и пользователей, многообразие технологий приводит к повышению сложности процесса управления доступом. Сервисы разных типов, использующие разные технологии управления доступом, требуют создания инфраструктуры управления: репозитория правил разграничения доступа, системы управления доступом и т.п. Большое количество сервисов разных типов приводит к большому количеству инфраструктур управления доступом, которые нужно создавать и поддерживать. Часть информации в таких инфраструктурах дублируется и требует синхронизации при изменениях.

Для упрощения процесса управления доступом необходимо выполнить интеграцию систем управления доступом к сервисам, использующим различные технологии управления доступом.

Система Satori должна удовлетворять следующим требованиям: интероперабельность (управление доступом к сервисам разных типов); масштабируемость (возможность добавления новых сервисов при их появлении); гибкость управления доступом (возможность сочетания нескольких популярных методов управления доступом); простота в использовании и администрировании.

В настоящее время распространены три подхода к интеграции информационных систем, обеспечивающие различные уровни интероперабельности [1]:

- синтаксическая интеграция основывается на использовании согласованных форматов данных;

- структурная интеграция обеспечивает согласование структур данных путем преобразования форматов с применением метаданных;
- семантическая интеграция устанавливает смысловое соответствие между сущностями.

Нами проектируется система централизованного управления учетными записями и правами доступа Satori, назначением которой является автоматизация процедуры управления доступом пользователей к ресурсам предприятия. Кроме того, система будет обеспечивать для разрабатываемого и поддерживаемого корпоративного программного обеспечения потребности в создании ролей и назначении прав пользователей, что освобождает ИТ-отдел предприятия от решения этих задач в каждом отдельном приложении.

Система Satori взаимодействует с тремя группами пользователей:

1. Сотрудники предприятия. Рядовым пользователям система Satori предоставляет портал самообслуживания, с помощью которого они могут проводить простые операции по управлению доступом к ИТ-ресурсам и настройке своего профиля.

2. Администраторы системы. Основной задачей администратора является управление пользователями и выполнение контроля корректности функционирования системы.

3. Внешние программы. Сторонние сервисы обращаются к системе Satori для управления доступом пользователей.

Архитектура комплекса Satori будет состоять из нескольких логически обособленных уровней:

- Поддержка репозитория правил разграничения доступа;
- Реализация прикладной логики управления доступом;
- Взаимодействие с внешними клиентами (сервисами, пользователями).

Планируется обеспечить поддержку работы с системой как через Web интерфейс, так и посредством командной строки.

Система Satori имеет распределенную архитектуру. Для реализации клиентской части системы используется технология ExtJS. К серверным приложениям относятся компоненты, отвечающие для обработку и переадресацию запросов к БД и внутреннюю логику. Для реализации данных компонент был выбран язык Java. В качестве СУБД планируется использовать продукт Oracle 11g.

Результатом внедрения системы Satori будет являться наличие единого имени и пароля у каждого пользователя для входа во все информационные ресурсы предприятия. Единая процедура управления правами имеет большие возможности по автоматизации процесса управления и поддержания данных в актуальном состоянии. Практическая значимость предлагаемой системы Satori заключается в повышении уровня управляемости и защищенности информационных систем предприятия, повышении производительности труда администраторов.

Список литературы

1. Управление учетными записями и правами доступа [Электронный ресурс]. – Режим доступа: <http://usscltd.tmweb.ru/services/22/71/?read-all=Y>. – дата доступа: 23.01.2014.

2. *Созыкин А.В.* Семантическая интеграция управления доступом к сервисам : дис... к-та тех. наук. – Пермь, 2008. – 107 с.

МАСШТАБИРУЕМЫЕ ВЕБ-ПРИЛОЖЕНИЯ

А.П. ШКОР

*Белорусский государственный университет информатики и радиоэлектроники
ул. П. Бровки, 6, г. Минск, 220013, Республика Беларусь
alex@shkor.com*

Масштабируемые веб-приложения – один из самых сложных аспектов современной веб-разработки. Тем не менее любое высоконагруженное приложение должно хорошо поддаться как вертикальному так и горизонтальному масштабированию, чтобы справляться с огромными нагрузками. Открытое программное обеспечение стало основным структурным элементом при создании некоторых крупнейших веб-сайтов. С ростом этих веб-сайтов возникли передовые практические методы и руководящие принципы их архитектуры.

Ключевые слова: масштабируемость, архитектура, cqrs, event sourcing, lambda architecture.

Анализ разработки масштабируемых веб-приложений и сведения о конкретных проблемах масштабирования.

Проектирование масштабируемого веб-приложения и выбор технологий, позволяющих прямо или косвенно добиться масштабирования веб-приложения.

Детали реализации масштабируемого веб-приложения в конкретной предметной области, выбор предметной области и описания логики предметной области.

MongoDB - это документно-ориентированная база данных, предназначенная для гибкой, масштабируемой и очень быстрой работы даже при больших объемах данных. При ее проектировании изначально закладывалась высокая доступность, поддержка сложных динамических схем и простое распределение данных по нескольким серверам.

MongoDB написана она на языке C++ и распространяется в рамках лицензии Creative Commons. Так же стоит заметить, что MongoDB хранит данные не в виде таблиц, а в виде коллекций и хранит их в формате BSON, который является бинарным подвидом языка JSON. Данные хранящиеся в коллекции MongoDB не являются строго типизированными.

CQRS расшифровывается как Command Query Responsibility Segregation (разделение ответственности на команды и запросы). Это паттерн проектирования. В его основе лежит простое понятие, что вы можете использовать разные модели для обновления и чтения информации. Однако это простое понятие ведет к серьезным последствиям в проектировании информационных систем.

Многие компании сталкиваются с необходимостью хранить и анализировать большие объемы данных (порядка терабайт и более). Рано или поздно системы хранения перерастают возможности отдельного сервера, и перед разработчиками и архитекторами встает проблема выбора распределенной системы, а также стандартные вопросы вроде производительности, масштабируемости, отказоустойчивости и т. д. Существует несколько подходов к созданию распределенных систем хранения данных, которые по-разному выполняют перечисленные требования. Однако не все из них хорошо подходят для анализа больших объемов данных. Универсальных инструментов не существует.

ИМИТАЦИЯ УПРАВЛЕНЧЕСКОЙ ДЕЯТЕЛЬНОСТИ С ИСПОЛЬЗОВАНИЕМ ЭЛЕМЕНТОВ РЕФЛЕКСИИ

Е.В. РУЛЬКО

*Военная академия Республики Беларусь
пр-т Независимости, 220, г. Минск, 220057, Республика Беларусь
achtung-2@yandex.ru*

При создании систем имитационного моделирования военных действий необходимо чтобы поведение модельного объекта на виртуальном поле боя было максимально приближено к поведению реального объекта в тех же условиях обстановки. В связи с этим требуется разработка методики имитации управленческой деятельности.

Ключевые слова: моделирование военных действий, рефлексивное управление.

Основной задачей научно – исследовательского центра моделирования военных действий УО «Военная академия Республики Беларусь» является создание системы имитационного моделирования, которая представляет собой совокупность отдельных и агрегированных моделей, воспроизводящих характерные особенности функционирования тех или иных реальных объектов. Главное требование к такой системе моделирования – это адекватное воспроизведение событий реальных военных действий. Исходя из этого, необходимо чтобы поведение модельных объектов не противоречило логике здравого смысла.

Анализ исторического опыта ведения вооруженной борьбы свидетельствует в пользу того, что успех военных действий во многом зависит от умения командира творчески подойти к задаче управления и способности перехитрить противника.

Для формализации данного вида мыслительной деятельности противоборствующих сторон наиболее близким по сути является математический аппарат рефлексии. Основная идея рефлексивного управления заключается в том, что принимающий решение в условиях противоборства ставит себя мысленно на место противника и, имитируя его рассуждения, принимает решение за противостоящую сторону, которое затем учитывает при планировании своих действий. Этот процесс может повторяться несколько раз, то есть полученное таким образом решение может использоваться для повторной имитации рассуждений противника с учетом того, что это решение в той или иной степени противнику известно.

Для решения задачи рефлексивного управления предлагается создание двух независимых подсистем – подсистемы ситуационного управления и подсистемы рефлексирования.

Подсистему ситуационного управления предлагается построить в соответствии со структурной схемой, приведенной на рис. 1.

Любая система ситуационного управления функционирует по принципу «ситуация - действие», следовательно, необходим механизм идентификации данных проблемных ситуаций, при наличии которых необходимо принимать определенные действия (например, нами обнаружен противник). Проблемная ситуация представляет собой по сути дела факт того, что определенные параметры системы находятся в заданных пределах. Следовательно, для введения пользователем проблемных ситуаций необходимо предоставить ему набор возможных системных параметров и конструктор выражений,

позволяющий создавать логические выражения, в которых данные параметры сравниваются с граничными значениями.



Рис. 1. Система ситуационного управления

После того, как проблемная ситуация будет сформирована, она должна возникнуть в системе. Следовательно, необходим механизм, осуществляющий периодическое сравнение указанных в конструкторе выражений параметров с пороговыми значениями, указанными в нем же. Эту роль выполняет модуль анализа ситуаций.

Экспертную систему предлагается построить на основе метода анализа иерархий.

Подсистема рефлексирования может обращаться к подсистеме ситуационного управления системе моделирования в целом.

Суть предлагаемой методики рефлексивного управления концептуально заключается в следующем: если в текущий момент времени имеет место описанная проблемная ситуация, организуется цикл перебора возможных сочетаний вероятных структур информированности и вероятных структур доктрин моделируемого объекта. Для каждого из их возможных сочетаний осуществляется последовательное принятие решений системой ситуационного управления за противника, на основании которых осуществляется принятие решений за моделируемый объект в иерархии рефлексивных рассуждений. После принятия окончательного решения моделируемым объектом и вывода соответствующего ему варианта поведения, организуется вложенный цикл перебора альтернативных сочетаний вероятных структур информированности и вероятных структур доктрин моделируемого объекта, включающий также и текущий вариант сочетания вероятной структуры информированности и вероятной структуры доктрин определенной внешним циклом. После чего для текущего варианта поведения моделируемого объекта определяемого внешним циклом и сочетания вероятной структуры информированности и структуры доктрин, определяемого внешним циклом, путем проведения имитационных экспериментов находится система показателей успешности развития событий и соответствующих им альтернативных вариантов доктрины и информированности противника с соответствующими им условными вероятностями. После завершения внутреннего цикла перебора альтернативных вариантов структур информированности и доктрин, для рассматриваемого во внешнем цикле варианта поведения моделируемого объекта, осуществляется подсчет общего показателя успешности развития событий для рассмотренных во внутреннем цикле альтернативных сочетаний структур доктрин и структур информированностей. В качестве конечного варианта поведения моделируемого объекта выбирается вариант с наибольшим значением общего показателя успешности развития событий.

ВЕРИФИКАЦИЯ ПРОЕКТОВ СБИС: ПРОБЛЕМЫ И РЕШЕНИЯ

Л.А. ЗОЛОТОРЕВИЧ

*Белорусский государственный университет информатики и радиоэлектроники
ул. П. Бровки, 6, г. Минск, 220013, Республика Беларусь
zolotorevichla@bsuir.by*

Дается анализ состояния проблемы верификации проектов СБИС. Рассматриваются вопросы построения тестов на основе функциональных и структурных моделей.

Ключевые слова: верификация проектов СБИС, построение тестов, применение методики АВУ.

Разработка методов верификации проектов, моделирования и построения тестов контроля цифровых устройств и систем являются наиболее сложными, наукоемкими и многоплановыми задачами проектирования функционально-сложных объектов.

По мере возрастания объема и функциональной сложности проектов время, требуемое на верификацию, увеличивается и в отдельных случаях достигает 90% и более от всего времени разработки проекта. Моделирование остается широко применяемым на практике методом верификации проектов, при этом полнота применяемых тестов определяет полноту верификации проектов.

Исторически сложилось, что на начальных этапах развития цифровой электроники основным методом контроля объектов на всех этапах жизненного цикла была проверка реализации устройств заданных функций, основанная на применении функциональных тестов. При этом не учитывалась структура аппаратной реализации, возможные ошибки проектирования, а также дефекты, возможные в реальном объекте. На определенном этапе оказалось, что построить функциональный тест, который способен обеспечить требуемое покрытие в указанном спектре проблем практически невозможно. В результате этого усилия исследователей оказались направленными на построение тестов контроля схемных реализаций цифровых структур.

В литературе наиболее широко исследуются вопросы разработки тестов в различных классах логических неисправностей. При этом рассматриваются разные уровни представления исследуемого проекта, от структурного представления до системного уровня и уровня межрегистровых передач. Разработаны методы и программные средства построения тестов контроля объектов, в основном, в классе неисправностей константного типа. В то же время полученные результаты оказались далеко не удовлетворительными для решения задач проектирования сложно-функциональных объектов большого объема по следующим причинам:

- 1) Отсутствуют практически пригодные методы для построения тестов контроля последовательностных устройств;
- 2) Программные средства генерации тестов не работают применительно к цифровым системам большого объема;
- 3) Все полученные решения направлены разработку тестов контроля неисправностей константного типа. В то же время эти тесты покрывают не все возможные неисправности других классов (обрыва, замыкания, типа ПЗТ (постоянно закрытый транзистор)) и др. Высокие требования к обеспечению надежности систем требуют обеспечения контроля всех возможных дефектов.

На сегодняшний день построение тестов и верификация проектов современных цифровых сложно-функциональных систем лежит в плоскостях как функционального,

так и структурного синтеза тестов, во многом использует подходы, основанные на эрудиции и интуиции и доступно лишь высококвалифицированному составу разработчиков.

Разработан метод направленного построения тестов контроля на начальных этапах проектирования, основанный на иерархическом подходе и использующий тесты функциональных блоков, которые являются компонентами механизмов реализации операторов программного кода описания объекта. Для решения данной задачи необходимы средства построения тестов контроля объектов, представленных в разных системах идентификации (списком связности компонентов структуры, некоторым представлением булевой функции и др.).

Рассматриваются неисправности типа задержки, которые способны влиять на корректность системы синхронизации синхронных структур. Класс неисправностей задержки относится к классам не перечислимых неисправностей. Неисправности задержки могут вызываться ошибками проектирования, в результате которых нарушаются условия синхронизации работы блоков и системы в целом. На этапе производства корректность функционирования системы синхронизации может нарушаться вследствие неточности технологии, влияния ряда конструктивно-технологических факторов. Кроме того, на этапе эксплуатации под воздействием дестабилизирующих факторов влияния внешней среды могут изменяться задержки на срабатывание элементов. Важным является то, что неисправности задержки приводят не только и не столько к понижению быстродействия объекта, а к нарушению алгоритмов его функционирования. Поэтому построению тестов контроля неисправностей задержки в литературе уделяется много внимания. Особенно актуальна данная задача при разработке систем на кристалле.

Разработан алгоритм построения теста для контроля неисправностей задержки цифровой структуры, основанный на моделировании процесса распространения фронта сигнала от входа к выходу методом моделирования неисправности константного типа объекта в двух временных интервалах функционирования.

Рассмотрены особенности квазистатического моделирования цифровых устройств в разных алфавитах с целью анализа состязаний сигналов. Показано, что увеличение числа переменных в логической модели ограничивает скорость моделирования и не позволяет достичь точности получаемых моделей, достаточной для верификации проектов. Предложен метод квазистатического моделирования цифровых устройств при построении тестов контроля объектов с высокоимпедансными линиями связи, основанный на применении четырехзначного алфавита. Метод может быть применен при построении тестов на основе методов многозначного моделирования.

В последнем десятилетии произошел возврат к применению первоначального подхода к построению тестов – функциональному тестированию. Данный подход получил развитие на основе новой платформы, основанной на применении ассертов (утверждений), проверка которых закладывается в описание объекта. Для реализации идеи разработаны языки SystemVerilog Assertions и PSL – Property Specification Language.

Основная идея верификации на основе применения ассертов заключается в том, чтобы формализовать знания разработчика о работе проекта и использовать её при моделировании в автоматическом режиме.

В докладе рассматривается практическое применение подхода и его взаимосвязь с методами построения тестов.

ОСОБЕННОСТИ НЕЛИНЕЙНЫХ ФИЛЬТРОВ В ПРИКЛАДНЫХ ЗАДАЧАХ

О.П. СОЛОВЕЙ¹, Н.Н. ИВАНОВ²

*Белорусский государственный университет информатики и радиоэлектроники
ул. П. Бровки, 6, г. Минск, 220013, Республика Беларусь
¹oleg@solovey.me; ²nnivanov@gmail.com*

В приложениях нелинейные фильтры представляют более высокую точность оценки по сравнению с фильтром Калмана и его производными. В сообщении сравниваются частичные фильтры и фильтры Калмана. Основная проблема практического применения нелинейных фильтров заключается в их вычислительной сложности, которую по аналогии с динамическим программированием часто называется «проклятием измерений», однако сеществуют модификации нелинейных фильтров, которые избегают эту проблему.

Ключевые слова: нелинейные фильтры, фильтр Калмана, частичные фильтры.

Фильтр Калмана чрезвычайно полезен в разнообразных прикладных приложениях, включая робототехнику, коммуникационные системы, GPS, инерциальную навигацию, предсказания погоды, трекинг авиатранспорта, спутников, кораблей, ракет, людей, а также в приложениях предсказания состояния фондовых рынков. Более того, фильтры Калмана относительно легко спроектировать и реализовать, что довольно часто дает довольно точную оценку. С другой стороны, точность фильтров Калмана может быть неожиданно плохой в некоторых определенных приложениях по нескольким причинам, таким как: 1) нелинейность в уравнениях, описывающих физическую систему; 2) плохо обусловленная ковариационная матрица; 3) неточность или незавершенность моделей основной физической проблемы.

Ключевая практическая проблема нелинейных фильтров заключается в вычислительной сложности для достижения необходимой точности оценки в реальном времени. Однако со времени опубликования известной статьи Калмана в 1960 г. [1] скорость вычислений и доступная память выросли в разы. В результате в настоящее время представляется возможным использовать нелинейные фильтры в приложениях реального времени на отдельно взятом персональном компьютере для множества практических приложений. Вычислительная сложность таких фильтров зависит от множества факторов: 1) размерность вектора состояний; 2) разреженность ковариационной матрицы; 3) скорость самих измерений; 4) требуемая точность вектора состояния; 5) степень нелинейности; 6) вид распределения вероятности (унимодальное, мультимодальное).

Частичные фильтры (Particle Filters) — относительно новый класс нелинейных фильтров, анонсированных в основополагающей статье около десяти лет тому назад. С того времени появилась масса статей и приложений в области Частичных фильтров [2]. Не существует единственного Частичного фильтра, а существует огромное множество классов Частичных фильтров. Любой исследователь может создать достаточно хороший Частичный Фильтр из 40–100 операторов MatLab. Также совсем не обязательно владеть тонкостями стохастического исчисления, уравнениями Фоккера-Планка или другими современными численными методами для решения дифференциальных уравнений в частных производных. Частичные Фильтры аппроксимируют полностью негауссову плотность вероятности вектора состояния, обусловленную измерениями.

Ключевая проблема любого нелинейного фильтра – «проклятие размерности». Это понятие, придуманная Ричардом Беллманом около сорока лет назад, описывает экспоненциальный рост вычислительной сложности как функция измерения вектора состояния. Вычислительная сложность фильтра Калмана прямопропорциональна кубу размерности вектора, однако при решении общих нелинейных проблем с использованием фильтров, которые дают оптимальную точность, вычислительная сложность растет экспоненциально с измерением вектора состояния. Эта проблема не является математической, но она исключительно важна для инженеров, решающих конкретные прикладные задачи. Очевидно, что постоянный рост возможностей вычислительной техники не сможет преодолеть экспоненциальную сложность задачи.

Ранее предполагалось, что Частичные фильтры полностью решат проблему «проклятия измерений», однако этого неслучилось. Детальный анализ проблем вычислительной сложности Частичных фильтров представлен в [3, 4]. Для задач с малой размерностью вектора состояний Частичные фильтры достигают оптимальной точности оценки сравнимой с вычислительной сложностью фильтра Калмана. Однако для решения задач с большой размерности вычислительная сложность лучших современных Частичных фильтров огромна. Хотя с точки зрения математики Частичный фильтр успешно решает прикладные задачи, его программная реализация зачастую не гарантирует получения решения за приемлемое время. На сегодняшний день существует множество публикаций, посвященных сокращению вычислительной сложности для решения задач с заданной оценкой точности.

Список литературы

1. *Kalman R.E.* // Journal of Basic Engineering Transactions ASME. 1960. №82 (Series D). P. 35–45.
2. *Ristic B., Arulampalam S., Gordon N.* Beyond the Kalman Filter: Particle Filters for Tracking Applications. 2004.
3. *Daum, F., Raytheon C., Huang, J.* // In proceedings of IEEE Conference on Aerospace (Volume 4), Big Sky, MT, March 8-15 2003. P.1979–1993.
4. *Daum, F., Raytheon C., Huang, J.* // In proceedings of Signal and Data Processing of Small Targets Conference (Volume 5204), San Diego, CA, 2003. August 3 2003. P. 458–479.

О ПРИВОДИМОСТИ НЕКОТОРЫХ ДИНАМИЧЕСКИХ СИСТЕМ С ХАОТИЧЕСКИМ ПОВЕДЕНИЕМ К АВТОМОДЕЛЬНОМУ УРАВНЕНИЮ КУРАМОТО – СИВАШИНСКОГО

В.В. ЦЕГЕЛЬНИК

*Белорусский государственный университет информатики и радиоэлектроники
ул. П. Бровки, 6, г. Минск, 220013, Республика Беларусь
tsegvv@bsuir.by*

Показано, что нелинейное обыкновенное дифференциальное уравнение третьего порядка, к которому приводятся известные динамические системы с хаотическим поведением (системы из списка Спротта, тороидальная система Рёсслера), есть точная автомодельная редукция известного уравнения Курамото – Сивашинского.

Ключевые слова: системы с хаотическим поведением, уравнение Курамото – Сивашинского.

В работе рассматриваются системы [1]

$$\dot{x} = -z, \quad \dot{y} = -x^2 - y, \quad \dot{z} = \alpha + \beta x + y, \quad (1)$$

$$\dot{x} = -z, \quad \dot{y} = x - y, \quad \dot{z} = \alpha x + y^2 + \beta z, \quad (2)$$

$$\dot{x} = -x - \alpha y, \quad \dot{y} = x + z^2, \quad \dot{z} = \beta + x \quad (3)$$

и система [2]

$$\dot{x} = -y - z, \quad \dot{y} = x, \quad \dot{z} = \alpha(y - y^2) - \beta z \quad (4)$$

с произвольными фиксированными параметрами α и β .

Системы (1) – (3) являются обобщениями систем M , Q , S из списка Спротта [2]. Система (4) обобщает известную тороидальную систему Рёсслера [3].

Характерной (с качественной точки зрения) особенностью систем (1) – (4) является их хаотическое поведение при определенных значениях входящих в них параметров, в частности, наличие странных аттракторов.

Интерес к построению и исследованию таких систем связан с их приложениями в различных разделах науки, техники, экономики и т. д. [3]. В частности, системы с хаотическим поведением используются при защите информации в телекоммуникациях.

В работе [4] показано, что каждая из систем (1) – (4) с точностью до линейного преобразования одной из неизвестных компонент эквивалентна уравнению

$$\ddot{q} = k_1 \dot{q} + k_2 q + q^2 + k_3, \quad (5)$$

в котором коэффициенты k_i ($i = \overline{1,3}$) являются функциями параметров α, β .

Несложно убедиться, что уравнение (5) представляет собой автомодельную редукцию хорошо известного уравнения Курамото – Сивашинского [5]

$$p_s + v p_{\tau\tau\tau} + b p_{\tau\tau} + \mu p_{\tau\tau} + p p_{\tau}, \quad (6)$$

в переменных бегущей волны. Действительно, полагая $p(s, \tau) = c + p(t)$, $t = \tau - cs$ относительно $p(t)$ получим дифференциальное уравнение $v \overset{\dots}{p} + b \ddot{p} + \mu \dot{p} + p \dot{p} = 0$, первым интегралом которого является уравнение

$$v \ddot{p} + b \dot{p} + \mu p + \frac{p^2}{2} + A = 0, \quad (7)$$

с произвольной постоянной A . Легко видеть, что уравнение (7) с точностью до обозначений совпадает с (6). Отметим, что в (6) ν, b, μ – произвольные фиксированные действительные параметры, причем $\nu \neq 0$.

В [6] показано, что уравнение (7) не проходит формальный тест Пенлеве. Установлено, что системы (1) – (4) также не проходят тест Пенлеве при наличии хаотического поведения.

Проведен Пенлеве-анализ уравнений

$$\ddot{q} = k_1 \dot{q} + k_2 q + q \dot{q} + k_3, \quad (8)$$

$$\ddot{q} = k_1 \ddot{q} + k_2 \dot{q} + k_3 q^2 + \dot{q} q + k_4, \quad (9)$$

$$\ddot{q} = k_1 \ddot{q} + k_2 \dot{q} + k_3 q^2 + q \ddot{q} + k_4, \quad (10)$$

$$\ddot{q} = k_1 \dot{q} + k_2 q^2 + k_3 \dot{q}^2 + q \ddot{q}, \quad (11)$$

$$\ddot{q} = k_1 \ddot{q} + k_2 \dot{q} + k_3 q^2 + k_4 \dot{q}^2 + q \ddot{q} + k_5, \quad (12)$$

$$\ddot{q} = k_1 \ddot{q} + k_2 \dot{q} + k_3 q^2 + k_4 \dot{q}^2 + k_5 q \dot{q} + q \ddot{q} + k_6 \quad (13)$$

с коэффициентами $k_i (i = \overline{1,6})$, зависящими от параметров α, β . Уравнениям (8) – (13) эквивалентны группы или отдельные системы, приведенные в [1]. Установлено, что ни одно из уравнений не удовлетворяет тесту Пенлеве.

Список литературы

1. *Eichhorn R., Linz S.J., Hänggi P.* // Phys. Rev. E. 1998. Vol. 58, № 6. P. 7151 – 7164.
2. *Sprott J.C.* // Phys. Rev. E. 1994. Vol. 50, № 2. P. R 647 – R 650.
3. *Rössler O.E.* // Ann. (N.Y) Acad. Sci. 1979. Vol. 316. P. 376 – 392.
4. *Chen G.* Controlling chaos and bifurcations in engineering systems. CR C Press. Boca Raton. FL. 1999.
5. *Kuramoto Y., Tsuzuki T.* // Prog. Theor. Phys. 1976. Vol. 55, № 2. P. 356 – 369.
6. *Конт Р., Мюзетт М.* Метод Пенлеве и его приложения. Москва – Ижевск, 2011.

МЕЖКАНАЛЬНАЯ СПЕКТРАЛЬНАЯ КОРРЕЛЯЦИЯ В ЗАДАЧАХ СЖАТИЯ ГИПЕРСПЕКТРАЛЬНЫХ ИЗОБРАЖЕНИЙ

Д.Ю. ПЕРЦЕВ¹, А.А. ДУДКИН²

¹Белорусский государственный университет информатики и радиоэлектроники
ул. П. Бровки, 6, г. Минск, 220013, Республика Беларусь
DmitryPertsev@gmail.com

²Объединенный институт проблем информатики НАН Беларуси
ул. Сурганова, 6, г. Минск, 220012, Республика Беларусь
doudkin@newman.bas-net.by

Проведен анализ межканальной спектральной корреляции, а также перспективные направления в области сжатия гиперспектральных изображений и использующих информацию о корреляции, их достоинства и недостатки.

Ключевые слова: гиперспектральные изображения, спектральная корреляция, сжатие данных.

Гиперспектральное изображение представляет собой куб данных, включающий в себя пространственную информацию (2D) об объекте и расширенную данными о спектре для каждой координаты в пространстве (1D). Учитывая высокое пространственное разрешение (наземный интервал дискретизации) и охватываемое количество спектральных каналов, объем данных, получаемых в единицу времени, может измеряться гигабайтами данных, что усложняет передачу данных на Землю и их хранение.

Большинство современных алгоритмов сжатия гиперспектральных изображений основаны на поиске оптимальной последовательности расположения спектральных каналов для последующего применения алгоритмов, понижающих избыточность данных. В основе данного подхода применяется корреляция (для гиперспектральных изображений различают пространственную и спектральную корреляцию). Особенностью изображений такого класса является высокая степень корреляции (для большинства изображений), что объясняется тем, что близкорасположенные места практически состоят из одного материала (например, лес, вода или почва). При этом большинство алгоритмов в основном работает со спектральной корреляцией [1], которая рассчитывается по формуле (1):

$$c_{u,v} = \frac{\sum_{i=1}^M \sum_{j=1}^N \tilde{x}_{i,j,u} \cdot \tilde{x}_{i,j,v}}{\sqrt{\sum_{i=1}^M \sum_{j=1}^N \tilde{x}_{i,j,u}^2 \cdot \sum_{i=1}^M \sum_{j=1}^N \tilde{x}_{i,j,v}^2}} \quad (1)$$

где $\tilde{x}_{i,j,k} = x_{i,j,k} - \bar{x}_k$, $x_{i,j,k}$ – пиксель изображения с координатами (i, j, k) , \bar{x}_k – среднее арифметическое для канала k , M и N – размерность пространства.

Для анализа корреляции были выбраны спутниковые изображения, полученные с использованием спектрометра AVIRIS. Результат для откалиброванного изображения Yellowstone, сцена 11 приведен на рис. 1 (в сравнении с каналом 116). Среднее значение пространственной корреляции составляет около 0,96, минимальное значение – 0,87.

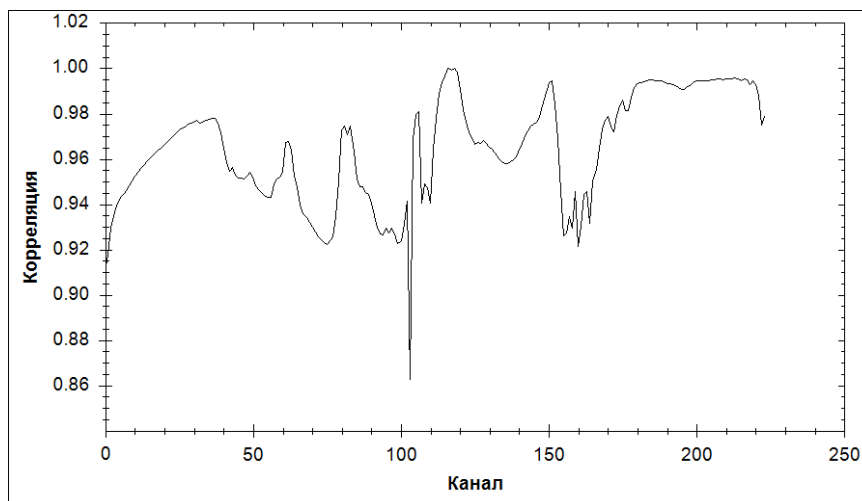


Рис. 1. Спектральная корреляция для изображения Yellowstone, сцена 11 (откалиброванные данные)

Анализ современных алгоритмов в области сжатия гиперспектральных изображений формирует два основных подхода к сжатию, в которых используются данные по спектральной корреляции:

- основанный на декорреляции данных, т.е. на понижении степени избыточности с сохранением возможности восстановить всю информацию;
- основанный на устранении избыточных каналов без возможности их восстановить.

Наиболее перспективными алгоритмами в случае декорреляции данных является предиктивное кодирование. Данный тип алгоритмов может быть реализован аппаратно и применяться на спутнике для обеспечения предварительного сжатия. Однако основной проблемой данного типа алгоритмов является низкая степень сжатия (при анализе алгоритмов не было найдено ни одного алгоритма, который на реальных спутниковых данных обеспечил коэффициент сжатия более 4 раз).

В основе алгоритмов, основанных на устранении избыточных каналов, находится принцип перехода от данных к информации, которая хранится в этих данных. Это позволяет добиться более высокой степени сжатия. Однако данный тип алгоритм трудно реализуем аппаратно и требует четкого представления об информации, которая может храниться в гиперспектральном изображении.

Список литературы

1. Zhou Z., Tan Y., Liu J. // International Conference on Wireless Communications, Networking and Mobile Computing. Wuhan, China, 22-24 September 2006. P. 1–4.
2. Chengfu Huo, Rong Zhang, Dong Yin // International Journal of Remote Sensing, 2012. Vol. 33, Issue 5. P. 1586-1604.
3. Wang W., Zhao Z., Zhu H. // 2nd International Congress on Image and Signal Processing, Tianjin, China, 17–19 October 2009. P. 1–5.
4. Chang Chein-I. Hyperspectral data processing. Maryland, 2013.

К МОДЕЛИРОВАНИЮ ПОВЕДЕНИЯ ВИРУСОВ В КОМПЬЮТЕРНОЙ СЕТИ

А.В. БОРЗЕНКОВ¹, О.Л. КОНОВАЛОВ²

¹Белорусский государственный университет информатики и радиоэлектроники
ул. П. Бровки, 6, г. Минск, 220013, Республика Беларусь
borzenkov_a@mail.ru

²Белорусский государственный университет
ул. П. Бровки, 6, г. Минск, 220013, Республика Беларусь
konovalovol@bsu.by

Рассматриваются модели взаимодействия двух антагонистических классов программного обеспечения в компьютерной сети. Один класс программ (вирусы) стремится нанести вред работоспособности сети, другой класс (антивирусы) пытается нейтрализовать эти действия. Для моделирования используются системы нелинейных обыкновенных дифференциальных уравнений и экстремальные задачи на состояниях систем дифференциальных уравнений.

Ключевые слова: интернет, патч, вирус, антивирус, обыкновенные дифференциальные уравнения, задача оптимального управления, оптимальное управление типа обратной связи.

Проведенные численные исследования для рассмотренных моделей типа вирус-антивирус [1],[2], в целом подтвердили эффективность используемых подходов. Рассмотренная модель является сосредоточенной. Для более детального исследования необходимо привлекать дифференциальные уравнения в частных производных и эмулировать распределенные модели.

Наряду с мониторингом состояния сетей актуальной остается проблема отражения хакерских атак на компьютерные сети. Хакерская атака на сеть моделируется нелинейной системой обыкновенных дифференциальных уравнений с неопределенной функцией возмущения в правой части системы. Требуется противодействовать атаке, то есть в режиме реального времени строить функцию, которая, управляя поведением уравнений, обеспечивает количество неинфицированных машин, достаточных для устойчивой работы компьютерной сети. Это задача построения оптимального управления типа обратной связи по быстродействию. Численный эксперимент, проведенный для похожей задачи в частных производных [3], дал обнадеживающий результат.

Список литературы

1. Борзенков А.В., Коновалов О.Л., и др. // Сб. науч. статей Информационные компьютерные технологии: проектирование, разработка, применение. ГрГУ им. Я. Купалы, 2013. С. 97–106.
2. Борзенков А.В., Коновалов О.Л., Анисеев А.А. // Матер. 2-й междунар. НПК Веб-программирование и интернет-технологии WebConf 2012. Минск, 5–7 июня 2012. С. 130.
3. Арико И.В., Борзенков А.В., Борзенков В.А. // Сб. науч. статей Современные информационные компьютерные технологии. ГрГУ им. Я. Купалы, 2008. С. 170–173.

ПРИМЕНЕНИЕ XSLT ПРЕОБРАЗОВАНИЙ В АНАЛИТИЧЕСКОЙ СУБД

Д.Д. ЧЕРКАСОВ¹, О.В. ЗАБРОДИН²

¹*Санкт-Петербургский государственный политехнический университет
ул. Политехническая, 5, 195251, г. Санкт-Петербург, Россия
daniil-cherkasov@mail.ru*

²*Санкт-Петербургский государственный электротехнический университет «ЛЭТИ»
ул. Профессора Попова, 5, 197376, г. Санкт-Петербург, Россия
zolegzabrodin@gmail.ru*

Рассматривается применение языка преобразований XSLT в аналитике СУБД в контексте стандарта SQL/MM. Выполнен анализ реализации функционала аналитической подсистемы стандарта SQL/MM с использованием XML технологии. Показана структура типов аналитической подсистемы, определяющая интерфейс взаимодействия с моделью.

Ключевые слова: извлечение знаний, аналитическая платформа, унифицированный интерфейс.

Введение

Тенденция последних лет в развитии интеллектуальных систем заключается в интеграции средств аналитической обработки, алгоритмов извлечения знаний, управления метаданными и визуализации результатов на одной программной аналитической платформе. Проведенный анализ показал, что стандарт SQL/MM [1], согласованный с форматом представления моделей извлечения данных PMML [2], достаточен для разработки архитектуры аналитической платформы и удовлетворяет основным требованиям: организация процесса аналитической обработки данных, унификация интерфейсов, единое соглашение по хранению и передаче моделей DataMining.

Начало XML технологии было положено в 1996 году. Консорциум всемирной паутины (W3C) обозначил задачу создания такого языка разметки, который обеспечивал стандартный способ структурирования информации, т.е. отделение данных от их представления и передачу ее из одной программы в другую. Такой язык получил название extensibleMarkupLanguage (XML) – расширяемый язык разметки. Спустя некоторое время возник вопрос о возможности преобразования данных представленных в виде XML-документа, эта предпосылка определила язык преобразований XSLT, что означает eXtensibleStylesheetLanguage: Transformations (расширяемый язык таблиц стилей: Преобразования).

Предусмотренное стандартом SQL/MM представление алгоритмов в виде PMML моделей основано на языке разметки XML. Такое решение позволяет интегрировать в систему аналитики новые разработки при сохранении действующего функционала.

Основными методами стандарта SQL/MM, выполняют функции преобразования одного XML документа в другой без вычислительных процессов, и XSLT – идеальный инструмент для этой работы. Рассмотрим пример выполнения метода DM_getClasTask() класса DM_ClasModel.

Описание метода:

1) Результат выполнения метода *DM_getClasTask()* определяется следующим образом:

а) Если *DM_ClasModel* не содержит значение *DM_ClasBldTask* которое было использовано для построения модели тогда возвращается значение null.

b) В противном случае возвращается значение *DM_ClasBldTask* которое было использовано для вычисления модели.

Согласно описанию метода входной XML-документ должен быть представлен типом *DM_ClasModel*, который должен соответствовать заданной XMLсхеме:

1) *DM_ClasBldTask* является абстракцией всей информации, которая определяет задачу классификации, в частности содержит входные данные для обучения и валидации (*trainData*, *validationData*) и параметрические настройки (*settings*)

2) Тег *PMML* содержит *PMML*-модель. Применение стандарта *PMML* дает целый ряд важных технологических и организационных преимуществ для нейросетевой технологии.

3) Атрибуты и пара тегов: *RankQuality*, *PredAccuracy* заданные в схеме являются качественными характеристиками построенной модели.

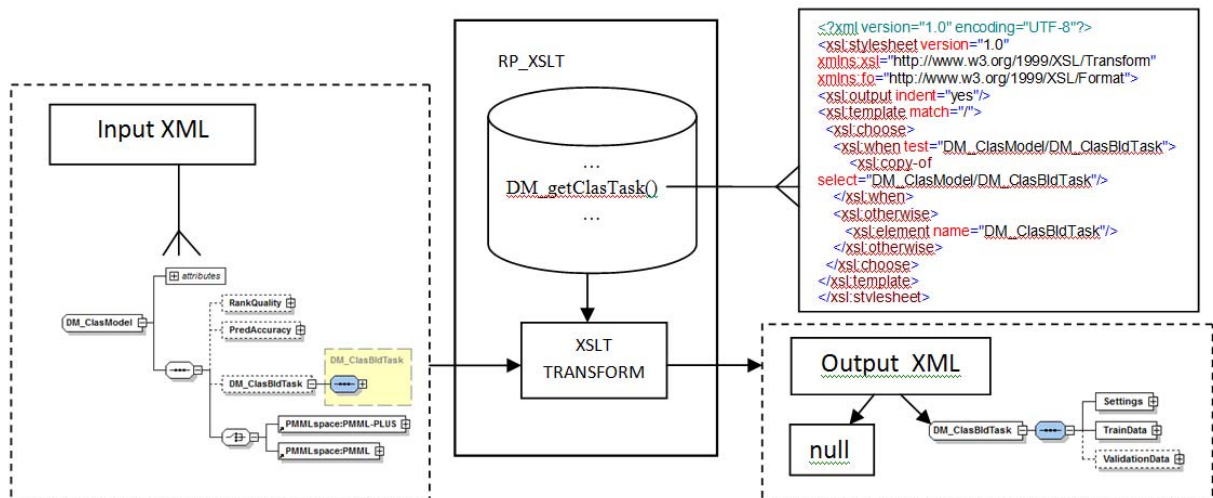


Рис. 1. Структурная схема XSLT преобразования в рамках SQL/MM стандарта

Структурная схема XSLT преобразования (рис. 1) определяет процесс XSLT преобразования в контексте стандарта SQL/MM. Состоит из трех блоков: входной XML-документ; функциональный блок XSLT преобразования; выходной XML-документ. Основным блоком данной схемы является функциональный блок и представляет из себя две составные части: системное хранилище, поддерживающее инфраструктуру аналитической подсистемы и XSLT преобразователь или XSLT-процессор. Выходной XML-документ определяется двумя возможными случаями, описанными в методе, и может возвращать значение null или *DM_ClasBldTask*.

Подобным образом реализуются все функции стандарта SQL/MM.

Список литературы

1. ISO/IEC 13249-6-2006, SQL/MM Part 6.
2. PMML Version 4.1, 2012, Data Mining Group (DMG) <http://www.dmg.org/>

APPS: ADAPTIVE PRIORITY BASED PACKET SCHEDULING METHODE IN IP

SEYED ENAYATALLAH ALAVI¹, MARJAN NADERAN-TAHAN²,
MOHAMAD AMINIAN³

¹*Shahid Chamran University of Ahvaz, Ahvaz, Iran; se_alavi@yahoo.co.uk*

²*Shahid Chamran University of Ahvaz, Ahvaz, Iran; m.naderan@scu.ac.ir*

³*Shahid Chamran University of Ahvaz, Ahvaz, Iran; m.aminiyan@gmail.com*

We propose adaptive scheduling paradigm that make it possible for IP routers to satisfy different requirements of packet flows. This method promotes conditionally small data packets, which require minor service times, with respect of the maximum delay they can handle that preventing expiration for packets, as long as their total service times cause insignificant delays to other packets in the queue.

Keywords: Artificial intelligence, IP Method, Packet scheduling.

Introduction—Traditional scheduling paradigms of IP networks do not match well the requirements of all categories of packets (for instance sensor packets). Applications which are belong to sensor networks, VoIP and etc. do not really cause significant delays. So naturally prioritized them practically will enhance the overall performance the network and satisfy the needs of this type of packets.

Our primary assumption is that applications like sensor applications generate packets in form of non-congestive traffic. This observation calls for a new metric for application fairness as well, which relies mainly on the delay rather than throughput. In order to avoid the cost of packet preparation for differentiated services, we take advantage of distinctive property of typical non-congestive data:

- *The small size of non-congestive packets*

The key idea of this approach is the service discipline called: “Less Impact Better Service” (LIBS).

II Related works

A lot has been done in the networking community aiming at controlling traffic based on its characteristics. Floyd and Fall introduced mechanisms based on the identification of high-bandwidth flows from the drop-history of RED. The RED-PD algorithm (RED with Preferential Dropping) uses per-flow preferential dropping mechanisms. Two other approaches that use per-flow preferential dropping with FIFO scheduling are Core-Stateless Fair Queuing (CSFQ) and Flow Random Early Detection (FRED). CSFQ marks packets with an estimate of their current sending rate. The router uses this information in conjunction with the flow’s fair share estimation in order to decide whether a packets needs to be dropped. FRED does maintain a state although only for the flows which have packets in the queue.

III. Adaptive priority based packet scheduling

First, we assume two different classes of packets(congestive and non-congestive). We use three priority queues. The queue with lower number has higher priority. When packet received by router first based on delay flag we decide to put it in first two high prior queues or the second two queues.

Second step is making decision based on size of packet. So we assume PL (packet length) as critical parameter in our work. Based on what we said the packet with smaller size

and lowest flow should receive better services, so we considering two threshold PL1=130 bytes and PL2 as average of length of receiving packets. And we define function PPL as first parameter to decision (if PL2 > PL1 we use (1) but if PL2 < PL1 we use (2)):

$$P_{PL} = \begin{cases} 1 & PL \leq PL1 \\ \frac{(PL - PL1)}{(PL2 - PL1)} & PL1 < PL < PL2 \\ 0 & PL \geq PL2 \end{cases} \quad (1)$$

Where PL is the packet length if any received pocket.

$$P_{PL} = \begin{cases} 1 & PL \leq PL1 \\ 0 & PL \geq PL1 \end{cases} \quad (2)$$

The second parameter is the priority probability. This probability will be calculated separately for packet that classified in first step and it is called HPP (high priority probability) and LPP (low priority probability):

$$HPP = \frac{\text{number packets in Queue 0}}{\text{number of all received delay sensitive packets}} \quad (3)$$

$$LPP = \frac{\text{number of all received delay sensitive packets}}{\text{number of all received delay sensitive packets}}$$

To continue the first step classification, we integrate two mentioned parameters PPL and HPP/LPP and introduce Final Probability (FP) as an average of the two other probabilities. We use average with same weight. So we define FP as follow :

$$FP = \begin{cases} \frac{(P_{PL} + HPP)}{2} & \text{for delay sensitive packets} \\ \frac{(P_{PL} + LPP)}{2} & \text{for non - delay sensitive packets} \end{cases} \quad (4)$$

At the end based on FP, we finalize classification of the received packet. After classification of packets, we need a scheduling method. We choose simple priority scheduling. This method service queues based of their priority.

IV Simulations and results

To test our proposed method we use OPNET 14 simulator. We use dumbbell network topology. We consider the number of delay sensitive flows to the 10 percent of the total flows, and we increased the number of flows in the simulation to examine the network's behavior. The simulations illustrate that our proposed method increases goodput and decrease the overall delay (Fig.1).

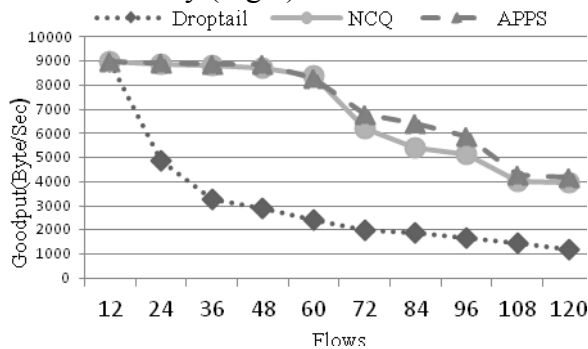


Figure1.a Goodput diagram

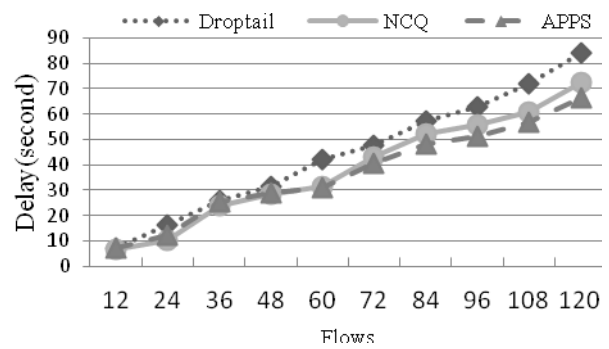


Figure1.b Average overall delay

V. Conclusion

In this work we demonstrate that APPS can be adjusted to promote service for low delay applications. For future works we suggest to do simulations on actual data packets of the applications. Also we suggest performing optimization on parameters using evolutionary algorithms to improve the performance of algorithms.

ПРОГРАММНЫЙ КОМПЛЕКС ДЛЯ РАЗРАБОТКИ И ТЕСТИРОВАНИЯ СИСТЕМ МАССОВОГО ОБСЛУЖИВАНИЯ

А.И. КЛИМЕНКО¹, В.С. БУРЧИК²

*Белорусский государственный университет информатики и радиоэлектроники
ул. П. Бровки, 6, г. Минск, 220013, Республика Беларусь
¹ak.igorevich@gmail.com; ²candentira@gmail.com*

Повсеместное использование систем массового обслуживания привело к возрастанию интереса в определении зависимости характеристик работы системы от её входных параметров (таких как интенсивность потока заявок и обрабатываемого потока, число каналов, закон распределения времени появления и обслуживания заявки). Это привело к идее создания программного комплекса, который позволяет создавать и моделировать работу систем массового обслуживания. Результаты моделирования можно использовать для оптимизации работы системы.

Ключевые слова: система массового обслуживания, имитационное моделирование.

Системой массового обслуживания (СМО) называется система, предназначенная для обслуживания заявок (требований), поступающих на нее в случайные моменты времени. В качестве процесса обслуживания могут быть представлены различные по своей физической природе процессы функционирования экономических, производственных, технических и других систем. В качестве реально действующих примеров систем массового обслуживания можно привести любые магазины, банки, почтовые отделения, ремонтные автомастерские, аэропорты и любые другие системы, работу которых можно свести к обработке поступающих в случайное время заявок. Следует отметить характерную черту работы таких объектов – случайное появление заявок (требований) на обслуживание и завершение обслуживания в случайные моменты времени, т. е. стохастический характер процесса их функционирования.

Задачи теории массового обслуживания – нахождение вероятностей различных состояний СМО, а также установление зависимости между заданными параметрами (числом каналов, интенсивностью потока заявок и интенсивностью потока обработки заявок, распределением времени появления заявки и её обслуживания, дисциплиной обслуживания заявок в канале и др.) и характеристиками эффективности работы СМО. В качестве важных характеристик работы системы могут рассматриваться, например, следующие:

- абсолютная пропускная способность СМО (среднее число заявок, обслуживаемое СМО в единицу времени);
- относительная пропускная способность СМО (вероятность обслуживания поступившей заявки);
- вероятность отказа, т.е. вероятность того, что поступившая заявка не будет обслужена, получит отказ;
- среднее время ожидания в очереди;
- средняя длина очереди;
- количество работающих и простаивающих каналов в единицу времени;

В виду повсеместного распространения систем массового обслуживания, а также исследовательского интереса к данной теме, был создан программный комплекс, имитирующий работу СМО с заданными параметрами (рис. 1).

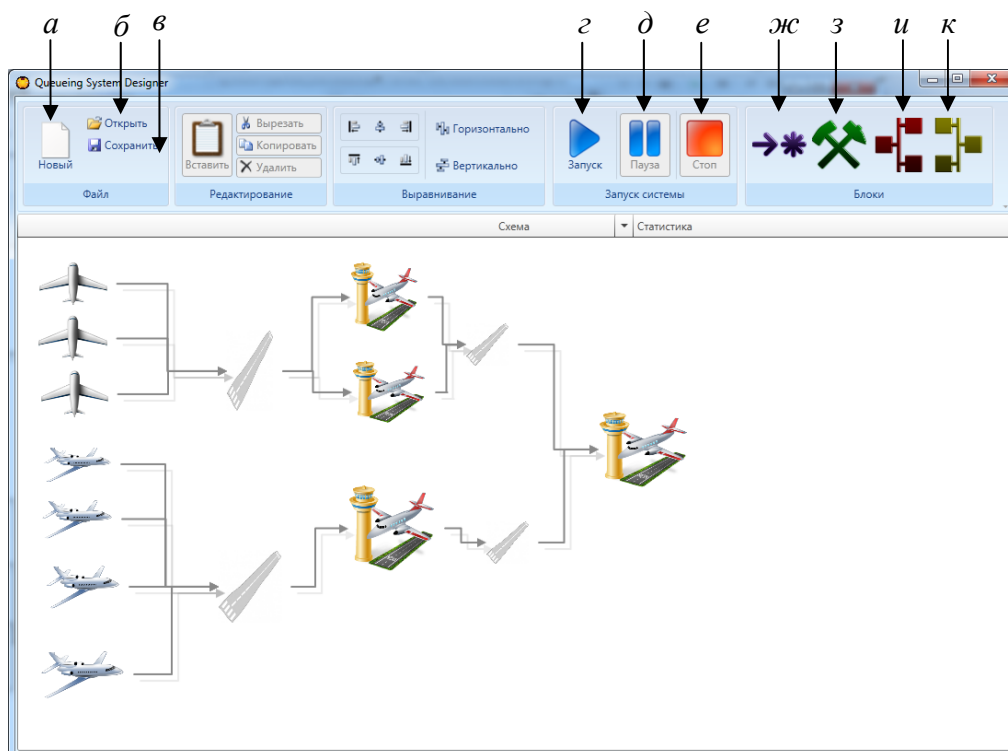


Рис. 1. Пользовательский интерфейс комплекса и пример созданной модели:
а – создание новой модели; *б* – открытие сохраненной модели; *в* – сохранение текущей модели;
г – запуск созданной модели; *д* – приостановка работы запущенной модели;
е – остановка работы текущей модели; *ж* – блок генерации заявок;
з – блок обработки заявок; *и* – блок, разделяющий входящий поток заявок на несколько;
к – блок, суммирующий входящие потоки заявок;

Основными функциями данного комплекса является моделирование работы системы массового обслуживания. В рамках программного комплекса возможно: создание потока заявок с требуемыми вероятностными характеристиками, создание обслуживающего устройства и задание параметров его работы, проверка работы СМО, получение статистических результатов работы смоделированной СМО (вероятности обслуживания заявки и вероятности отказа заявки, абсолютной пропускной способности системы, среднего времени ожидания, обслуживания и пребывания заявки в системе, а также графики вероятности отказа, занятости каналов и средней длины очереди во времени). В качестве основных структурных компонентов можно выделить: блоки генераторов заявок (рис. 1, *ж*), блоки каналов обслуживания (рис. 1, *з*), блоки суммирования и разделения заявок (сумматоры (рис. 1, *и*) и сплитеры (рис. 1, *к*)).

Список литературы

1. *Волорова Н.А.* Лекции по дисциплине курса Имитационное и статистическое моделирование. Минск, 2011.
2. *Вентцель Е.С.* Теория вероятностей. Москва, 1969.
3. *Гмурман В.Е.* Теория вероятностей и математическая статистика. Москва, 2003.

ПАРАЛЛЕЛЬНАЯ ОБРАБОТКА ЗАПРОСОВ СИСТЕМ УПРАВЛЕНИЯ ИДЕНТИФИКАЦИОННЫМИ ДАННЫМИ

П.П. ПИНЮТА¹, В.В. БАХТИЗИН²

Белорусский государственный университет информатики и радиоэлектроники

ул. П. Бровки, 6, г. Минск, 220013, Республика Беларусь

¹pinyuta-p@yandex.ru; ²bww@bsuir.by

Рассмотрен ряд методов планирования. Произведён их сравнительный анализ с точки зрения адаптации к системам управления идентификационными данными. Разработан алгоритм распределения задач на базе кооперационного метода планирования. Приведены результаты экспериментального исследования.

Ключевые слова: идентификационные данные, кооперационный метод планирования, распределения задач.

Одним из главных элементов информационной инфраструктуры служит Система управления идентификационными данными (СУИД). Она обеспечивает ролевое управление доступом к ресурсам на основе единой политики безопасности и должностных обязанностей сотрудника. Безопасность любого крупного бизнеса опирается на централизованное управление идентификационными данными совместно с корпоративной стратегией безопасности, обеспечением защиты корпоративных сетей и межсетевых взаимодействий.

Внедрение на предприятие СУИД позволяет в значительной степени минимизировать перечисленные риски. С ростом числа корпоративных приложений некоторые задачи СУИД способны заблокировать работу целевой системы на некоторое время. Под целевой системой следует понимать систему, дающую авторизованному пользователю определённые права и привилегии. Так же целевая система делегирует СУИД механизмы управления идентификационными данными пользователей. Процесс передачи этих механизмов называется интеграцией целевой системы в СУИД. Таким образом, при использовании современных СУИД риск простоя оборудования и персонала остаётся высоким. Для его уменьшения необходимо разработать метод планирования запросов к целевой системе.

Каждый запрос, пришедший от пользователя, делится на две категории: краткосрочный и долгосрочный. Под краткосрочным понимается такой запрос, время выполнения которого незначительно, а приоритет выполнения низкий. Примером могут служить следующие типы запросы: добавление, удаление, блокирование учётной записи пользователя, выделение ресурсов пользователю и т.п. Приоритет долгосрочного запроса выше. Он занимает больше ресурсного времени целевой системы. Это связано с тем, что для обработки запроса требуется опрос большого количества целевых систем со значительным количеством пользователей. Примерами таких запросов могут являться аудит целевой системы, генерация различных отчётов и другое.

В настоящее время планирование широко применяется в различных областях. Для операционных систем все разработанные методы планирования основываются на следующих трёх алгоритмах: невытесняющий, вытесняющий, кооперационный. Они изначально разрабатывались с целью автоматического управления процессами в компьютерах. Рассмотрим возможность адаптации данных алгоритмов к СУИД.

Суть работы невывесняющего алгоритма планирования заключается в следующем. Один процесс загружается на выполнение. Ресурс освобождается только после полного выполнения данного процесса. В контексте СУИД это видится как недостаток, так как запросы часто требуют согласования с пользователями. Следовательно, все остальные запросы из очереди будут вынуждены ожидать, и общая скорость работы снизится.

Альтернативным алгоритмом планирования является вытесняющий алгоритм. При его использовании операционная система сама передает управление от одной выполняемой программы другой в случае появления некоторых событий. Отдельно стоит отметить, что время передачи ресурсного времени от одного процесса другому не зависит от состояния задачи, что означает возможность её прерывания в любое время. При попытке адаптации к СУИД было отмечено, что указанная особенность рассматриваемого алгоритма крайне негативно влияет на работу с целевой системой. При поступлении нового запроса выполняющийся может быть прерван. А это противоречит требованию непрерывности работы долгосрочных запросов.

Основное отличие кооперационного алгоритма от предыдущего заключается в том, что следующая задача выполняется только после того, как текущая явно объявит себя готовой отдать ресурсное время. Таким образом, следующая задача из очереди выполняется только после того, как текущая либо завершится, либо приостановится в связи с синхронизацией с каким-либо объектом. Применение данного алгоритма в классическом виде связано с монополизацией процессора, что, в общем, является недостатком. Однако с точки зрения СУИД это становится достоинством. Это связано с тем, что управляющий процесс блокирует работу всех остальных запросов к целевой системе. Таким образом, удовлетворяется требование о непрерывности работы долгосрочных запросов. Для создания метода управления запросами СУИД на основе кооперационного алгоритма, уменьшающего риски простоя оборудования и персонала, необходимо ввести некоторые дополнительные требования. В частности одним из таких требований является уменьшение количества блокировок во время выполнения долгосрочных запросов.

Для реализации указанного требования время запуска долгосрочных запросов необходимо смещать. При этом оно должно определяться наименьшим количеством заблокированных задач за время от начала до завершения работы долгосрочного запроса. Это значит, что время запуска должно быть выбрано такое, при котором число кратковременных запросов, запущенных ранее в аналогичный период, минимально. Для этих целей предлагается алгоритм подсчёта наилучшего времени запуска длительных задач на основе истории запросов к целевой системе.

Внедрение в СУИД разработанного метода планирования задач положительно влияют на работу системы. Пиковые значения функции количества заблокированных запросов резко уменьшаются. Это способствует уменьшению времени ожидания запрошенного сотрудником ресурса. Данное обстоятельство увеличивает производительность труда на предприятии, так как персоналу больше не требуется ожидать окончания работы длительных запросов. Также следует отметить и другое достоинство применения предложенного метода. Оно заключается в снижении уровня загрузки целевой системы, что облегчает пользователям работу с ней.

УПРАВЛЕНИЕ КАЧЕСТВОМ МОБИЛЬНЫХ ПРИЛОЖЕНИЙ

А.А. ШЕЛКОВИЧ¹, В.В. БАХТИЗИН

*Белорусский государственный университет информатики и радиоэлектроники
ул. П. Бровки, 6, г. Минск, 220013, Республика Беларусь*

¹*alexey@shelkovich.com*

Качество современного мобильного приложения является одной из его важнейших характеристик, которая относится к способности удовлетворять установленные или предполагаемые потребности. Создание качественного мобильного приложения невозможно без построения системы управления качеством и следования соответствующим международным стандартам. В этой связи большой интерес представляют исследования о области контроля, обеспечения и оценки качества мобильных приложений.

Ключевые слова: мобильные приложения, управление качеством, модель качества.

Развитие и распространение мобильных технологий позволило обеспечить доступ к информационным ресурсам повсеместно и в любой момент времени, а техническое совершенствование мобильных устройств сделало возможным их применение для решения широкого круга деловых, практических и ежедневных задач. Развитие глобального рынка мобильных приложений привело к появлению огромного количества продуктов, значительная часть из которых предназначена для решения сходного круга задач и обладает похожей функциональностью. В данных условиях качество мобильного приложения является не только сферой теоретического интереса, но так же определяет возможность практического применения приложения в той или иной ситуации, обеспечивает конкурентоспособность и успех продукта на рынке.

Процессы управления качеством проектов по разработке мобильных приложений должны охватывать все операции, осуществляемые разработчиком с целью определения политики, целей и ответственности в области качества для обеспечения соответствия проекта и программных продуктов предъявляемым к ним требованиям. Управление качеством осуществляется посредством системы, которая предусматривает соответствующие правила, процедуры и процессы, позволяющие планировать, обеспечивать, контролировать и совершенствовать качество. Основная цель системы управления качеством – создание условий для постоянного улучшения каждого из производственных процессов, взаимное взаимодействие которых приводит к более совершенной системе и, как следствие, и производству более качественных мобильных приложений. Пример взаимодействия процессов управления качеством проекта показан на рис. 1. [1]

Управление качеством проекта направлено как на управление качественными состояниями самого проекта, так и на качество результата проекта – мобильного приложения. Можно выделить три основных процесса, которые лежат в основе системы управления качеством:

- процесс планирования качества: определение требований и/или стандартов качества, относящиеся к проекту и продукту; документирование, каким образом будет продемонстрировано достигнутое им соответствие;
- процесс обеспечения качества: проверка соблюдения требований к качеству и результатов измерений в процессе контроля качества для обеспечения использования советующих стандартов качества и мер качества;

- процесс контроля качества – мониторинг контрольных точек процессов разработки, направленный на обеспечение качества, оценку исполнения и выработку рекомендаций относительно необходимых изменений.



Рис. 1. Управление качеством проекта

Увеличение сложности и размера современных мобильных приложений и одновременный рост ответственности и важности выполняемых ими функций отразились на предъявляемых требованиях в области качества и безопасности их использования со стороны заказчиков и пользователей. Одной из основных проблем обеспечения качества программных средств является формализация характеристик качества и методология их оценки. Существует ряд современных инструментов и технологий, позволяющих добиться высокого уровня организации процесса разработки приложений, однако для обеспечения высокой эффективности и качества функционирования программного обеспечения, адекватности процесса оценки качества, создания возможностей для дальнейшего совершенствования и развития приложения необходимо применение действующих международных стандартов.

Актуальным стандартом в области качества является серия стандартов ISO/IEC 25000: Software product Quality Requirements and Evaluation (SQuaRE). Стандарт ISO/IEC 25010 [2] определяет две трёхуровневые иерархические модели качества – модель качества программной продукции и модель качества в использовании. Данные модели служат основой для построения модели качества приложений мобильных устройств, которая учитывает особенности данных приложений при их разработке и использовании.

Оптимальный выбор модели качества, следование актуальным международным стандартам, постоянный контроль и обеспечение качества разработки лежат в основе успешной реализации мобильного приложения, в полной мере отвечающего предъявляемым к нему современным требованиям.

Список литературы

1. A Guide to the Project Management Body of Knowledge – PMBok Guide, Fifth edition. – Project Management Institute, 2013. – 590.
2. ISO/IEC 25010: Systems and software engineering – Systems and software product Quality Requirements and Evaluation (SQuaRE) – System and software quality models. – Женева: ISO/IEC, 2011. – 34 с.

РЕКУРСИВНЫЙ АЛГОРИТМ СРАВНЕНИЯ ДЕРЕВЬЕВ

А.В. ЛЫЧКОВСКИЙ

Белорусский государственный университет информатики и радиоэлектроники
ул. Л.Беды, 2(б), г. Минск, 220040, Республика Беларусь
mkdnmail@gmail.com

Задача сравнения моделей данных широко исследуется в информатике. Большой практический интерес имеют задачи связанные с данными представленными деревьями. Одной из задач является сравнение деревьев. Данная работа дает представление о подходе к сравнению и описывает рекурсивный алгоритм решающий такую задачу.

Ключевые слова: дерево, сравнение деревьев, рекурсивный алгоритм.

Деревья являются одной из хорошо изученных структур данных в информатике. На практике задача сравнения деревьев возникает в таких областях, как вычислительная биология, структурированные базы данных текста, анализ изображений, оптимизация компиляторов.

Дерево – узел (называемый корневым) соединенный с упорядоченной последовательностью несвязных деревьев[1]. Такая последовательность называется **лесом**. Под обозначением $l(A_1 \circ \dots \circ A_n)$ будем понимать дерево, составленное из узла l , соединенного с последовательностью деревьев $A_1 \circ \dots \circ A_n$. Это определение предполагает, что деревья являются упорядоченными, а вершины помеченными. Дерево называется помеченным, если определен порядок среди узлов дерева. Для вычисления редакционного расстояния на дереве определяются следующие операции:

- **удаление** узла и соединение его детей к предку узла с сохранением порядка;
- **вставка** узла между существующими узлом и подпоследовательностью упорядоченных детей узла;
- **переименование** метки узла.

Пусть F и G два леса. Редакционное расстояние между F и G , обозначаемое $d(F, G)$, минимальная стоимость операций редактирования необходимых для преобразования F в G . Используя рекуррентный подход, редакционное расстояние может быть представлено следующими формулами:

$$d(l(F) \circ T, l'(F') \circ T') = \min \begin{cases} c_d(l) + d(F \circ T, l'(F') \circ T') \\ c_i(l') + d(l(F) \circ T, F' \circ T') \\ d(l(F), l'(F')) + d(T, T') \end{cases} \quad (1)$$

или

$$d(T \circ l(F), T' \circ l'(F')) = \min \begin{cases} c_d(l) + d(T \circ F, T' \circ l'(F')) \\ c_i(l') + d(T \circ l(F), T' \circ F') \\ d(l(F), l'(F')) + d(T, T') \end{cases} \quad (2)$$

Разбиение согласно (1) называется левым, согласно (2) правым. Разные разбиения ведут к разному количеству рекурсивных вызовов.

Прямые попытки реализации рекурсивного алгоритма имеют экспоненциальную сложность. Современные алгоритмы динамического программирования имеют полиномиальное время выполнения. Решение подзадач сохраняется и используется повторно. Они используют стратегии разбиения для выбора левого или правого на каждом шаге.

Выбор между левым и правым рекурсивным решением может быть определен через **декомпозиции путей**[2]. Они используют корень-лист пути в дереве, для декомпозиции его на поддеревья и подлеса. Левый путь, правый путь и тяжелый путь рекурсивно соединяют предка с его крайним левым ребенком, крайним правым ребенком, с ребенком который определяют наибольшее поддерево соответственно. **Релевантные поддеревья** дерева для некоторого корень-лист пути – все поддеревья, которые получаются удалением пути из дерева. **Релевантные подлеса** дерева – это само дерево и все его поддеревья полученные удалением узлов в следующем порядке: удалить корень и остановиться, если узлов не осталось; удалить крайний левый корневой узел в результирующем дереве пока крайний левый корневой узел находится на корень-лист пути; удалить крайний корневой узел пока крайний правый корневой узел находится на корень-лист пути; рекурсивно повторять процедуру для результирующего поддерева.

Для каждой задачи состоящей из двух релевантных поддеревьев мы должны выбрать путь для декомпозиции. Путь может быть выбран как в левом так и в правом дереве. Набор путей который мы выбираем определяет **стратегию путей**.

Концепция декомпозиции путей использована для вычисления редакционного расстояния в оптимальном размере памяти $O(mn)$ [3]. **Функция одного пути** вычисляет редакционное расстояние для дерева из двух релевантных поддеревьев в соответствии с выбранным путем. Вычисленные расстояния сохраняются и повторно используются.

На основании описанного выше общий алгоритм может быть сформулирован. Для двух деревьев F и G расстояние между ними может быть вычислено в два шага. Во первых надо выбрать путь в одном из них, например F , и вычислить расстояние между релевантными поддеревьями F и деревом G . Эти расстояния будут повторно использованы на следующем шаге. Во вторых, нужно вычислить расстояние между F и G от низа к верху, вычисляя расстояния между релевантными подлесами F и всеми соответствующими подлесами G . На этом шаге мы не вычисляем заново уже полученные расстояния с первого этапа.

Общий алгоритм редакционного расстояния деревьев использует стратегию путей для его вычисления. Алгоритм состоит из следующих шагов:

- для данной пары деревьев F, G находит путь используя стратегию путей;
- если путь в F выполняет следующие шаги, иначе выполняет алгоритм для G, F : выполняет алгоритм для каждого релевантного поддерева F^i в F и дерева G , вычисляет функцию одного пути для F, G в соответствии с типом пути.

Список литературы

1. Computer Science Laboratory of Lille: Analysis of tree edit distance algorithms. [Электронный ресурс]. – Режим доступа: <http://www.lifl.fr/~touzet/Publications/cpm03.pdf>. – Дата доступа: 10.12.2013.
2. Tree Edit Distance: General Tree Edit Distance Algorithm. [Электронный ресурс]. – Режим доступа: <http://www.inf.unibz.it/dis/projects/tree-edit-distance/tree-edit-distance.php>. – Дата доступа: 10.12.2013.
3. K. Zhang, D. Shasha. Simple fast algorithms for the editing distance between trees and related problems. SIAM J. Comput. 1989.

МЕТОД КЛАССИФИКАЦИИ ТЕКСТОВ НА РУССКОМ ЯЗЫКЕ

Ф.И. ТРЕТЬЯКОВ, Л.В. СЕРЕБРЯНАЯ

*Белорусский государственный университет информатики и радиоэлектроники
ул. П. Бровки, 6, г. Минск, 220013, Республика Беларусь
Fiodor.Tretyakov@gmail.com, l_silver@mail.ru*

Метод классификации текстов напрямую зависит от языка, на котором написан текст. Разнообразие словоформ русского языка усложняет процедуру классификации. Необходимы способы, приводящие к единому виду все многообразие словоформ. Разработан алгоритм, определяющий корни лексем и выполняющий автоматическую классификацию текстов на их основе.

Ключевые слова: классификация, разделяющая функция, стеммер.

Огромные коллекции электронных документов требуют создания эффективных методов обработки данных. Одному из них — частотной классификации текстов посвящена данная работа.

Существует множество способов решить названную задачу. Прежде всего выбор подхода зависит от количества исходных данных. Если имеется набор текстов-образцов и категорий, то речь идет о контролируемом обучении и классификации. Затем необходимо определить решающее правило и разделяющую функцию, с помощью которых будет выполняться классификация «незнакомых» текстов. Они подставляются в качестве параметра в разделяющую функцию, в результате чего определяется их принадлежность к одному из классов. Одним из существенных факторов, влияющих на выбор класса, является язык, на котором написан текст [1]. Настоящая работа посвящена русскоязычным текстам.

В русском языке лексемы имеют сложные и разнообразные структуры, что существенно затрудняет процедуру классификации текстов. Однокоренные слова могут иметь различные окончания, суффиксы и приставки, которые не должны влиять на результат классификации. Однако при проверке формального совпадения однокоренных лексем и получении отрицательного результата сравнения классификация текстов, построенная на основе неточных результатов сравнения, оказывается неверной. Поэтому для анализа русского языка в качестве разделяющей необходимо выбрать функцию, оперирующую только частью слова и выдающую ответ на его основе.

Одним из способов выделения определенной части слова в русском языке является стемминг. Это процесс нахождения основы слова заданной лексемы. Основа не всегда совпадает с морфологическим корнем слова [2]. Задача нахождения основы слова представляет собой давнюю проблему в области компьютерных наук. Первая публикация на заданную тему датируется 1968 годом. Стемминг применяется в поисковых системах для расширения поискового запроса пользователя и является частью процесса нормализации текста. На сегодняшний день созданы различные реализации алгоритмов стемминга. Они применяются для решения различных задач интеллектуальной обработки текстовой информацией.

Для решения задачи классификации используется специальный алгоритм стемминга под названием — стеммер [2]. Он может выделять значимую часть слова (стем). Для русского языка это основа и (или) корень. Основа определяется проще, однако в последствии с ее помощью получается менее качественная классификация. Например,

корень одного слова может быть эквивалентен корню и суффиксу другого. Поэтому алгоритмы, опирающиеся на корень слова, считаются более точными.

Для качественного выделения корня слова одного стеммера оказывается недостаточно. Для работы с русским языком можно использовать два дополнительных модуля грамматического словаря: лемматизатор и флексер (склонение и спряжение). С помощью лемматизатора слова приводят к базовой форме, что выполняется после обработки лексемы стемом. Флексер умеет выдавать все грамматические формы слова на основе базовой. Это позволяет улучшить результат, проверяя найденные фрагменты по набору форм ключевого слова.

Среди всех реализаций стеммеров можно выделить два типа:

- использующие словарь для выделения части слова;
- использующие эвристическую модель [2].

Для выделения корня слова был разработан программный модуль, включающий в себя стеммер, флексер и лемматизатор. Стеммер использует эвристическую модель. Рассмотрим алгоритм классификации на основе созданного модуля.

1. Обработать название всех категорий с помощью модуля, выделив корни слов и поместив результаты в соответствующий словарь. Каждая строка в нем имеет ключ, которым является корень слова, а значение в строке — количество всех словоформ по ключу из названия категории.

2. Выполнить шаг 1 для всех текстов, применив его не к названиям текстов, а к ним самим.

3. Найти для каждого текста наиболее подходящую категорию. Ее номер определяется значения переменной T , вычисленной по следующей формуле:

$$T = \sum_{\substack{i < n, \\ j < m, \\ i=0, \\ j=0}} a_i * b_j,$$

где n — размер словаря категории,
 m — размер словаря текста,
 a_i — слово из словаря категории,
 b_k — о слово из словаря текста.

4. Выбрать для текста категорию, где T максимально.

С помощью стеммера, флексера и лемматизатора можно классифицировать тексты с высокой точностью. Минусом является сложность архитектуры модуля.

Список литературы

1. TextMining. Глубинный анализ текста. Из цикла лекций «Современные Internet-технологии» для студентов 5-го курса кафедры Компьютерных технологий физического факультета Донецкого национального университета. ДонНУ, кафедра КТ, проф. В. К. Толстых.

2. Третьяков, Ф.И. Методы обработки текстовой информации [Текст] / Ф.И. Третьяков, Л. В. Серебряная // VI международная научно-практическая конференция «Актуальные вопросы методики преподавания математики и информатики». — Биробиджан, 2011. — С. 175–181.

АЛГОРИТМ ТЕКСТУРНОГО АНАЛИЗА ИЗОБРАЖЕНИЙ

М.М. ЛУКАШЕВИЧ

*Белорусский государственный университет информатики и радиоэлектроники
ул. П. Бровки, 6, г. Минск, 220013, Республика Беларусь
lukashevich@bsuir.by*

Предложен алгоритм вычисления текстурных признаков, отличающийся простой и эффективной схемой. Приведены результаты экспериментов на тестовых наборах из базы изображений текстур.

Ключевые слова: обработка изображений, текстурный анализ, текстурные признаки.

Анализ текстуры – фундаментальная проблема в обработке изображений, машинном зрении и его использовании при распознавании объектов. Большинство реальных объектов состоят из различных видов поверхностей текстуры, практически на любых изображениях можно выделить области, относящиеся к текстурным [1]. Поэтому текстурный анализ играет важную роль в обработке и распознавании изображений, таких как аэрофотоснимки, медицинские изображения, спутниковые изображения и т.д. Однако, несмотря на серьезные исследования в этой области, в настоящее время не создано теоретических основ и алгоритмических решений в области обработки текстурных изображений, которые до конца удовлетворяли бы требованиям практики [2].

Одной из задач исследований в данной области была задача разработки алгоритма вычисления набора текстурных признаков, отличающегося вычислительной простотой и не уступающего по эффективности существующим алгоритмам. Был предложен алгоритм, основанный на идеях [3] и предполагающий вычисление текстурных признаков на основе локальных бинарных масок. Для вычисления текстурных признаков используются маски размером 3×3 , 5×5 и 7×7 – всего $N = 11$ масок. Рассматриваются лишь те пиксели, которые находятся под серыми клетками маски («значащие пиксели»). На рис. 1 приведена графическая форма представления локальных масок.

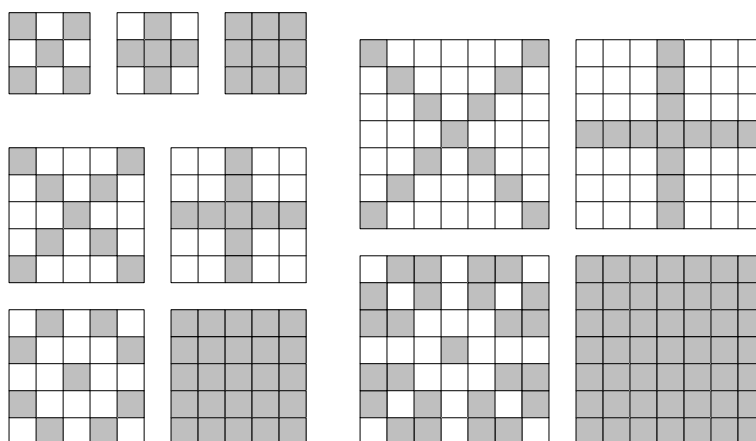


Рис. 1. Локальные бинарные маски

Только «значащие пиксели» (при наложении маски на изображение значение элемента маски равно 1) берутся в расчет для последующего вычисления текстурных признаков, соответствующих пикселю под центром маски. Функцию обработки изо-

бражения локальными бинарными масками в пространственной области можно записать в виде

$$G(x, y) = \frac{1}{N} \sum_{k=1, N} F_k [I(x, y)],$$

где $I(x, y)$ - входное изображение; k - число локальных бинарных масок; $G(x, y)$ - массив текстурных признаков; $N = 11$ - число локальных бинарных масок.

Текстурные признаки рассчитываются по соответствующим формулам:

$$G^{\min}(x, y) = \frac{\min}{k=1, N} F_k [I(x, y)]$$

$$G^{\max}(x, y) = \frac{\max}{k=1, N} F_k [I(x, y)]$$

$$G^{\text{median}}(x, y) = \frac{\text{median}}{k=1, N} F_k [I(x, y)]$$

Исследование эффективности алгоритма проводилось на базе изображений текстур Outex [4]. Эффективность алгоритма сравнивалась с алгоритмами, которые схожи по идеологии с предложенным алгоритмом (LBP^{riu2}, LBP^{riu2}/VAR, LBP-HF, VZ_MR8, VZ_Ptch). Для классификации использовался метод k ближайших соседей. Эксперименты выполнялись для двух тестовых наборов из базы Outex: Outex_TC_00010 (TC10) и Outex_TC_00012 (TC12), табл. 1.

Табл. 1. Точность классификации для TC10 и TC12, %

	TC10	TC12
LBP ^{riu2}	97,19	86,67
LBP ^{riu2} /VAR	98,33	87,01
LBP-HF	96,59	92,52
VZ_MR8	87,75	87,49
VZ_Patch	94,11	92,64
предложенный алгоритм	98,64	93,01

При экспериментальном исследовании алгоритма вычисления текстурных признаков на основе локальных бинарных масок было определено, что алгоритм инвариантен к повороту и показывает точность распознавания 98,64% на тестовом наборе из базы текстур Outex. Также алгоритм показывает робастность к незначительным вариациям угла обзора и освещенности. Достичь максимальной точности распознавания при данных условиях не удалось, но алгоритм показывает эффективность сравнимую с аналогичными алгоритмами.

Список литературы

1. *Gonzalez, R.C.* Digital image processing / R.C. Gonzalez, R.C. Woods. – Upper Saddle River : Prentice Hall, 2008. – 954 p.
2. *Handbook of texture analysis / ed. by M. Mirmehdi, X. Xie, J.S. Suri.* – London : Imp. College Press, 2008. – X, 413 p.
3. *Mäenpää, T.* Texture analysis with local binary patterns / T. Mäenpää, M. Pietikäinen // *Handbook of pattern recognition and computer vision / ed. by C.H. Chen; P.S-P Wang.* – 3rd ed. – River Edge, 2005. – P. 197–216.
4. *Outex is a framework for empirical evaluation of texture classification and segmentation algorithms [Electronic resource] / Univ. of Oulu.* – Mode of access : www.outex oulu.fi. – Date of access : 30.01.2014.

ЛОГИКО-ВЕРОЯТНОСТНЫЕ ПОДХОДЫ В АВТОМАТИЗАЦИИ МОДЕЛИРОВАНИЯ СТРУКТУРНО-СЛОЖНЫХ СИСТЕМ

Н.В. ЛАПИЦКАЯ

*Белорусский государственный университет информатики и радиоэлектроники
ул. П. Бровки, 6, г. Минск, 220013, Республика Беларусь
lapan@bsuir.by*

Предложены алгоритмы детерминированного моделирования в логико-вероятностном методе (ОЛВМ), теории и технологии автоматизированного структурно-логического моделирования (АСМ). Рассматриваются подходы, позволяющие повысить оперативность принятия решений при управлении структурами катастрофоустойчивых информационных систем (КАИС) за счет предварительной структуризации информации.

Ключевые слова: логико-вероятностных методов, вероятностные модели, схема функциональной целостности.

Логико-вероятностные методы системного анализа имеют четко выраженные детерминированные составляющие на всех основных этапах моделирования [1].

На этапе постановки задач к детерминированным относятся все виды графических средств и методики построения структурных моделей исследуемых свойств (деревья отказов, событий, графы связности блок схемы, схемы функциональной целостности и др.). На промежуточных этапах детерминированными являются методы, алгоритмы и программы построения на основе заданной структурной схемы логических и вероятностных математических моделей исследуемых свойств системы. На завершающем этапе ОЛВМ детерминированными выступают методы и процедуры вычислений вероятностных показателей свойств систем, на основе построенных точных или приближенных аналитических вероятностных функциях. В ОЛВМ для построения моделей используется универсальный графический аппарат структурных схем функциональной целостности (СФЦ). [1, 2, 3].

Детерминированные функции работоспособности системы и вероятностные функции определяются в ОЛВМ для всех видов монотонных и немонотонных моделей исследуемых свойств систем большой размерности и высокой структурной сложности. Для построения используется универсальный графоаналитический метод. В статическом ЛВМ детерминированными являются средства построения имитационных моделей исследуемых свойств структурно-сложных систем. На основе сформированных имитационных моделей методами статистических испытаний определяются количественные оценки вероятностных показателей исследуемых свойств системы. Исследование и решение задач повышения катастрофоустойчивости реализации целевых процессов и информационных систем, их обеспечивающих, осуществляется в рамках междисциплинарного подхода, представляющего задачи управления структурной динамикой [4]. Одним из этапов решения задач анализа и синтеза КАИС является предварительная кластеризация возможных многоструктурных состояний указанных систем и выделение типовых (эталонных) структур кластеров. Формально постановка задачи кластеризации множества многоструктурных состояний КАИС сводится к следующему: известно множество многоструктурных состояний $S = \{S_1, \dots, S_r\}$, каждое многоструктурное состояние характеризуется конечной совокупностью показателей $F = \{F_1, \dots, F_m\}$, с помощью которых оцениваются различные аспекты структур функционирования КАИС.

На рис. 1 представлена структура сложной системы, содержащая эквивалентированные вершины, полученная в результате автоматизированного моделирования с использованием ПК «Арбитр».

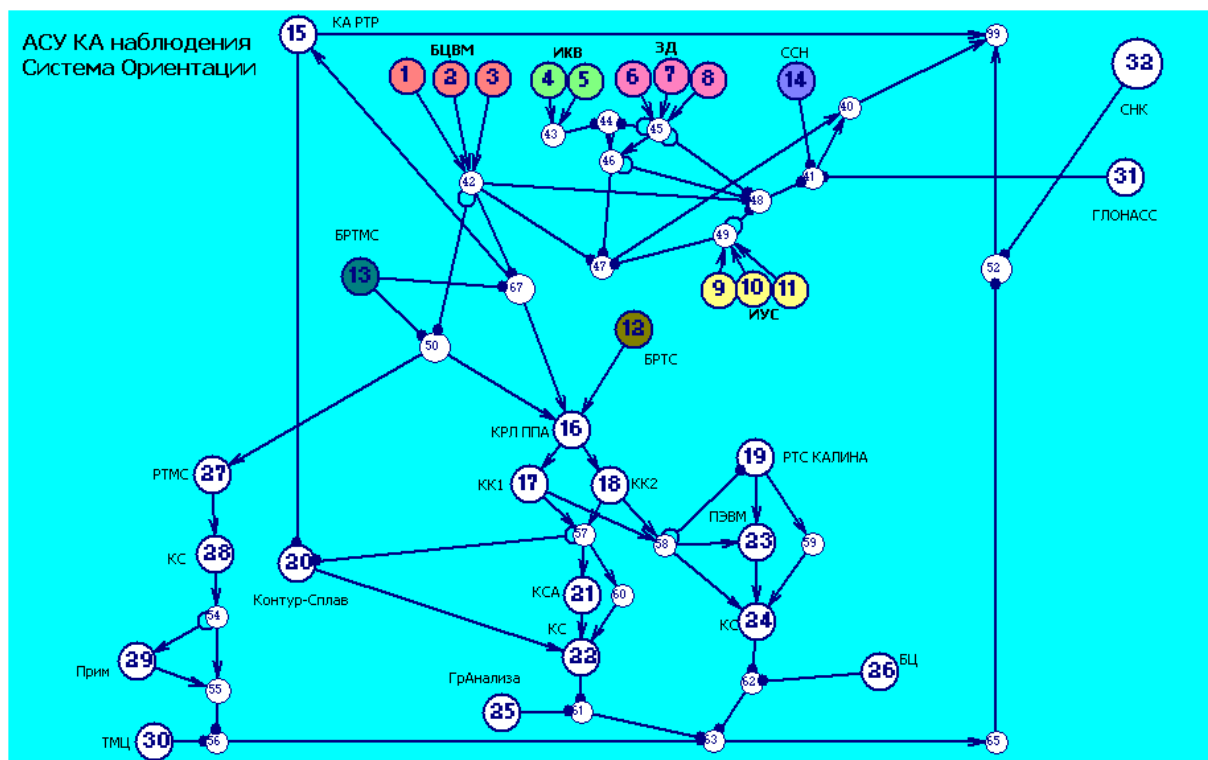


Рис. 1. Описание структуры сложной системы, содержащей эквивалентированные вершины

Список литературы

1. *Гладкова, И.А.* Детерминированные разделы общего логико-вероятностного метода / И.А. Гладкова // Труды международной научной школы «Моделирование и анализ безопасности и риска в сложных системах». МА БР – 2010. СПб.: Издательство «Бизнес-Пресса», 2010. – С. 453–460.
2. *Можаяев, А.С.* Общий логико-вероятностный метод автоматизированного структурно-логического моделирования надежности, безопасности и риска сложных систем. // Многотомное издание «Безопасность России». Анализ риска и проблем безопасности. В 4-х частях. – М.: МГФ «Знание», 2006. – 640 с. (С. 153–197.)
3. АРБИТР, «Программный комплекс автоматизированного структурно-логического моделирования и расчета надежности и безопасности систем (ПК АСМ СЗМА), базовая версия 1.0». Автор Можаяев А.С. Правообладатель ОАО «СПИК СЗМА».
4. *Лапицкая, Н.В.* Комбинированные методы классификации рисков чрезвычайных ситуаций и многоструктурных макросостояний в информационных системах / Д.А. Вятчин, Н.В. Лапицкая, О.И. Семенов, Б.В. Соколов, А.Н. Павлов // Труды международной научной школы «Моделирование и анализ безопасности и риска в сложных системах». МА БР – 2011, Санкт-Петербург, 28 июня – 2 июля, 2011) / СПб.: ГУАП, СПб., 2011 – С. 300–305.

ТЕСТИРОВАНИЕ ПРОИЗВОДИТЕЛЬНОСТИ ВЕБ-ОРИЕНТИРОВАННЫХ ПРИЛОЖЕНИЙ

Ю.П. КУРМАЗ¹, С.С. КУЛИКОВ²

*Белорусский государственный университет информатики и радиоэлектроники
ул. П. Бровки, 6, г. Минск, 220013, Республика Беларусь
¹yurytmxyzptlk@gmail.com; ²kulikov@bsuir.by*

Обеспечение производительности и надежности веб-приложений является очень важным направлением в изучении и формализации. В связи с этим большой интерес представляет подход к тестированию веб-приложений, основанный на построении математических моделей. Разработка программных средств автоматизированного тестирования веб-ориентированных приложений, использующих модельный подход, является актуальной задачей.

Ключевые слова: тестирование, производительность, модели тестирования, автоматизация, программное средство, веб-приложения.

На протяжении последних десятилетий наблюдается все более глубокое проникновение самых разнообразных аспектов человеческой деятельности во всемирную сеть. Если на заре развития Интернет веб-технологии использовались для простых сайтов со статическим содержимым, то сейчас многие веб-приложения являются сложными вычислительными системами и системами, предоставляющими динамически генерируемое и медиа содержимое. С возрастанием сложности приложений возрастают и требования к обеспечению качества программного обеспечения. Обеспечением качества программных продуктов занимается такой процесс, как тестирование программного обеспечения. Тестирование, которое проводится с целью обеспечения надежности и определения того, как ведет себя система под различной нагрузкой, называется тестированием производительности программного обеспечения.

Большинство организаций, занимающихся разработкой веб-ориентированного программного обеспечения, обязательно включают в цикл разработки фазу тестирования. Они содержат штат специалистов по обеспечению качества. Но, с ростом сложности приложений, возрастает и количество специалистов, необходимых для тестирования продукта. А для тестирования производительности, например для нагрузочного тестирования, использование «живых» специалистов просто невозможно. В таких случаях процессу тестирования необходима автоматизация.

Автоматизированное тестирование программного обеспечения – часть процесса тестирования на этапе контроля качества в процессе разработки программного обеспечения. Оно характеризуется использованием программных средств для выполнения тестов и проверки результатов выполнения. Автоматизация тестирования позволяет значительно сократить расходы ресурсов (временных и человеческих) и упростить процесс тестирования.

Для обеспечения качественного автоматизированного тестирования, необходимо использовать подход к тестированию на основе построения моделей. При таком подходе необходимо построить математическую модель системы, затем протестировать приложение по построенной модели.

В тестировании производительности веб-приложений такой моделью является модель виртуального пользователя. Для наибольшего подобия реальному пользователю, такая модель должна быть основана на марковских цепях.

Марковская цепь — последовательность случайных событий с конечным или счётным числом исходов, характеризующаяся тем свойством, что, говоря нестрого, при фиксированном настоящем будущее независимо от прошлого. На рисунке 1 представлен пример графа переходов марковского процесса.

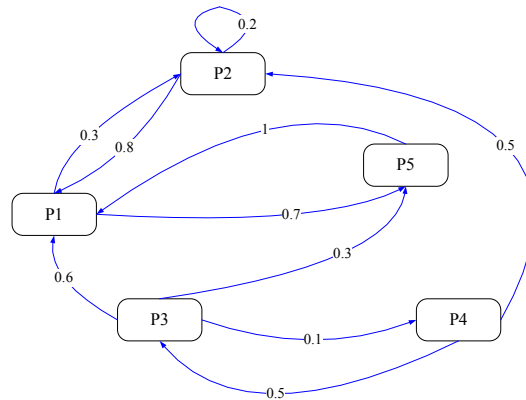


Рис. 1. Пример графа переходов для марковского процесса

Если за множество состояний принять операции (атаки), совершаемые имитируемым пользователем, а за множество переходных вероятностей примем вероятности, с которыми пользователь переходит от одного действия к другому, то данная модель подходит для реализации модели виртуального пользователя тестирования производительности. На рисунке 2 приведен пример сценария поведения пользователя, основанный на марковской цепи.

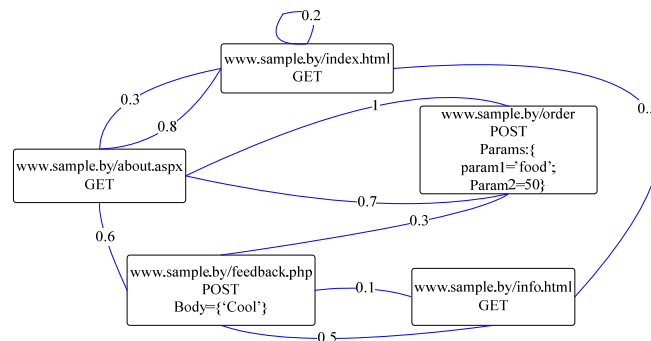


Рис. 2. Пример сценария поведения пользователя, основанного на марковской цепи

Подход к тестированию, основанный на построении моделей, требует специальных знаний специалистов по тестированию и привлечение к разработке сценариев аналитиков бизнес-логики, но он обеспечит более серьезный подход к тестированию веб-приложений, что с большей вероятностью гарантирует высокое качество приложений.

Таким образом, разработка программного средства автоматизированного тестирования производительности веб-приложений, основанного на модельном подходе к тестированию, является актуальной задачей.