

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ
РОССИЙСКОЙ ФЕДЕРАЦИИ
ГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО
ОБРАЗОВАНИЯ
«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»
Факультет компьютерных наук

**СБОРНИК СТУДЕНЧЕСКИХ
НАУЧНЫХ РАБОТ
ФАКУЛЬТЕТА КОМПЬЮТЕРНЫХ
НАУК ВГУ**

Выпуск 8

Под редакцией кандидата технических наук Д.Н. Борисова

Воронеж
Издательский дом ВГУ
2014

УДК 004.65 + 004.438.5

ББК Ч481(2)22

С23

Рекомендовано к печати

Ученым советом факультета компьютерных наук ВГУ

С23 Сборник студенческих научных работ факультета компьютерных наук ВГУ / под ред. Д.Н. Борисова; Воронежский государственный университет. – Выпуск 8. – Воронеж : Издательский дом ВГУ, 2014. – 204 с.
ISBN 978=5=9273=2129=2

В сборник включены научные работы студентов факультета компьютерных наук ВГУ, выполненные в 2013–2014 гг. под руководством преподавателей факультета, представленные в виде докладов и рекомендованные к опубликованию оргкомитетами студенческих научных конференций.

УДК 004.65 + 004.438.5

ББК Ч481(2)22

ISBN 978=5=9273=2129=2

© Воронежский государственный университет, 2014

© Оформление. Издательский дом ВГУ, 2014

ОБ ОДНОЙ МОДИФИКАЦИИ ЗАДАЧИ КОММИВОЯЖЕРА

Г.Ю. Белый
Студент

В.А. Родин
Профессор

Введение

В данный момент для некоторых северных областей России существует задача охраны нефтяных вышек или других объектов, находящихся внутри акватории. Установка и обслуживание радиолокационных датчиков вокруг патрулируемых объектов обходится очень дорого, поэтому считаем, что между объектами передвигается катер, на борту которого установлен надводный и подводный радары.

1. Теоретический анализ

Для уменьшения затрат на перемещение необходимо найти путь, при котором катер проходил бы все патрулируемые объекты за минимальное время. Представим патрулируемые объекты как вершины связного графа. Имея N вершин связного графа B_1, \dots, B_{N-1}, B_N , нужно найти путь, при котором все точки должны быть посещаемы один раз. При этом вершина, из которой вышел катер, является конечной для оптимального маршрута.

Эта задача решается как задача коммивояжера, но мы не применяем известный метод ветвей и границ (алгоритм Литтла). Учитывая незначительное число объектов охраны и быстроедействие современных компьютеров, будем перебирать все возможные пути внутри графа (для n – вершин их число $(n-1)!$) и выберем путь, при котором катер проходил бы все патрулируемые объекты за минимальное время [1, 2].

Вычислим сложность алгоритма перебора на компьютере. Внутри пути все точки массива $A, B_1, \dots, B_{N-1}, A$ (где A – это вершина, являющаяся началом пути) можно переставлять. Число перестановок можно выразить формулой $P_{N-1} = (N-1)!$, где N – количество вершин. На место пункта A может претендовать любая из точек массива B_1, \dots, B_{N-1}, B_N . Итоговое число всех возможных переборов $N \times P_{N-1} = N!$.

Целевой функцией является сумма всех затрат на перемещение по графу массива $A, B_1, \dots, B_{N-1}, A$. Или по формуле:

$$F = \sum_{i=1}^N \gamma(B_i, B_{i-1}) \rightarrow \min, \quad (1)$$

где N – количество вершин в графе; $\gamma(B_i, B_{i-1})$ – время, потраченное на перемещение из вершины B_i в вершину B_{i-1} .

Внутри акватории существует множество внешних факторов, влияющих на время прохождения пути. Основным фактором является течение. Поэтому время, затраченное катером на преодоление расстояния между вершинами, вычисляется с учётом воздействия течения, что соответствует следующей формуле:

$$\gamma(B_i, B_{i-1}) = \frac{S(B_i, B_{i-1})}{V_{\text{катера}} + V_{\text{течения}} \cos \alpha}, \quad (2)$$

где $S(B_i, B_{i-1})$ – расстояние от вершины B_i до вершины B_{i-1} ; $V_{\text{катера}}$ – скорость катера перпендикулярно течению; $V_{\text{течения}}$ – скорость течения; α – угол между направлением движения катера и направлением течения; $\gamma(B_i, B_{i-1})$ – время, за которое катер преодолеет расстояние $S(B_i, B_{i-1})$.

Заметим, что время прохождения пути в одном и другом направлении без учёта течения оказывается равным, тогда как, учитывая фактор течения, мы получаем значительную разницу во времени, следовательно $\gamma(B_i, B_{i-1}) \neq \gamma(B_{i-1}, B_i)$.

2. Экспериментальная часть

Продемонстрируем задачу коммивояжера с учетом течения на конкретном примере. Для наглядности представим скорость катера равной 1 км/ч, а скорость течения равной 0,5 км/ч. Пример графа представлен на рисунке.

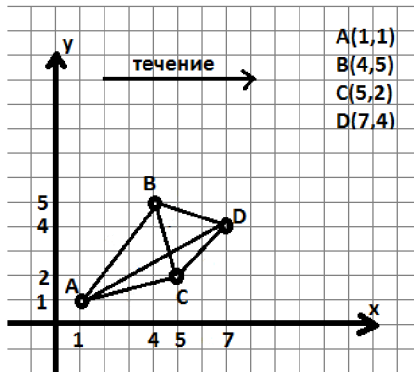


Рисунок. Граф патрулируемых объектов

Катер начинает движение из точки А. Схему пути, а также время его прохождения с учетом или без учета течения можно представить в виде таблицы, результаты которой были получены с помощью созданной нами программы.

Таблица

Результаты эксперимента

	Путь	Время прохождения без учета течения	Время прохождения с учетом течения
	A-B-C-D-A	17ч 42мин	18ч 00мин
	A-B-D-C-A	15ч 07мин	15ч 17мин
	A-C-B-D-A	17ч 09мин	16ч 51мин
	A-C-D-B-A	15ч 07мин	14ч 57мин
	A-D-B-C-A	17ч 09мин	17ч 28мин
	A-D-C-B-A	17ч 42мин	17ч 23мин

Таким образом, наблюдается разница во времени при прохождении пути с учетом течения в одном и другом направлении.

Заключение

С помощью такой модификации задачи коммивояжера можно уменьшить временные затраты при патрулировании нескольких объектов внутри акватории.

В настоящее время данная модификация также востребована при использовании небольших автоматизированных летательных аппаратов для съемки охраняемой территории с учетом движения воздушных потоков.

Данную модификацию также можно реализовать для охраны сухопутных объектов при учете рельефа местности.

Список литературы

1. Бурков В.Н. Теория графов в управлении организационными системами / В.Н. Бурков, А.Ю. Заложнев, Д.А. Новиков. – М. : Синтег, 2001. – 124 с.
2. Орлов А.И. Задачи оптимизации и нечеткие переменные / А.И Орлов. – М. : Знание, 1980. – 64 с.

РАЗРАБОТКА СИНТАКСИЧЕСКОГО И СЕМАНТИЧЕСКОГО КОМПОНЕНТОВ ЛИНГВИСТИЧЕСКОГО ПАРСЕРА РУССКОГО ЯЗЫКА

Ю.А. Богоявленская
Студент

В.В. Гаршина
Доцент

Введение

Естественный язык занимает центральное место в информатике. Ранние работы в этой области были связаны с автоматизацией раскрытия военных шифров. В 50-е годы XX века усилия были направлены на разработку автоматических переводчиков с одного естественного языка на другой. И хотя первоначальная цель так и не была достигнута, эти работы серьезно продвинули решение проблемы естественно-языкового общения с ЭВМ. В настоящее время разрабатываются грандиозные проекты по использованию естественного языка в качестве средства "общения" с компьютерами [1].

В компьютерной лингвистике существует много прикладных задач, которые используют результаты работы специального класса программ – *лингвистических парсеров*. К таким задачам можно отнести: анализ/синтез предложения на естественном языке (ЕЯ) с целью формирования/распознавания запроса на искусственном языке (SQL, SPARQL, поисковые запросы и др.), преобразование предложения с одного ЕЯ на другой ЕЯ (системы машинного перевода), задачи семантического анализа текстов – TextMining, семантическая классификация текстовых ресурсов и др. [2].

Несмотря на то, что исследования в области обработки данных на естественном языке проводятся достаточно давно, задача разработки парсера для русского языка не теряет свою актуальность. В отличие от большинства европейских языков, русский является флективным и допускает свободный порядок слов в предложении. В связи с этим решений, существующих для русского языка, относительно немного, и лишь малая их часть находится в открытом доступе. Поэтому было решено разработать собственный парсер.

1. Лингвистический парсер

Лингвистический парсер — комплекс программных модулей, обеспечивающий разбор линейной последовательности лексем (слов) языка исходного текста во внутреннее представление смысла этого

предложения. При этом используется многоуровневый анализ предложения на ЕЯ, реализующий морфологический, синтаксический, семантический (иногда и прагматический) языковой уровень. На каждом языковом уровне используются свои структуры данных, которые обрабатываются и формируются соответствующими компонентами лингвистического парсера [2].

Лингвистический парсер состоит из морфологического, синтаксического и семантического анализаторов. Далее подробнее описано, что представляют собой эти компоненты, и как они реализованы.

2. Морфологический анализатор

Морфологический анализатор, используя морфологические словари, строит разбор последовательности входящих в анализируемое предложение слов с указанием части речи и морфологических характеристик. [2]

Этот компонент лингвистического парсера уже был создан в рамках моей предыдущей работы [3, 4], и только подвергался усовершенствованиям. В частности, было принято решение использовать в качестве источника данных о словоформах не файлы, а базу данных. Морфологический анализатор получает на вход слово и строит гипотезы о том, чем это слово является. Полученная совокупность гипотез возвращается затем синтаксическому анализатору.

3. Синтаксический анализатор

Синтаксический анализатор реализует построение дерева зависимостей, в узлах которого стоят слова данного предложения с указанием части речи и грамматических характеристик, а дуги соответствуют специфичным для данного естественного языка отношениям подчинения [2].

В начале своей работы синтаксический анализатор проводит графематический анализ – выделяет отдельные слова и знаки препинания в предложении. Эти слова передаются морфологическому анализатору, после чего он пытается устранить имеющиеся неоднозначности. Для этого проверяются окрестности слов, для которых было получено несколько гипотез, с целью выявить отношения согласования, управления, или примыкания.

Далее происходит проверка того, является ли предложение простым или сложным, и если сложным, то каким именно – сложносочиненным или сложноподчиненным. Существовавшая система правил синтаксического анализа подверглась значительным изменениям, чтобы повысить точность этой проверки и последующего разбора сложных предложений.

На выходе получается дерево разбора. Если предложение было

простым, то в вершине находится сказуемое. Для сложных предложений возможны вариации: корневым узлом для сложносочиненного предложения считается союз или знак препинания, разделяющий простые предложения; для сложноподчиненных – сказуемое главного предложения.

4. Семантический анализатор

Семантический анализатор также использует дерево зависимостей, но в его узлах стоят либо предметные имена, либо слова универсального семантического языка (например, имена таблиц, в которых сосредоточены сведения о данной предметной области, атрибуты таблиц, операторные символы). Дуги соответствуют универсальным отношениям семантического подчинения: аргументное, атрибутивное, конъюнкция, дизъюнкция, равенство, неравенство, больше, меньше, принадлежит, не принадлежит и т. п. [2].

Если синтаксический анализатор работает на уровне предложения, то семантический – занимается разбором на уровне абзаца текста. Для него определен ряд категорий, таких как субъект, объект, действие, время, место и др. Обобщенные категории вроде субъекта или объекта соотносятся со словами на основе результатов синтаксического анализа; для специфических категорий, таких, как время или место, данные берутся из базы.

Как только семантические категории получены для всех предложений, происходит их соотнесение между собой уже на уровне текста. Например, если субъектом первого предложение является одушевленное лицо, а субъектом второго – местоимение того же рода, парсер предполагает, что речь идет об одном и том же действующем лице. Таким образом, на выходе получается обобщенная формальная семантическая модель.

В настоящий момент для разбора поддерживаются только небольшие связные тексты на русском языке, не включающие в себя цифр и математических знаков. Правильность разбора зависит от сложности поступающего текста.

Заключение

В дальнейшем планируется провести серию статистических экспериментов для оценки точности разбора. На основе полученных данных будет осуществлено пополнение базы правил синтаксического анализа, добавление новых семантических категорий и правил для сопоставления этих категорий с входными данными.

Также будут проводиться исследования для решения проблемы омонимии, которая пока что разрешается только в простейших случаях.

Полученный в рамках данной работы парсер, может использоваться

в качестве демонстрационного образца в учебном процессе. Впоследствии допускается его интеграция в приложения, решающие частные задачи обработки данных на русском языке.

Список литературы

1. Большакова Е.И. Автоматическая обработка текстов на естественном языке и компьютерная лингвистика : учеб. Пособие / Е.И. Большакова, Э.С. Клышинский, Д.В. Ландэ и др. – М. : МИЭМ, 2011. – 272 с.
2. Апресян Ю.Д. Лингвистический процессор для сложных информационных систем / Ю.Д. Апресян, И.М. Богуславский, Л.Л. Иомдин и др. – М. : Наука. – 1992. – 256 с.
3. Гаршина В.В. Разработка лингвистического парсера русского языка / В.В. Гаршина, Ю.А. Богоявленская // Вестник Воронежского государственного университета. Сер. Системный анализ и информационные технологии. – 2012. – № 2. – С. 174–182.
4. Гаршина В.В. Разработка модулей морфологического и синтаксического анализа лингвистического парсера русского языка / В.В. Гаршина, Ю.А. Богоявленская // Вестник факультета прикладной математики и механики. – 2013. – Вып. 10. – С. 63–67.

МЕТОДЫ И АЛГОРИТМЫ ПОСТРОЕНИЯ ТРЕХМЕРНЫХ МОДЕЛЕЙ В СИСТЕМАХ

А.О. Божок
Студент

С.Д. Кургалин
Профессор

А.А. Крыловецкий
Доцент

Введение

Компьютерная графика – это область деятельности, в которой компьютер используется для синтеза изображений и обработки информации из реального мира. Компьютерная обработка визуальной информации получила в настоящее время новый импульс развития и широкое развитие в различных отраслях науки и производства, появились новые, смежные дисциплины, одна из которых – машинное зрение.

В основе любой системы компьютерного зрения лежат две задачи: получение дальнометрических данных (на основе двух или более камер

путем сопоставления изображений сцены, либо с использованием активных сенсоров глубины) и построение трехмерных моделей по множеству точек. На современном этапе развития алгоритмы стереосопоставления позволяют с использованием пары камер и без применения какого-либо специализированного оборудования производить построение трехмерной модели объектов сцены в режиме реального времени [1].

1. Постановка задачи

Целью данной работы является создание системы компьютерного зрения, позволяющей производить построение трехмерной модели объектов сцены в режиме реального времени. Для получения дальнометрических данных необходима последовательная реализация следующих этапов обработки изображений:

1. *Стереокалибровка камер в общей системе координат и получение матриц проекции для каждой из камер.* Необходимо получить матрицу, описывающую перенос точки из системы координат одной камеры в систему координат другой. В случае если камеры неподвижно закреплены друг относительно друга, полученные в ходе калибровки данные могут быть вычислены однократно и представлены в удобном для дальнейших преобразований виде.

2. *Захват и ректификация изображений по эпиполярным линиям, построенным на основе матриц преобразований.* Для пары изображений задаются линии сечения плоскостей левого и правого изображений одной и той же плоскостью. Изображения необходимо деформировать таким образом, чтобы строчки пикселей левого и правого изображений с одинаковыми индексами соответствовали эпиполярным линиям. Ректификация изображений является важным этапом в разработке систем стереозрения. Она позволяет минимизировать искажения, связанные с неправильным расположением камер.

3. *Обработка стереопары, триангуляция точек, построение дальнометрических изображений и карт глубины.* Для ректифицированной стереопары устанавливаются соответствия между пикселями левого и правого изображений. Полученные данные – разница координат соответствующих пикселей – используются при построении дальнометрических изображений. Цвет пикселя на таком изображении определяет, насколько каждая конкретная точка пространства удалена от картинной плоскости камер.

4. Дальнейшие действия включают в себя построение и использование в том или ином виде *трехмерного облака точек сцены*. Процесс построения трехмерной модели снимаемой сцены предполагает нахождение для каждой точки карты глубины соответствующих

трёхмерных точек в системе координат одной из камер. Для этого, используя параметры калибровки камер, нужно восстановить параметры лучей, на которых находится точка для каждой из камер, и найти точку их пересечения или максимального сближения. В результате получается трёхмерное облако точек (mesh), лежащих на поверхности сканируемого объекта [2, 3].

2. Реализация

Стереосопоставление является ядром системы стереозрения. На вход алгоритма подается стереопара, а результатом работы алгоритма является карта диспаратитетов (disparity map). Диспаратитетом является разница в значении абсциссы пикселя на левом и правом изображении. От точности работы данных алгоритмов напрямую зависит возможность применения системы в целом. Множество алгоритмов стереосопоставления удобно разбить на две категории: методы локального и глобального сопоставления. Соответственно определяется расстояние до каждого индивидуального пикселя, либо до некоторых объектов – групп пикселей. Такими объектами могут быть отрезки, области, многоугольники. Каждая из групп алгоритмов имеет свои границы применимости: первый тип алгоритмов имеет лучшие показатели на изображениях, имеющих много мелких деталей, второй тип лучше работает с большими однородными участками без текстуры.

В реальных системах компьютерного зрения используется смешанная, адаптивная методология. Кроме увеличения качества сопоставления, это также позволяет оптимизировать производительность алгоритма до скорости выполнения в режиме реального времени. В работе применены алгоритмы стереоскопической калибровки камер и ректификации стереопары по эпиполярным линиям, а также алгоритм получения карты глубины, реализованных в библиотеке OpenCV (и ее расширения на платформу java – JavaCV) с использованием параллельного кода, способного выполняться на современных многоядерных CPU и GPU [4].

В данной работе рассмотрена математическая модель компьютерного стереозрения, включающая трехмерную модель сцены, восстановленную по ее стереоизображениям. При получении трехмерной модели используется дальнометрическая информация в виде карты глубины и внутренних параметров камеры. В настоящий момент экспериментальная реализация модели включает в себя данные о координатах и цвете точки в системе RGB и требует 290 байт для представления каждой точки пространства. Модель также подвергается фильтрации для удаления избыточной информации – намеренно отсекаются точки, расположенные на условно бесконечном расстоянии от

картинной плоскости. Кроме того, возможно огрубление модели путем объединения нескольких близко расположенных точек. Данный подход позволяет снизить вычислительную нагрузку при аффинных преобразованиях облака точек в процессе визуализации, удалить избыточные детали при построении полигональной модели, увеличить быстродействие при передаче модели по сети. Построение и отображение трехмерной модели сцены в представленной работе производится при помощи графической технологии OpenGL и её расширения JOGL, задействующей только ресурсы видеокарт.

На основе описанных алгоритмов построен модульный программный комплекс получения и обработки стереоизображений, особенностью которого является слабая связность модулей, кросс-платформенная реализация и ориентированность на параллельную обработку данных. На рисунке последовательно продемонстрирован процесс построения эпиполярных линий изображения, слияния ректифицированных изображений в анаглифную стереопару, интерфейс трехмерного визуализатора оригинальной разработки.



Рисунок. Формирование анаглифного стереоизображения и реконструкция облака точек

Заключение

На настоящий момент результатом работы являются прототип приложения расширенной реальности, позволяющий строить и визуализировать трехмерные изображения сцены. Специфика построения модульного приложения позволяет быстро менять функциональность, использовать только отдельные блоки в рамках других научных проектов, а также обеспечивает легкость и быстроту переноса приложения на различные системы – данное приложение является кроссплатформенным.

Уровень технологического развития сегодня позволяет создавать системы обработки изображений, функционирующие в реальном времени на персональных компьютерах. Примером таких систем являются решения для формирования стереоизображений. Сегодня создание подобных средств на базе мобильных устройств является одной из приоритетных задач для производителей техники и программистов.

Помимо очевидной полезности аналогичных решений в области компьютерного стерео для производства и различных направлений робототехники, возможно также применение данных технологий в сфере развлечений, например, в сфере дополненной реальности, распознавания жестов или для систем видеокommunikации.

Список литературы

1. Вахитов А.Т. Обзор алгоритмов стереозрения. Стохастическая оптимизация в информатике. Выпуск 4 / А.Т. Вахитов, Л.С. Гуревич, Д.В. Павленко; под ред. О.Н. Граничина. – СПб. : Издательство С.-Петербургского университета, 2008. – 299 с.
2. Технологии 3D кино. – Режим доступа: <http://hdsofa.ru/Stati/Tehnologii-3D-kino-IMAX-RealD-SuperD-Dolby-3D-XpanD-MasterImage.html>
3. Papadimitriou D. Epipolar line estimation and rectification for stereo image pairs / D. Papadimitriou, T. Dennis // IEEE Transactions on Image Processing, 1996
4. Протасов С.И. Использование web-камер как источника стереоизображений в реальном времени / С.И. Протасов, А.А. Крыловецкий // Информатика: проблемы, методология и технологии: материалы XI международной научно-методической конференции, Воронеж, 10-11 февраля 2011 г. – Т. 2. – С. 229–232.

НЕКОТОРЫЕ ВОПРОСЫ КЛАССИФИКАЦИИ ПЯТИМЕРНЫХ АЛГЕБР ЛИ

Д.А. Бубельник
Студент

А.В. Лобода
Профессор

Введение

Задачей данной работы является установление соответствий между некоторыми алгебрами Ли, отвечающими аффинно-однородным вещественным гиперповерхностям 3-мерного комплексного пространства, и 5-мерными алгебрами из известной общей классификации Мубаракзянова. Для решения этой задачи используются инвариантные свойства алгебр Ли, связанные с их центральными рядами, изучаются свойства и понятия, традиционно считающиеся наиболее важными в теории алгебр Ли (такие как нильпотентность и разрешимость).

Поставленная задача полностью решена для двух предложенных семейств матричных алгебр Ли.

1. Замечания о классификации Мубаракзянова

В данном разделе мы приводим структурные соотношения для некоторых алгебр Ли из классификации Мубаракзянова [1, 2]. Напомним, что при описании 5-мерных алгебр приходится использовать информацию об алгебрах меньшей размерности, а также опираться на свойства разложимости и разрешимости изучаемых алгебр. Нам потребуются ниже два типа разложимых 5-мерных алгебр

$$g_{4,k} + g_1 \quad (1)$$

В этой записи g_1 является одномерной алгеброй Ли (линейной оболочкой базисного элемента e_5 , не упоминаемого в приводимых ниже структурных соотношениях). Таблицы умножения в этих алгебрах имеют вид

$$\begin{aligned} g_{4,8} + g_1: e_1 \circ e_4 &= (1 + h)e_1, e_2 \circ e_3 = e_1, \\ e_2 \circ e_4 &= e_2, e_3 \circ e_4 = he_3. \end{aligned} \quad (2)$$

Примечание [2]: $|h| \leq 1$.

$$\begin{aligned} g_{4,9} + g_1: e_1 \circ e_4 &= 2pe_1, e_2 \circ e_3 = e_1, \\ e_2 \circ e_4 &= pe_2 - e_3, e_3 \circ e_4 = e_2 + pe_3. \end{aligned} \quad (3)$$

Из неразложимых алгебр Ли нам потребуются алгебры со следующими таблицами умножения в них:

$$g_{5,4}: e_2 \circ e_4 = e_1, e_3 \circ e_5 = e_1; \quad (4)$$

$$\begin{aligned} g_{5,26}: e_1 \circ e_5 &= 2pe_1, e_2 \circ e_3 = e_1, e_2 \circ e_5 = pe_2 + e_3, \\ e_3 \circ e_5 &= -e_2 + pe_3, e_4 \circ e_5 = \varepsilon e_1 + 2pe_4; \end{aligned} \quad (5)$$

Примечание [1]: $\varepsilon = \pm 1$.

$$\begin{aligned} g_{5,30}: e_1 \circ e_5 &= (2+h)e_1, e_2 \circ e_4 = e_1, e_2 \circ e_5 = (1+h)e_2, \\ e_3 \circ e_4 &= e_2, e_3 \circ e_5 = he_3, e_4 \circ e_5 = e_4; \end{aligned} \quad (6)$$

$$\begin{aligned} g_{5,37}: e_1 \circ e_4 &= 2e_1, e_2 \circ e_3 = e_1, e_2 \circ e_4 = e_2, \\ e_2 \circ e_5 &= -e_3, e_3 \circ e_4 = e_3, e_3 \circ e_5 = e_2. \end{aligned} \quad (7)$$

2. Изучаемые семейства алгебр

В связи с задачей описания однородных поверхностей 3-мерного комплексного пространства авторы работы [3] построили большое количество 5-мерных матричных алгебр Ли. В данной работе рассматриваются в сравнении с классификацией Мубаракзянова [1] два семейства таких алгебр, зависящие от нескольких параметров.

Семейство алгебр (2.2)

Базис любой алгебры из этого семейства описывается формулами ($m_1, m_2, t_7, t_8, t_{16}$ – вещественные параметры):

$$\begin{aligned} E_1 &= \begin{pmatrix} 2(t_7 - m_1 - im_2) & 0 & i(m_1 + im_2)(m_1 - t_7) & 1 \\ 0 & t_7 + it_8 & 0 & 0 \\ 4i & 0 & 2m_1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, \\ E_2 &= \begin{pmatrix} 2m_2 & 0 & 0 & i \\ 0 & m_2 + it_{16} & 0 & 0 \\ 0 & 0 & 2m_2 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, E_3 = \begin{pmatrix} 0 & -m_1 + t_7 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 2i & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, \\ E_4 &= \begin{pmatrix} 0 & i(m_1 - t_7) & 0 & 0 \\ 0 & 0 & 0 & i \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, \end{aligned} \quad (8)$$

$$\begin{aligned} E_5 &= \\ = & \begin{pmatrix} -m_2(m_1 - t_7) & 0 & 0 & 0 \\ 0 & -(\frac{1}{2}m_1 - \frac{1}{2}t_7)(m_2 + it_{16}) & 0 & 0 \\ 0 & 0 & -m_2(m_1 - t_7) & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix}. \end{aligned}$$

Используем матричное подобие $g^* = C^{-1}gC$, где матрица C имеет вид:

$$C = \begin{pmatrix} 1 & 0 & i \frac{m_1 - t_7}{2} & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}. \quad (9)$$

Вместо базисного элемента E_5 введем сумму $E_5 + \frac{m_1 - t_7}{2} E_2$. Тогда базис любой из алгебр (8) можно записать в упрощенной форме:

$$\begin{aligned} E_1 &= \begin{pmatrix} -2im_2 & 0 & 0 & 1 \\ 0 & t_7 + it_8 & 0 & 0 \\ 4i & 0 & 2t_7 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, \\ E_2 &= \begin{pmatrix} 2m_2 & 0 & 0 & i \\ 0 & m_2 + it_{16} & 0 & 0 \\ 0 & 0 & 2m_2 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, E_3 = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 2i & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, \\ E_4 &= \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & i \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, E_5 = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix}. \end{aligned} \quad (10)$$

Структурные соотношения (задающие таблицу умножения) в такой алгебре имеют вид

$$\begin{aligned} E_1 \circ E_2 &= -4E_5, E_1 \circ E_3 = t_7 E_3 + t_8 E_4, E_1 \circ E_4 = -t_8 E_3 + t_7 E_4, \\ E_1 \circ E_5 &= 2t_7 E_5, E_2 \circ E_3 = m_2 E_3 + t_{16} E_4, E_2 \circ E_4 = -t_{16} E_3 + m_2 E_4, \\ E_2 \circ E_5 &= 2m_2 E_5, E_3 \circ E_4 = -4E_5. \end{aligned} \quad (11)$$

Семейство алгебр (2.6)

Базисы алгебр из этого семейства имеют вид $(m_1, m_2, m_3, m_4$ - вещественные параметры):

$$\begin{aligned} E_1 &= \begin{pmatrix} -2m_1 & m_3 - im_4 & im_1^2 & 1 \\ 0 & 0 & 0 & 0 \\ 4i & 0 & 2m_1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, \\ E_2 &= \begin{pmatrix} 0 & 0 & 0 & i \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, E_3 = \begin{pmatrix} \frac{3}{2}m_3 & -m_1 & \frac{i}{4}m_1 m_3 & 0 \\ 0 & m_3 & 0 & 1 \\ 0 & 2i & 2m_3 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, \end{aligned} \quad (12)$$

$$E_4 = \begin{pmatrix} \frac{3}{2}m_4 & im_1 & \frac{i}{4}m_1m_4 & 0 \\ 0 & m_4 & 0 & i \\ 0 & 2 & 2m_4 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, E_5 = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix}.$$

Структурные соотношения в такой алгебре имеют вид:

$$\begin{aligned} E_1 \circ E_2 &= -2m_1E_2 - 4E_5, E_1 \circ E_3 = -\frac{1}{2}m_3E_1 - m_4E_2, \\ E_1 \circ E_4 &= -\frac{1}{2}m_4E_1 + m_3E_2, E_1 \circ E_5 = m_1^2E_2 + 2m_1E_5, \\ E_2 \circ E_3 &= -\frac{3}{2}m_3E_2, E_2 \circ E_4 = -\frac{3}{2}m_4E_2, \\ E_3 \circ E_4 &= -2m_1E_2 - m_4E_3 + m_3E_4 - 4E_5 \\ E_3 \circ E_5 &= \frac{1}{4}m_1m_3E_2 + 2m_3E_5, E_4 \circ E_5 = \frac{1}{4}m_1m_4E_2 + 2m_4E_5 \end{aligned} \quad (13)$$

Задача установления требуемых соответствий между алгебрами из [3] и классификацией [1] оказывается более простой для семейства (2.6). Здесь следует рассмотреть два случая:

1. $m_3 = m_4 = 0$.

Количество нетривиальных умножений в (13) уменьшается до двух после введения нового базисного элемента $E_5^* = m_1E_2 + 2E_5$. Структурные соотношения с введенным E_5^* :

$$E_1 \circ E_2 = -2E_5^*, E_3 \circ E_4 = -2E_5^* \quad (14)$$

2. $m_3 + im_4 \neq 0$.

В общем случае, при $m_3 + im_4 \neq 0$, алгебра с базисом (12) обладает свойством симметрии между m_3 и m_4 . Поэтому достаточно обсудить случай, при котором

$$m_3 \neq 0. \quad (15)$$

С учетом (15) матрица E_3 имеет простой спектр

$$\left(\frac{3}{2}m_3, m_3, 2m_3, 0\right). \quad (16)$$

Будем рассматривать подобия $g^* = S^{-1}gS$. Благодаря подобию с матрицей, состоящей из собственных векторов E_3

$$S = \begin{pmatrix} 1 & m_1 & im_1 & m_1 \\ 0 & m_3 & 0 & 2m_3 \\ 0 & -2i & 2 & -2i \\ 0 & 0 & 0 & -2m_3^2 \end{pmatrix} \quad (17)$$

мы можем перейти к новой алгебре. За счет другого подобия с диагональной матрицей

$$D = \text{diag}\left(\frac{m_3^2 + m_4^2}{m_3}, \frac{m_3 + im_4}{m_3}, \frac{m_3^2 + m_4^2}{m_3^2}, 1\right) \quad (18)$$

алгебра становится независимой от параметров m_3, m_4 , а ее базис принимает вид

$$\begin{aligned} E_1 &= \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 2i & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, E_2 = \begin{pmatrix} 0 & 0 & 0 & i \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, \\ E_3 &= \begin{pmatrix} 3 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 4 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, E_4 = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -2i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, \\ E_5 &= \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix}. \end{aligned} \quad (19)$$

Структурные соотношения

$$\begin{aligned} E_1 \circ E_2 &= -2E_5, E_1 \circ E_4 = -2E_2, E_3 \circ E_1 = E_1, E_3 \circ E_2 = 3E_2, \\ E_3 \circ E_4 &= 2E_4, E_3 \circ E_5 = 4E_5. \end{aligned} \quad (20)$$

После завершающих замен $E_1^* = \frac{1}{2}E_1, E_3^* = -E_3$ мы получим шесть следующих структурных соотношений

$$\begin{aligned} E_1^* \circ E_2 &= -E_5, E_1^* \circ E_4 = -E_2, E_3^* \circ E_1^* = -E_1^*, E_3^* \circ E_2 = -3E_2, \\ E_3^* \circ E_4 &= -2E_4, E_3^* \circ E_5 = -4E_5. \end{aligned} \quad (21)$$

в любой из алгебр этого семейства.

3. Теоремы о соответствии

Теорема 1. При различных значениях параметров m_2, t_7, t_8, t_{16} алгебры семейства (2.2) изоморфны следующим пятимерным алгебрам из классификации Мубаракзянова:

$g_{4,8} + g_1$, считая $h = 1$, при $(m_2 = t_8 = t_{16} = 0), (t_7 = t_8 = t_{16} = 0), (t_8 = t_{16} = 0)$;

$g_{4,9} + g_1$, считая параметр $\alpha = \frac{1}{p}$, при $(t_7 = t_8 = 0), (m_2 = t_{16} = 0)$;

$g_{5,37}$ при $(m_2 = t_8 = 0), (m_2 = 0), (t_7 = 0), (t_7 = t_{16} = 0), (t_8 = 0), (m_2, t_7, t_8, t_{16}) \neq (0, 0, 0, 0), (t_{16} = 0)$;

$g_{5,4}$ при $(m_2 = t_7 = t_8 = t_{16} = 0)$;

$g_{5,26}$, считая $p = 0, \varepsilon = 1$, при $(m_2 = t_7 = t_8 = 0), (m_2 = t_7 = 0), (m_2 = t_7 = t_{16} = 0)$.

Теорема 2. При различных значениях параметров m_3, m_4 алгебры семейства (2.6) изоморфны следующим пятимерным алгебрам из классификации Мубаракзянова:

$g_{5,4}$ при $(m_3 = m_4 = 0)$;

$g_{5,30}$, считая $h = 2$, при $(m_3 + im_4 \neq 0)$.

4. Схема доказательства

Используемые далее понятия подробно рассмотрены в работе [4], полные схемы доказательств для алгебр Ли вплоть до порядка 5 можно найти в работах [1, 2].

Существует два подхода к доказательству предложенных теорем.

Первый подход основан на использовании понятий нижнего центрального ряда и производного ряда алгебр и, как следствие, на нильпотентности и разрешимости. Докажем первым способом пункт 1 теоремы 2.

Доказательство (первый способ)

Рассчитаем размерность нижнего центрального ряда, получим последовательность 5, 1, 0.

Рассчитаем размерность производного ряда, получим последовательность 5, 1, 0.

Аналогичными размерностями обладает единственная алгебра из классификации Мубаракзянова – $\mathfrak{g}_{5,4}$. Алгебры изоморфны при следующих заменах:

$$e_1 \rightarrow -2E_5^*, e_2 \rightarrow E_1, e_3 \rightarrow E_3, e_4 \rightarrow E_2, e_5 \rightarrow E_4. \quad (22)$$

Второй подход основан на анализе инвариантных характеристик, таких как порядок (размерность) центра Z обсуждаемой алгебры, порядок (размерность) ее максимального нильпотентного идеала, разложимость алгебры. Приведем общую схему доказательства для пункта 3 теоремы 1.

Доказательство (второй способ)

$\dim(Z) = 0, \dim(\text{nilradical}) = 3$, причем nilradical не является разложимым. Согласно [1], в этом случае $\text{nilradical} \cong \mathfrak{g}_{3,1}$. Значит, надо искать канонические формы матриц S и H , которые соответствуют двум элементам, не вошедшим в nilradical .

В данном случае, канонические формы матриц, соответствующих элементам, имеют вид, приводимый к

$$\begin{pmatrix} 2 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{pmatrix}. \quad (23)$$

Им соответствует алгебра $\mathfrak{g}_{5,37}$.

Заключение

В работе показано, что семейства алгебр, получаемые «естественным» для задачи об однородности (см. [3]) способом, распадаются в классификации [1] на большое семейство разрозненных представителей. Поэтому при решении задачи только на основе имеющейся классификации могут быть утрачены «внутренние» родственные связи между однородными поверхностями и их алгебрами Ли.

Список литературы

1. Мубаракзянов М.Г. Классификация вещественных структур алгебр Ли пятого порядка / М. Г. Мубаракзянов // Изв. вузов. Матем. – 1963. – №3. – С. 99–106.
2. Мубаракзянов М. Г. О разрешимых алгебрах Ли / М.Г. Мубаракзянов // Изв. вузов. Матем. – 1963. – № 1. – С. 114–123.
3. Atanov A.V. Affine homogeneous strictly pseudoconvex hypersurfaces of the type $(1/2, 0)$ in C^3 / A. V. Atanov, A. V. Loboda, A.V. Shipovskaya. – Режим доступа: <http://arxiv.org/abs/1401.2252>
4. Джекобсон Н. Алгебры Ли: научное издание / Н. Джекобсон. – М. : Мир, 1964. – 358 с.

ПОСТРОЕНИЕ МОДЕЛИ ПЕРВИЧНОЙ АУТЕНТИФИКАЦИИ ПРИ ПОМОЩИ СИСТЕМЫ «ЭЛЕКТРОННЫЙ СЕКРЕТАРЬ»

Е.С. Воробьева
Студент

А.А. Жижелев
Ассистент

Введение

Зачастую перед секретарями ставится задача разграничения доступа «извне» к сотрудникам компании. В рамках конкурентной борьбы возможность свободного доступа к тем или иным сотрудникам, а так же извлечение из секретаря определенной информации о конкурирующей фирме часто может принести немалую пользу.

Существует множество стратегий, которые соответствующие специалисты используют, чтобы прорваться через секретаря. В таких ситуациях секретарь должен быть предельно внимателен и должен самостоятельно определять, каким объемом информации позвонивший уже владеет, чтобы принять или отклонить запрос в доступе. В данной статье предлагается определять это не эвристически, а при помощи системы «Электронный секретарь», которая самостоятельно оценивает уровень доступа позвонившего.

1. Теоретический анализ

Согласно конверсационному анализу (анализ разговора, который происходит в естественных ситуациях), некоторые диалоговые акты образуют смежные пары, в которых первый элемент гарантирует

появление второго элемента. Компьютер как участник диалога должен быть способен отличить первый элемент смежной пары (ожидающий ответа) от второго элемента, а также от реплик, не ожидающих специфических ответных действий (например, реплик обратной связи: *так/ясно*) [1]. Для этого, стоит обратиться к теории речевых актов (ТРА).

Основы ТРА были заложены в середине XX века английским философом лингвистом Джоном Остином, который предложил называть минимальную целостную единицу речевой деятельности речевым актом.

Так же он развенчал существовавшее тогда заблуждение, что любое высказывание либо утверждает какой-либо факт, либо описывает положение вещей, которые могут быть истинными или ложными. Дж. Остин называет такие высказывания констативам. Костативы обращены к явлениям мира, локализуемым вне сферы высказывания, описывают мир. Если же обратиться к практике повседневной речи, то вполне очевидно, что язык можно использовать для совершения разнообразных действий: отдавать приказы, приносить извинения, давать обещания, выдвигать предположения, предупреждать кого-то, порицать, приветствовать и т.д. Такие высказывания означают не простое произнесение слов, они ничего не констатируют, а предполагают совершение действия – являются частью поступков и действий. Дж. Остин предложил назвать их перформативами.

Перфомативы нельзя оценивать системой «истина-ложь», т.к. они не описывают мир, а выступают как словесные действия, меняя положение дел и создавая новые ситуации. Перформативные высказывания должны оцениваться как успешные (удачные, уместные) и неуспешные (неудачные, неуместные). Чтобы проиллюстрировать, автор приводит несколько стандартных высказываний, произносимых в определенных социальных кругах.

1) на церемонии бракосочетания: *«Да»* («я согласен взять эту женщину в качестве моей законной жены»);

2) на церемонии освящения судна с разбиванием бутылки с шампанским о нос корабля: *«Я присваиваю этому кораблю имя «Королева Елизавета»»*;

3) в тексте завещания: *«Я завещаю свои часы моему брату»*;

4) при заключении пари в споре: *«Держу пари на шесть пенсов, что завтра будет дождь»*.

Так, если в результате речевого акта заседание открылось, в результате речевого акта состоялось бракосочетание в церкви, профессор ответил на ваше приветствие и школьник хотя бы на некоторое время перестал шалить, то эти речевые акты можно назвать успешными.

Но если вы говорите: «Я приветствую вас, господин профессор!» — а профессор, вместо того чтобы ответить на приветствие, переходит на

другую сторону улицы, если мальчик, пообещав, что он «больше не будет», тут же начинает опять, если у священника к моменту бракосочетания был отнят сан и если собрание освистало председателя — эти речевые акты неуспешны.

Выполнить речевой акт по Дж. Остину значит:

- произнести членораздельные звуки, принадлежащие определенному общепринятому языковому коду;
- построить высказывание из слов данного языка по правилам его грамматики;
- снабдить высказывание смыслом и референцией (соотнести с действительностью), осуществив речение – *локутивный акт*.
- придать речению целенаправленность (выразить его коммуникативную/иллокутивную цель) – *иллокутивный акт*.
- вызвать искомые последствия: воздействовать на сознание или поведение адресата; создать новую ситуацию (напр., объявление войны) – *перлокутивный акт* [2].

В рамках наших исследований первый и последний уровни речевых актов нас мало интересуют. Обратимся к иллокутивному акту, который выражает коммуникативную цель речевого акта.

Как и самим Дж. Остином, так и многими его последователями предлагались различные классификации иллокутивных актов. После сравнительного анализа наиболее состоятельных классификаций, была выбрана классификация предложенная К. Бахом (основными критериями оценки были минимальное пересечение классов и возможность применения классификации в телефонном диалоге).

К. Бах предлагает классификацию с учетом глаголов. Он выделяет четыре основных класса:

- констати́вы: утверждение, подтверждение, объявление, ответ, определение, согласие, предположение, отрицание, несогласие, идентификация, информирование, и т.д.;
- директи́вы: совет, замечание, спрашивание, распоряжение, просьба, требование и т.д.;
- комиссивы: согласование, предложение, обещание, предложение услуг и т.д.;
- призна́ния: извинение, поздравление, приветствие, выражение благодарности, согласие принять предложение и т.д.

Данная классификация достаточно подробна, чтобы в дальнейшем адаптировать ее для системы «Электронный секретарь».

2. Экспериментальная часть

Телефонный диалог четко структурирован. В конвенциональной вводной части информатор представляется, и собеседники обмениваются

приветствиями. Отвечая на звонок, информатор придерживается шаблона: <название учреждения> <имя информатора> <приветствие>

Клиент, как правило, ограничивается ответным приветствием и переходит к главному информационному запросу, открывающему главную часть диалога. Информатор может инициировать уточняющие поддиалоги после запроса, клиент – после получения ответа на запрос. Диалог заканчивается конвенциональной частью, в которой клиент благодарит информатора, вместе с тем инициируя завершение разговора, а информатор принимает благодарность и тем самым завершает разговор. Если диалог длится дольше обычного, то клиент не ограничивается благодарностью, но и эксплицитно прощается. Таким образом, в общем виде структура диалога будет выглядеть следующим образом.

1. *Конвенциональное введение.* Взаимные приветствия и самоидентификация.
2. *Запрос (уточнение).* Выясняется цель звонка. Это может быть соединение с определенным сотрудником или запрос о предоставлении информации.
3. *Ответ (уточнение).* Выполнение или отклонение запроса.
4. *Конвенциональное окончание.* Выражение благодарности и формальное прощание.

Грамматика представляет телефонный звонок как последовательность смежных пар диалоговых актов, в которых первый элемент производится одним участником диалога, а второй – его собеседником.

Смежные пары диалоговых актов предполагают использование магазинной структуры. Начальный запрос клиента задает цель, помещаемую в магазин и остающуюся там, пока запрос не будет выполнен. Каждый последующий запрос задает новую (под)цель, которая должна быть выполнена, чтобы достичь начальной цели. Цели помещаются в магазин в хронологическом, а извлекаются в обратном хронологическом порядке.

Работая над диалоговой системой, можно попробовать смоделировать «идеального» информатора – четко произносящего реплики, терпеливого и вежливого. В естественных диалогах информатор не всегда отвечает этому идеалу, однако люди информаторы пользуются удобными стратегиями, присущими устной речи и позволяющими информатору быть лаконичным и кооперативным.

Перед системой «Электронный секретарь» лежат две основные задачи:

- 1) определение уровня доступа оппонента (оценка его знаний);
- 2) ведение (поддержка) диалога.

Решение второй задачи происходит за счет подробного построения смежных пар диалоговой формы, к которым система обращается при обнаружении ключевых глаголов или оборотов речи, основываясь на классификации, предложенной К. Бахом. Так, при идентификации позвонившего как возможного клиента, система должна предложить ему, к примеру, связать с одним из менеджеров компании занимающимся сбытом товаров, а при обращении потенциально нового сотрудника – соединить с отделом кадров.

При запросе на соединение с определенным человеком система обращается к базе данных со списком сотрудников компании, каждый из которых принадлежит либо к группе со свободным доступом, либо к группе с ограниченным доступом. Для соединения с сотрудником из первой группы достаточно правильно назвать его ФИО. Для соединения с сотрудником из второй группы, позвонившему придется ответить на ряд вопросов, в ходе которых будет проведена оценка объемов информации, которыми он располагает.

Прежде всего, для каждого из сотрудников необходимо установить балльный порог, который отражает его заинтересованность/необходимость в соединении с позвонившим. Так, например, для старшего менеджера, чья работа предполагает частый контакт с клиентами, логично будет установить низкий порог, в то время как для главного бухгалтера этот порог будет значительно выше.

Во время разговора параллельно складываются баллы в трех разных переменных: за правильные ответы, неправильные ответы и не идентифицированные ответы (в случае затруднения с ответом или сбоя в системе). При пересечении порога одной из переменных, система принимает решение о доступе в первом случае или об отклонении запроса в двух других случаях.

Предварительно для каждого вопроса выставлены веса, причем таким образом, что при правильном ответе на «легкие» вопросы баллы зачисляются не высокие, но в других случаях, переменные могут принять критические значения, близкие к пороговым.

Так же стоит учитывать ошибки первого рода, когда система, вследствие сбоя при распознавании речи или неправильной интерпретации реплики, принимает клиента имеющего доступ за нарушителя, и ошибки второго рода, когда человек, пытающийся обмануть систему, получает доступ к сотруднику. Учитывая те гарантии, которые дают современные системы распознавания речи, а так же возможности системы «Электронный секретарь», вероятность появления ошибки стоит устанавливать около 0,8 для каждой из переменных.

Заключение

В данной работе была рассмотрена проблема первичной аутентификации при помощи системы «Электронный секретарь» и были предложены пути ее решения. Также была исследована теория речевых актов в ее прикладном значении, и выбрана наиболее подходящая классификация иллокутивных актов в рамках исследуемой проблемы.

Список литературы

1. Страндонсон К. Взаимодействие компьютера и человека на естественном языке / К. Страндонсон, О. Герасименко, Р. Кастерпалу, М. Койт и др. // Труды международной конференции «Диалог 2008». – 2008.
2. Городецкий Б.Ю. Новое в зарубежной лингвистике. Выпуск XVII. Теория речевых актов / Б.Ю. Городецкий, В.З. Демьянкова, Н.М. Кобозева. – М. : Прогресс, 1986. – 424 с.

СОЗДАНИЕ ЭФФЕКТИВНОЙ СИСТЕМЫ БЮДЖЕТНОГО УПРАВЛЕНИЯ С ИСПОЛЬЗОВАНИЕМ ПРОЦЕССНОГО ПОДХОДА

Е.В. Киреевская
Студент

И.В. Абрамов
Доцент

Введение

Успешность деятельности любой организации зависит от эффективно налаженной системы управления. Для того чтобы успешно функционировать на рынке, организациям и, прежде всего их аппарату управления необходимо быстро реагировать на изменения требований потребителей. Современные организации должны отличаться гибкостью, эффективностью, инновационностью и ориентацией на клиента.

Существующая традиционная бюджетная система управления не может отвечать требованиям которые предъявляет рынок, что создает необходимость руководителям многих организаций искать более эффективный инструмент управления.

1. Теоретический анализ

Удовлетворение предъявляемых рынком требований к организации объединено в создании эффективной системы бюджетного управления организации с применением процессного подхода. Главным преимуществом данного подхода является возможность осуществления текущего управления за счет связей между отдельными процессами предприятия. Сегодня процессный подход стал значительно распространен среди многих успешных организаций. Фундаментальный аспект процессного подхода рассматривали в своих трудах многие ученые, в частности Бримсон Джеймс, Антонс Джон, Коллинз Джей, Ефимов В.В., Быков Ю.М., Кондратьев В.В., Кузнецов М.Н. и другие.

Целью исследования является рассмотрение основных аспектов сознания эффективной системы бюджетного управления организации с использованием процессного подхода.

Механизм бюджетирования уже много лет широко используется в мировой практике. Проблемы постановки внутрифирменного бюджетирования в организациях ведущих стран мира были решены 20-25 лет назад, но усовершенствование системы бюджетирования продолжается и сейчас.

Бюджетирование это управленческая технология, которая включает управление финансово-хозяйственной деятельностью субъектов хозяйствования на основе разработки бюджетов, организации контроля за их выполнением, анализа отклонений от бюджетных показателей и регулирования на этой основе хозяйственной деятельности с целью достижения определенных целей [1, 2].

Существует два основных методических подхода к бюджетированию:

1. Построение бюджетов на основе финансовой структуризации;
2. Построение бюджетов на основе выделения бизнес-процессов (видов деятельности) – АВВ-бюджетирование.

С целью обеспечения эффективности и стабильности их деятельности организации, аппарату управления необходимо применять именно ту систему управления, которая бы наиболее точно решала поставленные задачи. На сегодняшний день наиболее гибкой и клиентоориентированной является бюджетная система с применением процессного подхода.

Данный методический подход применяется теми организациями, которые уже используют традиционный метод бюджетирования и внедрили у себя процессное управление. Так как у таких организаций возникает необходимость планировать и оценивать свои процессы,

традиционный метод перестает их устраивать с точки зрения эффективности.

2. Экспериментальная часть

Особенно актуальным процессный подход становится в связи с развитием концепции системы качества, в рамках которой возникает механизм непрерывного совершенствования всех аспектов деятельности предприятия за счет систематического анализа бизнес-процессов.

Для учета принципов процессного подхода в организационном управлении при разработке системы бюджетирования предполагается выполнение ряда последовательных шагов:

1. Выявление потребностей клиентов;
2. Формирование целевых установок;
3. Составление списка необходимых работ по каждой целевой установке;
4. Детализация этих работ на процессы;
5. Определение количества необходимых ресурсов на каждом процессе, т.е. составление бюджетов процессов.

В таблице представлены результаты сравнительного анализа методических подходов к бюджетированию.

Таблица

Сравнительная характеристика бюджетирования для двух подходов построения бизнеса – функционального и процессно-ориентированного

п/п	Характеристика	Функциональный подход к бюджетированию	Процессно-ориентированное бюджетирование
1	2	3	4
1	Главная цель	Эффективное распределение ресурсов между отделами предприятия	Создание стоимости для клиентов и одновременно максимизация экономической добавленной стоимости предприятия
2	Объект концентрации внимания	Основное внимание концентрируется на: - входящих ресурсах - функциональных отделах	Основное внимание концентрируется на: - исходящих результатах - взаимосвязи между отделами, поставщиками, покупателями

Окончание табл.

1	2	3	4
3	Место ведения учета	Учет ведется по функциональным отделам	Учет ведется по видам деятельности
4	Способ планирования	Бюджеты планируются на основе результатов прошлых периодов	Бюджеты планируются в соответствии со стратегией предприятия
5	Методика распределения бюджетных средств	Бюджетные средства распределяются по функциональным подразделениям	Бюджетные средства распределяются по видам деятельности
6	Барьеры между подразделениями	Наличие «стен» между функциональными подразделениями предприятия	Отсутствие «стен» между функциональными подразделениями предприятия
7	Способ формирования цены продажи	Продажная цена определяется затратными факторами	Продажная цена определяется рыночными факторами
8	Взаимосвязь со стратегией предприятия	Отсутствие четкой связи со стратегией предприятия	Четкая связь со стратегией предприятия
9	Оценка эффективности бизнеса	Осуществляется через оценку эффективности структурных подразделений предприятия	Осуществляется через оценку количества и стоимости процессов предприятия

На основе сравнительной характеристики, представленной в таб. 1 можно отметить преобладание преимуществ процессно-ориентированного бюджетирования по сравнению с традиционным. Ключевое значение имеет главная цель подхода к бюджетированию и его соответствие цели

организации, на котором он применяется. Стоит отметить клиентоориентированность бюджетирования с использованием процессного подхода, оно в большей степени соответствует современным рыночным условиям.

Огромное значение при разработке бюджетов имеет способ их планирования. При традиционном бюджетировании бюджеты планируются на основе результатов прошлых периодов, а при процессном в соответствии со стратегией предприятия.

Стоит отметить более четкую связь процессного бюджетирования со стратегией организации по сравнению с традиционным.

Заключение

На основе проведения сравнительного анализа построения систем бюджетирования для двух подходов построения бизнеса, – функционального и процессно-ориентированного подхода было выявлено, что в сложившихся условиях управления и в связи с постоянными колебаниями конъюнктуры рынка отечественным организациям целесообразно применять бюджетирование с использованием процессного подхода.

Список литературы

1. Бримсон Д. Процессно-ориентированное бюджетирование. Внедрение нового инструмента управления стоимостью компании / Д. Бримсон, Д. Антос, Дж. Коллинз; пер. с англ. В.Д. Горюновой; под общ. ред. В.В. Неудачина. – М. : Вершина, 2007. – 336 с.
2. Просветов Г.И. Бюджетирование: задачи и решения / Г.И. Просветов. – М. : Айрис-Пресс, 2010. – 210 с.

ПРИМЕНЕНИЕ СКРЫТЫХ МОДЕЛЕЙ МАРКОВА ДЛЯ РАСПОЗНАВАНИЯ РУКОПИСНЫХ СИМВОЛОВ

Э.Э. Киселёв
Студент

С.А. Запрягаев
Профессор

Введение

В системах on-line рукописного ввода последовательность координат изображения рукописного символа (x, y) составляют набор данных используемых для процедуры распознавания. Хотя технологии

распознавания развиваются уже в течение нескольких десятков лет, проблема распознавания рукописного текста и символов продолжает оставаться предметом исследований. Сложность ее решения определяется вариацией форм символов в зависимости от индивидуальных характеристик человека, а также сложностью сегментации рукописного текста и наличием диакритических знаков и пометок. Набор методов решения такой задачи достаточно широк и не универсален [1]. В связи с этим, разработка инструментов анализа различных методов распознавания для сравнения их характеристик является актуальной задачей и способствует поиску адекватных решений.

Из-за схожести проблемы распознавания рукописного текста и распознавания речи во многих исследованиях успешно применены технологии, разработанные именно для распознавания устной речи [2]. Среди данных технологий скрытые модели Маркова (СММ, Hidden Markov Model – НММ) являются наиболее популярными и достигшими наилучших результатов распознавания.

1. Модель Маркова

Скрытая марковская модель с дискретным временем, конечным пространством состояний и конечным пространством наблюдений состоит из следующих элементов [3]:

1. Конечное пространство скрытых состояний $S = \{S_1, \dots, S_N\}$, где N – их число. Вероятности переходов между состояниями образуют ориентированный граф (рис. 1), который принято задавать матрицей переходов $A = \{a_{ij}\}$ размерностью $N \times N$, где

$$a_{ij} = P[q_{t+1} = S_j | q_t = S_i] \quad (1)$$

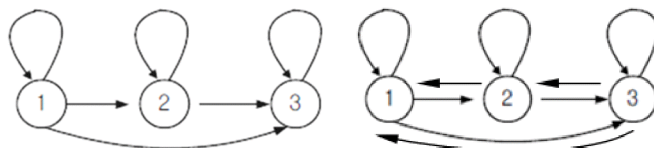


Рис. 1. Общий вид LR-графа переходов (слева) и Full-графа (справа)

2. Последовательность случайных величин (стационарная марковская цепь первого порядка) – «алфавит» наблюдаемой последовательности $V = \{v_1, \dots, v_M\}$, где M – размер «алфавита». Вероятность наблюдения величины v_k в j -ом состоянии определяется матрицей наблюдений $B = \{b_j(k)\}$ размерности $N \times M$, где

$$b_j(k) = P[v_k | S_j] \quad (2)$$

3. Распределение вероятностей начального состояния $\pi = \{\pi_i\}$, где $\pi_i = P[q_1 = S_i]$, то есть вероятность того, что в начальный момент времени система находится в состоянии S_i .

Совокупность значений $\lambda = (A, B, \pi)$ представляют собой скрытую марковскую модель, которая может сгенерировать наблюдаемую последовательность.

Задачи, которые можно решить с помощью СММ делятся на три типа:

1. Имеется наблюдаемая последовательность и СММ. Требуется определить, с какой вероятностью эта последовательность может быть сгенерирована данной СММ. Основные алгоритмы решения: алгоритмы «Вперед» и «Назад».

2. Имеется наблюдаемая последовательность и СММ. Требуется определить наиболее вероятную последовательность скрытых состояний, которая вероятнее всего описывает данную наблюдаемую последовательность. Основной алгоритм решения: алгоритм Витерби.

3. Имеется последовательность наблюдений (или несколько последовательностей). Требуется подобрать параметры СММ $\lambda = (A, B, \pi)$ таким образом, чтобы вероятность генерации этой (этих) последовательности была максимальной при заданных параметрах. Основной алгоритм решения – алгоритм Баума-Велша.

2. Описание модели

СММ нашли широкое применение в технологиях распознавании образов. Вероятностная модель Маркова похожа на методику использования нейронных сетей [4] для классификации объектов. Сходство заключается в том, что СММ, также как и нейросеть достаточно сложна в конфигурации. Правильный подбор параметров модели $\lambda = (A, B, \pi)$ является приоритетной задачей, от решения которой зависит качество распознавания (третья задачи СММ). Из данной проблемы вытекает немаловажный аспект обучения модели. Для этого на вход необходимо подать несколько объектов, принадлежащих одному и тому же классу.

Для описания объектов используются не сами физические экземпляры, а выделенные специальным образом наборы признаков. Эти наборы являются атрибутикой, рассматривающие объект относительно другой системы координат и не несут в себе информацию, теоретически достаточную для классификации.

Для каждого из класса символов СММ обучается с собственными

параметрами $\lambda_i = (A_i, B_i, \pi)$, где $i \in N$, N – количество классов.

После «тренировки» (оптимизации) всех СММ при помощи алгоритма Баума-Велша, на вход каждой модели подается распознаваемая последовательность наблюдений (набор признаков, выделенный из рукописного символа). На этом этапе решается первая задача СММ: при помощи алгоритма «Вперед» определяется вероятность генерации данной последовательности каждой из моделей. Решение о принадлежности вектора признаков к тому или иному классу принимается исходя из максимума оценки правдоподобия (Maximum Likelihood Estimation).

3. Результаты численного эксперимента

В данной работе численный эксперимент был проведен на примере распознавания рукописных арабских цифр. Для выделения признаков в данной статье использовался метод, описанный в [5].

Представим поверхность ввода в виде координатной плоскости. Тогда направление минимального приращения координат (в дискретном случае это расстояние между двумя соседними точками ввода) будет иметь некоторое направление на координатной плоскости.

Разобьем координатную плоскость на 8 равных секторов (рис. 2). В общем случае количество секторов может быть произвольным.

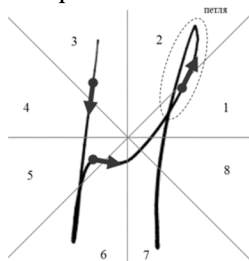


Рис. 2. Определение векторов направления на примере буквы «н»

Направление минимального приращения координат будет лежать в одном из секторов координатной плоскости. Таким образом, вместо абсолютных координат, можно получить набор цифр (в данном случае они не несут никакой математической нагрузки, а являются символьными обозначениями): 1, 2, 3, 4, 5, 6, 7 или 8. Они указывают направление вектора градиента для двух соседних точек. Эти направления обозначают один из восьми секторов координатной плоскости. Очевидно, что если начертание символа не меняет направление, получим последовательность одинаковых чисел. Как только начертание сменит направление – номер сектора также изменится. Объединяя точки с одним направлением движения в единственное значение, получим вектор признаков, имеющий

размерность гораздо меньшую, чем исходная последовательность точек ввода.

Для обучения использовались 3 вектора признаков для каждого из символов. Важно отметить, что в данном примере используются одинаковые начальные (инициализирующие) параметры системы: количество состояний, вид графа переходов, количество векторов для тренировки. В общем случае этот факт не должен влиять на качество распознавания [6].

Матрицы A и B до применения алгоритма Баума-Велша также могут быть заданы случайным образом. Однако практические исследования показали, что начальные значения изменяемых параметров моделей влияют на качество распознавания (определения). Чем ближе начальные вероятности к реальным, тем точнее будет обучена СММ, что увеличит качество распознавания.

Экспериментальные результаты распознавания рукописных арабских цифр представлены в таблицах, которые показывают количество верно распознанных векторов-признаков цифр из четырех введенных. В каждом из исследований менялся один из параметров (количество скрытых состояний, начальные условия, тип графа). В ходе эксперимента сравнивались результаты распознавания и были выявлены следующие зависимости:

1. Число правильно распознанных векторов-признаков (максимум 4) в зависимости от числа N скрытых состояний СММ $S = \{S_1, \dots, S_N\}$ (таблица 1). Для каждой строки – одинаковые начальные условия $\pi = \{1, 0, 0\}$ и тип графа (LR-граф), но разное количество скрытых состояний.

Таблица 1

Число правильно распознанных векторов-признаков в зависимости от числа N скрытых состояний

Кол-во состояний	Расознаваемый символ									
	«0»	«1»	«2»	«3»	«4»	«5»	«6»	«7»	«8»	«9»
$N = 2$	4	4	4	2	4	3	4	4	4	4
$N = 3$	4	4	4	2	4	4	4	4	4	4
$N = 4$	4	4	4	2	4	4	4	4	4	4

2. Число правильно распознанных векторов-признаков (максимум 4) в зависимости от начальных условий $\pi = \{\pi_i\}$ (таблица 2). Для каждой строки – одинаковые количества состояний (по 3) и тип графа (LR-граф), но разные начальные условия.
- 3.

Таблица 2

Число правильно распознанных векторов-признаков в зависимости от начальных условий

$\pi = \{\pi_i\}$	Распознаваемый символ									
	«0»	«1»	«2»	«3»	«4»	«5»	«6»	«7»	«8»	«9»
{1 0 0}	4	4	4	2	4	4	4	4	4	4
{0,34 0,33 0,33}	4	4	4	2	4	4	4	4	4	4

4. Число правильно распознанных векторов (максимум 4) в зависимости от типов графов переходов (LR-граф, Full-граф – рис. 1, таблица 3). Для каждой строки – одинаковые количества состояний (по 3) и начальные условия $\pi = \{1, 0, 0\}$, но разные типы графов.

Таблица 3

Число правильно распознанных векторов-признаков в зависимости от типов графов переходов

Тип графа	Распознаваемый символ									
	«0»	«1»	«2»	«3»	«4»	«5»	«6»	«7»	«8»	«9»
LR-граф	4	4	4	2	4	4	4	4	4	4
Full-граф	4	4	1	0	2	1	4	2	3	2

Заключение

Выполненное исследование показало, что для эффективного распознавания при помощи СММ предпочтительнее использовать схожие параметры модели для каждого из классов.

Из таблицы 1 видно, что качество распознавания не существенно, но зависит от количества состояний СММ: чем больше состояний, тем лучше качество распознавания. В аналогичном пространстве задач, связанных с идентификацией устной речи рекомендуется использовать 5 состояний для получения максимального качества. При увеличении количества состояний качество распознавания не улучшается.

Таблица 2 свидетельствует о том, что распределение начальных вероятностей не оказывает существенного влияния на качество распознавания.

Существенное влияние на решение задачи оказывает выбор направлений переходов между состояниями СММ. Таблица 3 показывает, что Left-Right модель графа обеспечивает более высокое качество, чем полностью связанный граф.

Исследование показало, что СММ являются эффективным средством для решения задачи распознавания рукописного текста и

подтвердило проблему сложной настройки системы для конкретной области применения.

Список литературы

1. Бутаков Б.А. Обработка изображений на ЭВМ / Б.А. Бутаков, В.Н. Островский, И.Л. Фадеев. – М. : Радио и связь, 1987. – 240 с.
2. Bardici N. Speech Recognition using Hidden Markov Model / N. Bardici, B. Skarin. – Thesis ... degree of Masters of Science Electrical Engeneering. – Blekinge: Blekinge Institute of Technology, 2006. – 79 p.
3. Rabiner L.R. A tutorial on hidden markov models and selected applications in speech recognition / L.R. Rabiner, H. Murray // Proceedings of the IEEE. – 1989. – v. 77. – №2. – P. 257–286.
4. Хайкин С. Нейронные сети: полный курс / С. Хайкин. – М. : Вильямс, 2006. – 114 с.
5. Киселёв Э.Э. Система on-line распознавания рукописного текста / Э.Э. Киселёв // Информатика: проблемы, методология, технологии: материалы XII Международной научно-методической конференции, 9-10 февраля 2012 г. – Воронеж, 2012. – Т. 1. – С. 173–175.
6. Introduction to Hidden Markov Model and Its Application. KISS ILVB Tutorial. – Режим доступа: <http://isoft.postech.ac.kr/Course/CS704/LectureNotes/HiddenMarkovModel.pdf>

РЕАЛИЗАЦИЯ АЛГОРИТМА STRUCTURED FOREST ДЛЯ ОБНАРУЖЕНИЯ ГРАНИЦ НА ИЗОБРАЖЕНИЯХ

К.В. Климов
Студент

Н.А. Тюкачев
Доцент

Введение

Обнаружение границ на изображениях — это классическая задача в машинном зрении. Под границами обычно подразумеваются границы объектов. Классические подходы к решению данной задачи основаны на вычислении градиентов изображения, и последующей их обработке. Пример такого алгоритма — детектор границ Canny. Такой подход прост в реализации, и обладает высокой производительностью. Однако границы, получаемые таким образом, не всегда совпадают с границами реальных объектов. При этом есть проблемы с отделением границ сложной текстуры объекта от границ самого объекта, требуется настройка пороговых

значений для разных сцен и т.д. В 2013 году Piotr Dollar из MS Research предложил новый алгоритм для обнаружения границ на основе алгоритма машинного обучения Structured forest. Ключевая идея алгоритма заключается в обучении детектора на изображениях, границы которых были размечены людьми. В данной работе, для обучения, использовались размеченные изображения из набора BSDS500.

В данной работе описывается сам алгоритм, и особенности его реализации на языке программирования C++ с использованием библиотек OpenCV и Intel TBB.

1. Анализ задачи

Согласно [1] на вход уже обученному алгоритму подается цветное изображение в формате RGB, на выходе имеем «карту границ» – изображение, в котором нулевое значение пикселя означает отсутствие границы в данной точке, значение отличное от нуля — границу. Обнаружение границ происходит в 4 этапа:

2. вычисление каналов дескриптора от входного изображения;
3. заполнение выходной карты границ по принципу — с некоторым шагом (в данной работе — 4 пикселя) по изображению проходит скользящее окно, содержимое окна отдается на вход Structured forest, полученные на выходе алгоритма части карты границ накладываются на соответствующие места выходной карты;
4. если значение какой-то точки выходной карты меньше порога — эта точка приравнивается к 0;
5. полученная карта сглаживается любым подходящим фильтром, к примеру, фильтром Гаусса.

Обучение Structured forest происходит аналогично обучению классического алгоритма машинного обучения Random forest, с некоторыми особенностями, описанными ниже.

2. Дескриптор

Дескриптор — вектор вещественных значений, получаемый из квадратной области изображения размером 32 x 32, называемой «патч». Дескриптор используется в качестве входного значения для Structured forest. Состоит из 13 каналов (каждый канал — матрица содержащая вещественные значения) размером 16 x 16, что дает 3228 элементов дескриптора, и из попарных разниц элементов этих каналов уменьшенных до размеров 5 x 5. Размер дескриптора с учетом попарных разностей — 7228 элементов.

Каналы вычисляются следующим образом:

- 1) первые три канала — изображение переведенное в цветовое пространство CIE-LUV. Размер изображения уменьшен вдвое;

2) магнитуды градиентов полученные из трех каналов которые были вычислены в пункте 1. Формируется только один канал, на основе трех каналов градиентов (по одному на каждый канал CIE-Luv), по принципу — для каждой точки выбирается максимальное значение градиента из трех возможных, и соответствующее ей значение направления;

3) направления градиентов разбитые на 4 канала по интервалам углов, соответственно, от 0 до 90, от 90 до 180, от 180 до 270 и от 270 до 360;

4) магнитуды и направления взятые от исходного изображения с разрешением уменьшенным в два раза — 5 каналов.

Для ускорения работы дескриптора, а так же получения адекватных значений градиентов на границах патчей сначала для всего изображения считаются каналы, и только затем вырезаются отдельные патчи, и считаются попарные разности.

3. Structured forest

Согласно [2], Structured forest — алгоритм машинного обучения основанный на алгоритме Random forest (случайный лес решающих деревьев). Ключевое отличие Structured forest в том, что вместо класса, вероятности или другого скалярного значения алгоритм сопоставляет входному вектору некоторую структуру. В данном случае это — карта границ размером 16 x 16.

4. Обучение

В связи с тем, что классические алгоритмы обучения решающих деревьев требуют, чтобы каждый элемент обучающей выборки был отмечен некоторым классом, в [1] была предложена процедура дискретизации, которая ставит в соответствие каждой структуре в обучающей подвыборке номер класса (0 или 1). В качестве обучающей выборки используются набор пар – цветное изображение-карта сегментов, где карта сегментов это изображение в котором каждая точка соответствует номеру сегмента которому она принадлежит. Из карты сегментов можно легко получить карту границ, и в тоже время карты сегментов намного проще кластеризовать.

5. Дискретизация

Данная процедура предназначена для кластеризации карт границ на этапе обучения. На этапе обнаружения границ она не используется.

Идея заключается в отображении набора карт границ в набор векторов вещественных значений, с последующим отображении набора векторов в набор номеров классов (0 или 1).

Отображение карт границ в набор векторов происходит следующим образом:

Выбираются 256 случайных пар индексов, если в карте сегментов соответствующие индексам точки имеют одинаковые значения, то соответствующий элемент вектора равен 1, иначе 0. Среди полученных векторов находится среднее от суммы элементов. Далее полученное значение делится на 256 и вычитается из всех векторов.

Отображение полученных векторов в номера классов осуществляется с использованием метода главных компонент (РСА). Размерность векторов уменьшается с помощью данного метода следующим образом: у полученных векторов отбрасываются все значения кроме первого и эти значения бинаризируются по принципу — если значение меньше 0, то ему в соответствие ставится 1, иначе — 0.

6. Реализация

Для реализации были использованы следующие средства:

- 1) язык программирования C++, позволяющий писать высокопроизводительные программы;
- 2) библиотека машинного зрения OpenCV, предоставляющая реализацию большого количества базовых алгоритмов;
- 3) библиотека параллельного программирования Intel TBB;
- 4) редактор текста Vim.

Для ускорения обучения был предложен следующий подход к распараллеливанию. Так как построение поддерева в процессе обучения (согласно [3]) не зависит от построения других поддеревьев, существует возможность выполнять это построение параллельно. Для этого, в данной работе используется механизм `task_group` из библиотеки TBB.

Заключение

В результате работы был реализован алгоритм Structured forest для обнаружения границ, включая дескриптор, Structured forest, алгоритм обучения Structured forest, а также его параллельная версия и сам алгоритм обнаружения границ.

Так же было осуществлено обучение с использованием набора данных BSDS500, на выборке из 200 000 примеров.

Полученные результаты несколько уступают результатам полученным Piotr Dollar, однако в основном это связано с небольшим объемом обучающей выборки.

Список литературы

1. Dollar P. Structred forests for fast edge detection / P. Dollar, C. Lawrence Zitnick // Proceedings ICCV, International Conference on Computer Vision. – 2013. – P. 1841–1848.

2. Kotschieder P. Structured class-labels in random forests for semantic image labeling / P. Kotschieder, S. Rota Bul'o, H. Bischof, and M. Pelillo // Proceedings ICCV, International Conference on Computer Vision. – 2011. – P. 2190–2197.

3. Leo Breiman. Random Forests. – Режим доступа: <http://oz.berkeley.edu/~breiman/randomforest2001.pdf>

ПРОЕКТИРОВАНИЕ АРХИТЕКТУРЫ АВТОМАТИЗИРОВАННЫХ ТЕСТОВ ПОЛЬЗОВАТЕЛЬСКОГО ИНТЕРФЕЙСА WEB-ПРИЛОЖЕНИЯ

Т.В. Колесникова
Студент

И.В. Илларионов
Доцент

Введение

В значительной мере оптимизировать процесс автоматизации тестирования позволяет разработка уникальной архитектуры программной системы. Подобная архитектура обеспечивает сокращение времени создания автоматизированных тестов, приводит к уменьшению количества ресурсов, затрачиваемых на модификации, требуемые в случае изменения структуры тестируемого программного обеспечения, позволяет значительно снизить трудозатраты на поддержку автоматизированного тестирования и многое другое.

Таким образом, задача разработки определенной архитектуры, которая будет использоваться в результате для написания, выполнения и оценки автоматизированных тестов графического пользовательского интерфейса веб-приложения является одним из определяющих факторов при оценке эффективности автоматизации.

1. Архитектура автоматизированного тестирования

Процесс проектирования архитектуры автоматизированных тестов обычно включает в себя [1, 2]:

1) Принятие решения, нужны ли для данного компонента автоматизированные тесты или трудозатраты на их написание и последующую поддержку будут несоизмеримы с ручным прогоном. Это важнейшая фаза, от которой зависит то, как будет выстроен процесс тестирования компонента в дальнейшем.

© Колесникова Т.В., Илларионов И.В., 2014

2) Если результаты первого пункта привели к решению создания автоматизированных тестов, то следующим этапом является обоснованный выбор инструментов для их написания.

3) Детальное проектирование проекта, которое подразумевает принятия решения по которому выделяются модули в архитектуре автоматизированных тестов, организацию взаимодействия с тестируемым компонентом, объединение с системами непрерывной интеграции.

4) Создание прототипа. Его создание обычно не требует много времени и трудозатрат, однако прототип может наглядно продемонстрировать все плюсы и минусы используемого подхода. Их анализ позволяет оперативно принять меры по устранению проблем на очень раннем этапе, потратив значительно меньше времени, чем если эти проблемы будут выявлены на стадии уже созданных тестов.

2. Детальное проектирование

В основу предложенной архитектуры положен принцип разделения высокоуровневой тестовой логики от низкоуровневой логики поиска конкретных элементов пользовательского интерфейса. Для его реализации использовались техники объектно-ориентированного программирования.

Для автоматизации тестирования был выбран инструмент Selenium WebDriver – драйвер, предоставляющий серию низкоуровневых команд для управления браузером.

Как показано на рисунке, предложенная архитектура автоматизированных тестов состоит из трех уровней: обёртка для Selenium WebDriver, доступ к Web-страницам, непосредственно тестовые сценарии.

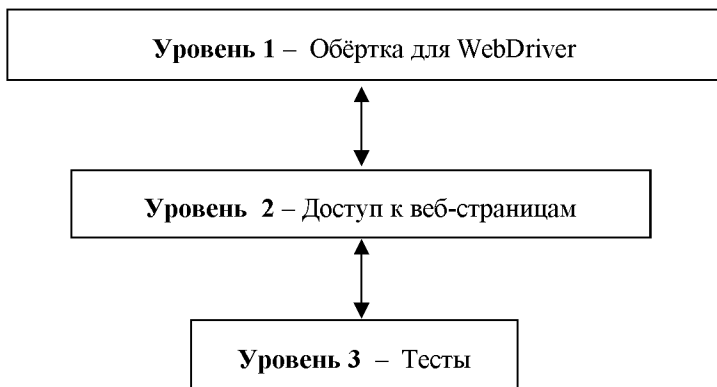


Рисунок. Блок-схема архитектуры автоматизированных тестов пользовательского интерфейса web-приложения

Первый уровень включает классы, которые используются как обёртка для функций Selenium WebDriver. Данные классы содержат методы для взаимодействия с элементами веб-страницы (клик по кнопке, ввод текста в текстовое поле, ожидание завершения ajax-запроса и др.). Этот уровень также содержит такие классы, как логгеры, классы, используемые для работы с конфигурационным файлом, генерации тестовых данных, отправки http запросов на сервер и т.д.

Второй уровень содержит код, используемый для доступа к отдельным веб-страницам. Например, тестируемое веб-приложение включает страницу для авторизации, страницу создания события, страницу профиля и т.д. Классы содержат код, который может выполнять действия в тестируемом веб-приложении, такие как метод авторизации, метод создания события, метод редактирования профиля и т.д. Такие классы можно назвать структурными модулями.

Третий и верхний уровень – это тестовый уровень. В своей основе этот уровень содержит все тест-кейсы со всеми данными, необходимыми для выполнения тестов.

Методы третьего уровня вызывают методы второго уровня, которые, в свою очередь, вызывают методы первого уровня. Например, сам тест может вызывать метод login (username, password) и передавать username – “testuser1” и пароль “secret”. Это означает, что легко могут быть переданы разные наборы данных.

В результате, предложенная архитектура автоматизированных тестов веб-приложения позволила [3]:

- упростить поддержку автоматизированных тестов;
- сделать тесты независимыми друг от друга;
- добиться простоты чтения тестовых скриптов;
- обеспечить структурирование программного кода: отделить низкоуровневую логику выполнения действий в браузере от высокоуровневой логики тестов;
- устранить дублирование кода за счет создания отдельных классов для каждой страницы (компонента), выноса логики работы со страницей в отдельные общедоступные методы и их переиспользования;
- выполнять тесты последовательно в одном экземпляре браузера без его перезапуска и сэкономить за счет этого время с одной стороны, при этом группировка тестов по классам позволила перезапускать браузер при инициализации каждого класса, чтобы избежать зависаний при выполнении большого количества тестов;
- вести прозрачное логирование выполняемых тестом действий.

Заключение

Таким образом, предлагаемая трехуровневая архитектура автоматизированных тестов позволяет при необходимости дополнять и изменять функциональность тестов, а также подменять модули без переделывания остальных компонентов архитектуры. Что в свою очередь обеспечивает, в случае изменения каких-либо алгоритмов в работе тестируемого приложения, возможность быстрой локализации ошибки тестируемого приложения, а так же реализацию небольших исправлений автоматизированных тестов без модификации большого количества программного кода.

Список литературы

1. Дастин Э. Автоматизированное тестирование программного обеспечения : Внедрение, управление и эксплуатация / Э. Дастин, Д. Рэшка, Д. Пол; пер. с англ. Е. Молодцовой, М. Павлова; под ред. А. Головки. – М. : ЛОРИ, 2003. – 567 с.
2. Тамре Л. Введение в тестирование программного обеспечения / Л. Тамре; пер. с англ. и ред. В.В. Марченко. – М. : Вильямс, 2003. – 359 с.
3. Синицын С.В. Верификация программного обеспечения : учебное пособие / С.В. Синицын, Н.Ю. Налютин. – М. : Бином. Лаборатория знаний, 2008. – 367 с.

ОЦЕНКА НАДЕЖНОСТИ УДОСТОВЕРЯЮЩЕГО ЦЕНТРА

М.И. Конотоп

Студент

С.А. Вялых

Доцент

Введение

Удостоверяющий центр (УЦ) – юридическое лицо или индивидуальный предприниматель, осуществляющие функции по созданию и выдаче сертификатов ключей проверки электронных подписей, а также иные функции, предусмотренные федеральным законом № 63-ФЗ [1].

Нарушить работоспособность УЦ можно, нарушив работоспособность одного из базовых элементов УЦ:

- Центр сертификации;
- Центр регистрации;
- Сервер публикации отозванных сертификатов (СОС) для

пользователей УЦ.

В большинстве УЦ доступ к Центру регистрации и Центру сертификации из не защищенных сетей запрещен, поэтому для оценки надежности УЦ необходимо проводить анализ СОС.

Реализация

Рассмотрим следующие варианты реализации СОС:

- 1) СОС функционирует под управлением операционной системы (ОС) Windows Server 2008;
- 2) СОС функционирует под управлением ОС Ubuntu Linux 11.10;
- 3) В системе функционируют два параллельных СОС один под управлением ОС Windows Server 2008, другой под управлением ОС Ubuntu Linux 11.10.

При этом на каждом из СОС отсутствуют средства защиты информации (СЗИ), препятствующие непосредственному негативному воздействию на программное обеспечение (ПО), используются только встроенные в ПО средства защиты с настройками «по умолчанию» и установлено следующее ПО:

- WEB-сервер Apache 2.2.16;
- Система управления WEB-контентом Туро3 4.5.6.

Представим процесс появления новых уязвимостей и их устранения как процесс функционирования системы массового обслуживания (СМО) [2]. На вход СМО поступает нестационарный пуассоновский поток заявок (уязвимостей) с интенсивностью $\lambda(t)$, зависящей от времени t Поток уязвимостей является нестационарным пуассоновским, так как фактически он представляет собой сумму порядка $100 \div 1000$ независимых нестационарных потоков с приблизительно одинаковой интенсивностью [3], порождаемых специалистами, занимающимися поиском новых уязвимостей. Далее СМО обслуживает заявки (устраняет заявки) с интенсивностью μ , рассчитывается выражением (1):

$$\mu = k\mu_v, \quad (1)$$

где μ_v – средняя скорость создания вендором патча, устраняющего уязвимость в данной программе, а k – коэффициент, характеризующий работу системного администратора.

Оценку μ_v будем производить по выражению (2):

$$\mu_v = \frac{1}{T_v}, \quad (2)$$

где T_v – среднее время создания вендором патча, закрывающего уязвимость, после ее обнаружения. Оценку μ_v следует производить для каждой программы отдельно, так как один и тот же вендор зачастую создает патчи для разных программ с разной скоростью.

Коэффициент k предлагается оценивать экспертным путем [4]:

- $k = 0$, если системный администратор вообще не устанавливает патчи, выпущенные вендором;
- $k < 1$, если системный администратор не своевременно устанавливает патчи, выпущенные вендором;
- $k = 1$, если системный администратор устанавливает патчи, выпущенные вендором, сразу же после их выпуска;
- $k > 1$, если системный администратор устанавливает патчи, выпущенные вендором, сразу же после их выпуска и при этом сам предлагает временные решения для устранения некоторых уязвимостей.

Среднестатистическое число уязвимостей в программе на момент времени t рассчитывается выражением (3) [4]:

$$N_{\text{cp}}(t) = \frac{T_{\text{в}} e^{-t}}{k} \left(\lambda(t) + \int_0^t \lambda(\tau) e^{\tau} d\tau \right). \quad (3)$$

В граничном случае, если коэффициент работы системного администратора $k=0$, то уязвимости, найденные в программе, не закрываются, а только накапливаются. Следовательно, в этом случае среднестатистическое число уязвимостей в системе равно среднестатистическому числу уязвимостей, обнаруженных в программе за время ее жизни (4):

$$N_{\text{cp}}(t) = \int_0^t \lambda(\tau) d\tau. \quad (4)$$

При данных предположениях вероятность того, что в конкретной программе отсутствуют уязвимости равна (5):

$$P_0(t) = e^{-N_{\text{cp}}(t)}. \quad (5)$$

Поскольку для функционирования УЦ требуется несколько разных программ, то простейшая математическая модель функционирования системы в условиях внутренних уязвимостей и конфликтных взаимодействий может быть представлена как совокупность систем массового обслуживания, каждая из которых моделирует динамику уязвимостей в каждой отдельной программе. Данная модель представлена на рис. 1, здесь $\lambda^{(m)}(t)$ – скорость обнаружения уязвимостей в m -й программе, $k^{(m)}$ – коэффициент, характеризующий обслуживание системным администратором m -й программы, $T_{\text{в}}^{(m)}$ – среднее время создания вендором патча, зарывающего уязвимость, после ее обнаружения в m -й программе, а M – общее число программ [4].

В этом случае среднестатистическое число уязвимостей в УЦ будет суммой среднестатистического числа уязвимостей в каждой программе, установленной в УЦ:

$$N_{\text{cp}}(t) = \sum_{m=1}^M N_{\text{cp}}^{(m)}, \quad (6)$$

где $N_{\text{cp}}^{(m)}$ определяется по формуле (3).

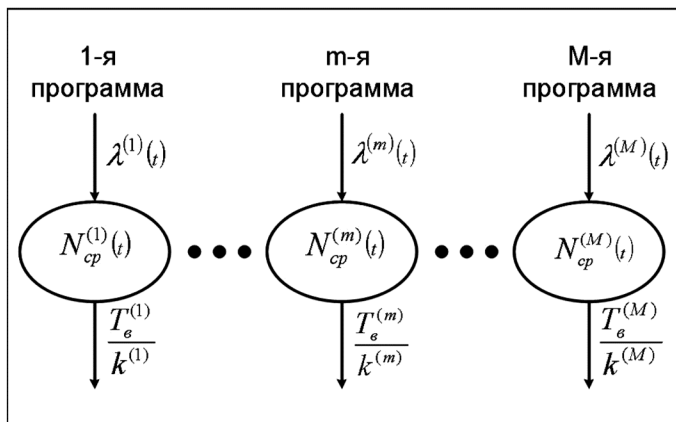


Рис. 1. Простейшая математическая модель функционирования УЦ (без СЗИ) в условиях внутренних уязвимостей и конфликтных взаимодействий

В случае, когда уязвимости каждой программы могут быть использованы непосредственно для негативного воздействия на УЦ, потенциальная вероятность того, что надежность УЦ в момент времени t не может быть нарушена, совпадает с вероятностью отсутствия в УЦ уязвимостей (7):

$$P_{\text{над}}(t) = P_0(t). \quad (7)$$

В таблице 1 приведены среднее время и скорость создания (определяется выражением (2)) вендором патча [5], закрывающего уязвимость, после ее обнаружения для ПО установленного на СОС в 1-3 вариантах.

Таблица 1

Среднее время и скорость создания вендором патча

	Windows Server 2008	Ubuntu Linux 11.10	WEB-сервер Apache 2.2.16	Система управления WEB-контентом Туро3 4.5.6
T_B (месяца)	0,51	0,22	0,05	0,04
μ_B	1,96	4,55	20	25

В таблице 2 приведена статистика обнаружения уязвимостей в ПО СОС за 2013 год [5].

Таблица 2

Статистика интенсивности обнаружения уязвимостей в ПО типового СОС за 2013 год

Месяц \ ПО	Windows Server 2008	Ubuntu Linux 11.10	WEB-сервер Apache 2.2.16	Система управления WEB-контентом Туро3 4.5.6
январь 2013	10	4	0	0
февраль 2013	52	66	2	0
март 2013	14	27	0	2
апрель 2013	6	11	0	0
май 2013	5	15	2	0
июнь 2013	3	2	1	0
июль 2013	11	15	3	4
август 2013	8	2	0	0
сентябрь 2013	10	20	1	0
октябрь 2013	7	3	1	0
ноябрь 2013	6	23	0	0
декабрь 2013	6	16	0	0

Коэффициент работы системного администратора примем от 0 до 1 с шагом 0,5. Если системный администратор не устанавливает патчи, выпущенные вендором, то $k=0$. Если системный администратор устанавливает патчи, выпущенные вендором, сразу же после их выпуска (т.е. настроены авто обновления ПО), то $k=1$. Если системный администратор не своевременно устанавливает патчи, выпущенные вендором, то $k=0,5$.

По формуле (3) и (4) определим среднестатистическое число уязвимостей в ПО, установленном на СОС за 2013 год, полученные значения приведены в таблице 3.

Таблица 3

Среднестатистическое число уязвимостей в ПО сервера публикаций отозванных сертификатов за 2013 год

ПО \ k	Windows Server 2008	Ubuntu Linux 11.10	WEB-сервер Apache 2.2.16	Система управления WEB-контентом Туро3 4.5.6
0	138	204	10	6
0,5	3,712	4,213	0,0077	0,0008
1	1,856	2,107	0,0039	0,0004

По формуле (5) определим вероятность отсутствия уязвимостей в ПО СОС за 2013 год, полученные значения приведены в таблице 4.

Таблица 4

Вероятность отсутствия уязвимостей в ПО СОС за 2013 год

ПО k	Windows Server 2008	Ubuntu Linux 11.10	WEB-сервер Apache 2.2.16	Система управления WEB-контентом Type3 4.5.6
0	0	0	0,000045	0,0025
0,5	0,024	0,015	0,992	0,999
1	0,156	0,122	0,996	1

Для 1 и 2 варианта СОС, надежность будет совпадать с вероятностью отсутствия уязвимостей в ПО, и будет определяться выражением (8):

$$P_{\text{над}}(t) = \prod_{m=1}^M P_0^{(m)}(t), \quad (8)$$

где $P_0^{(m)}(t)$ – вероятность отсутствия уязвимости в m -й программе.

Выведем выражение расчета надежности для 3го варианта реализации СОС. Для 2-х параллельных СОС вероятность надежности будет равна вероятности события, что хотя бы на одном из СОС отсутствуют уязвимости. При этом, если найдена уязвимость в одинаковых программах установленных на каждом из 2-х СОС, то она может быть использована для нарушения работоспособности обоих СОС одновременно. Введем события:

A_0 – отсутствие уязвимостей на 1м СОС;

B_0 – отсутствие уязвимостей на 2м СОС;

A – уязвимость есть на 1м СОС;

B – уязвимость есть на 2м СОС;

A_{00} – отсутствие уязвимостей в ОС на 1м СОС;

B_{00} – отсутствие уязвимостей в ОС на 2м СОС;

A_{01} – уязвимости есть в 1й программе на 1м СОС;

B_{01} – уязвимости есть в 1й программе на 2м СОС;

A_{02} – уязвимости есть во 2й программе на 1м СОС;

B_{02} – уязвимости есть во 2й программе на 2м СОС.

Вероятность наступления события A_0 определяется выражением (9):

$$P(A_0) = P(A_{00})P(A_{01})P(A_{02}), \quad (9)$$

аналогично для B_0 .

Так как, прикладные программы на 1 и 2 СОС одинаковые то:

$$P(A_{01}) = P(B_{01}),$$

$$P(A_{02}) = P(B_{02}). \quad (10)$$

Вероятность отсутствия уязвимости в СОС тогда будет определяться выражением (11):

$$P_0 = P(A_0B_0) + P(AB_0) + P(A_0B). \quad (11)$$

Используя теоремы о вероятности совместного наступления двух зависимых событий и о вероятности наступления двух зависимых событий, а также выражения (10) и (11), получим выражение (12):

$$P_0 = (P(A_{00}) + P(B_{00}) - P(A_{00})P(B_{00})) P(A_{01}) P(A_{02}). \quad (12)$$

Выражение (12) позволяет определить надежность для 3го варианта реализации СОС.

По выражениям (8) и (12) определим вероятность надежности СОС за 2013 год для 1-3 варианта реализации, полученные значения приведены в таблице 5.

Таблица 5

Вероятность надежности СОС за 2013 год

k \ Вариант	1	2	3
0	0	0	0
0,5	0,024	0,015	0,038
1	0,155	0,122	0,258

На рис. 2 приведена вероятность нахождения СОС за 2013 год. В зависимости от коэффициента работы системного администратора.

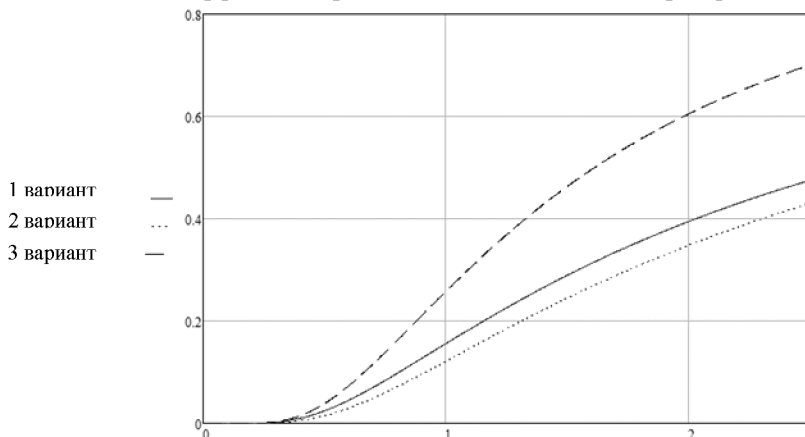


Рис. 2. Вероятность нахождения СОС в надежном состоянии за 2013 год

Заключение

Самым надежным является третий вариант реализации СОС, с использованием параллельных серверов под управлением разных ОС.

Наименее надежным является второй вариант реализации СОС, с использованием сервера под управлением ОС Ubuntu Linux 11.10. Таким образом, дублирование СОС повышает надежность УЦ в 2,11-2,53 раза.

Список литературы

1. Федеральный закон Российской Федерации от 6 апреля 2011 г. № 63-ФЗ «Об электронной подписи».
2. Саати Т. Элементы теории массового обслуживания и ее приложения / Т. Саати. – М. : Мир, 1991. – 397 с.
3. The Open Source Vulnerability Database. – Режим доступа: <http://osvdb.org>
4. Вялых А.С. Модели и алгоритмы анализа и прогнозирования надежности использования программного обеспечения информационных систем в условиях конфликтных взаимодействий: автореф. дис. ... канд. тех. наук / А.С. Вялых – Воронеж, 2014. – 16 с.
5. National Vulnerability Database // National Institute of Standards and Technology. – Режим доступа: <http://nvd.nist.gov>

АВТОМАТИЧЕСКОЕ ФОРМИРОВАНИЕ И ПОСТРОЕНИЕ ОНТОЛОГИЙ ПРЕДМЕТНОЙ ОБЛАСТИ ТЕКСТОВ НА ЕСТЕСТВЕННОМ ЯЗЫКЕ

А.П. Константинова
Студент

В.В. Гаршина
Доцент

Введение

В наше время все чаще встречаются термины компьютерная лингвистика, искусственный интеллект, Text Mining. Это связано с разработкой различных приложений, с активным ростом текстовой информации в сети Интернет, а также с растущей необходимостью автоматической обработки текстов на естественном языке.

В данной статье речь пойдет об одном из главных направлений компьютерной лингвистики совместно с искусственным интеллектом — автоматическом получении онтологии предметной области текста на естественном языке. Реализация данной задачи осуществляется на основе нового метода — полного лингвистического анализа текста и извлечении основной информации из него.

1. Актуальность задачи автоматического построения онтологий. Технология Text Mining

Интернет — это огромное информационное поле, которое содержит подробные сведения абсолютно разного рода. Однако, практически вся информация, доступная во всемирной паутине, не содержит семантики. Ее поиск и интеграция в рамках конкретной предметной области затруднены. Встает жесткая необходимость интеллектуализации, интеграцию данных в сети Интернет.

Достаточно эффективные средства решений такого рода проблем относятся к подходу Semantic Web. Например, использование онтологий, которые получили широкое распространение в решении данного рода проблем. Однако, построение онтологий требует больших знаний в исследуемой предметной области. Поэтому до сих пор остается актуальной задача автоматического построения онтологий. Реализация данной задачи опирается на технологию Text Mining.

Технология Text Mining представляет собой набор методов, которые предназначены для извлечения информации из текстов. Основная цель данной технологии — дать человеку возможность работать с большими объемами исходных данных за счет автоматизации процесса извлечения нужной информации.

Работа данной технологии заключается в том, что на вход подается текст, который написан на естественном языке, а на выходе пользователь получает запрошенную информацию в структурированном виде. Принято выделять 5 стадий в данной технологии:

1. Сбор текстовой информации.
2. Использование статистических и лингвистических методов анализа информации.
3. Выявление ключевых понятий («сущностей»).
4. Выявление взаимосвязей.
5. Интерпретация результатов — представление на естественном языке или в графическом виде [1].

Онтология (обобщ.) — это всеобъемлющая и детальная формализация некоторой области знаний. Основное свойство онтологий — интеграция информации.

Опираясь на работы знаменитых ученых и лингвистов, можно выделить следующие виды онтологий: онтология предметной области, онтология обрабатываемых задач, прикладная онтология, мета-онтологии, сетевая онтология [2].

В общем виде любая онтология представляет собой набор элементов, состоящих из четырех категорий: классы, слоты, отношения, экземпляры (сущности).

В данной работе рассматривается узкая предметная область. За основу берутся тексты биографического содержания. Суть метода, как и многих других, заключается в выделении «сущностей» предложений анализируемого текста. Технология базируется на многоуровневом представлении языка, т.к. этот подход наиболее удобен для объяснения внутреннего устройства систем обработки. Однако, ключевую роль в методе занимает семантический анализ. Именно на результатах его работы будет построена онтологическая модель биографических текстов.

2. Семантический анализ текста на основе лингвистического парсинга

Лингвистический парсер — комплекс программных модулей, обеспечивающий анализ и синтез текстов на естественном языке. Задачей лингвистического парсера является разбор и «понимание» поступившей на вход фразы на естественном языке или построение фразы на естественном языке, соответствующей формальному описанию ее смысла. В данной работе лингвистический парсер применяется непосредственно для анализа текста [3].

Лингвистический парсер представляет собой многоуровневый преобразователь. Его работа основывается на трех уровнях представления текста.

Морфологическая структура — это последовательность входящих в предложение слов с указанием их части речи и морфологических характеристик (падеж, число, род, время, вид и т. д.)

Синтаксическая структура — это дерево зависимостей, которое состоит из узлов (слова данного языка с указанием частей речи и грамматических характеристик) и дуг, соответствующих определенным для данного языка отношениям синтаксического подчинения.

Семантическая структура — это дерево зависимостей, которое состоит из узлов (предметные имена, слова универсального семантического языка) и дуг, соответствующих универсальным отношениям семантического подчинения (аргументное, атрибутивное, больше, меньше, принадлежит, не принадлежит).

3. Применение результатов парсинга в задаче автоматического формирования онтологии предметной области текста

Суть метода выделения «сущностей» на основе работы семантического компонента лингвистического парсера для автоматического построения онтологии текстов биографического плана приведена в примере ниже.

На вход семантического анализа поступает предложение: *Сергей родился 11 сентября* (рис. 1).

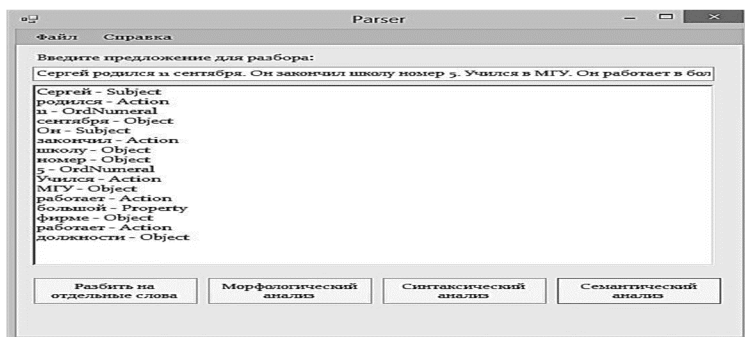


Рис. 1. Результат работы семантического анализа ЛП

«Сущностями», необходимыми для автоматического построения онтологии предметной области текста будут слова, роли которых относятся к Object (объекту), Subject (субъекту), OrdNumeral (порядковому числительному). Что касается роли Action (действие), в нашем методе она не значится.

4. Реализация системы автоматического формирования онтологии по биографическим текстам

Пользователь вводит текст, анализ которого его интересует. Далее по нажатию соответствующей кнопки пользователь проводит 3 вида анализа: морфологический, синтаксический или семантический.

В процессе морфологического анализа приложение обращается к морфологическому словарю системы Диалинг. Данный словарь хранит слова в виде псевдооснов и псевдофлексий.

При проведении синтаксического анализа приложение обращается к базе знаний на языке Prolog (среда разработки — SWI—Prolog 5.8.3). Для этого используется интерфейс взаимодействия SWI—Prolog и С.

Выполняя семантический анализ, приложение обращается к семантическому словарю, в котором словам русского языка сопоставлены некоторые категории. В результате семантического анализа на отдельной форме выводится семантическая сеть. В главном окне программы появляются интересующие нас «сущности» каждого предложения вводимого текста.

Следующим этапом работы программы является объединение данных сущностей в онтологию предметной области текста при помощи заранее определенной связи «is-a».

Результатом работы данного приложения является онтология предметной области анализируемого биографического текста (рис. 2).

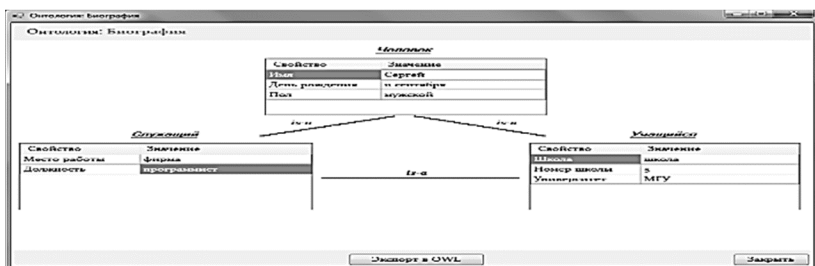


Рис. 2. Результат работы метода автоматического построения онтологии по тексту на естественном языке

Заключение

В результате проведенной работы достигнута главная цель, поставленная в начале. Создано приложение, которое автоматически формирует онтологию предметной области анализируемого текста. Также решены следующие задачи:

1. Разработана усовершенствованная модель лингвистического парсера.
2. Выполнена программная реализация приложения на C#.
3. Разработаны критерии оценки полученного приложения и проведен ряд тестирований.

На данный момент приложение способно работать с текстами биографического характера, состоящими из простых приложений. В скором времени планируется расширить его функциональность, добавив возможность разбора сложных предложений, а также возможность автоматического формирования онтологий других предметных областей. Не считая этого ограничения, пользователю предоставлены все заявленные возможности, включая и возможность редактирования семантического словаря.

Список литературы

1. Барсегян А.А. Технологии анализа данных. / А.А. Барсегян. – С.-Петербург : БХВ-Петербург, 2007. – 384с.
2. Шалфеева Е.А. Возможности использования онтологий при разработке программных систем / Е.А. Шалфеева. – Владивосток : ИАПУ ДВО РАН, 2011. – 21с.
3. Волкова И.А. Лингвистический процессор естественного языка / И.А. Волкова — М. : Издательский отдел факультета ВМиК МГУ, 2002. – 39 с.

РАЗРАБОТКА ИНФОРМАЦИОННОЙ СИСТЕМЫ ДЛЯ КООРДИНАЦИИ УЧЕБНЫХ И ИССЛЕДОВАТЕЛЬСКИХ ГРУПП

А.А. Коротких
Студент

С.В. Сапегин
Доцент

Введение

Проблема продуктивной координации исследовательской деятельности волновала исследователей с давних пор. В настоящий момент существует огромное количество методик разработки проектов. Все они по- своему и хороши и плохи.

Что же мы понимаем под «координацией исследовательской деятельности»? Это процесс, в котором происходит разделение группы людей на управляющих исследователями и, соответственно, исследователей. Первым необходимо выделять задачи и подзадачи проекта, назначать ответственных людей за её выполнение, следить за процессом реализации задачи. А вторым – знать полученную задачу, выполнять её, вести отчетность по процессу выполнения проблемы [1]. Поэтому возникает необходимость удобного взаимодействия внутри исследовательской группы.

1. Анализ задачи

На данный момент существует множество методик для организации управления деятельностью проектных групп:

1. Project Management Body of Knowledge (PMBoK) – это свод знаний по управлению проектами, который представляет собой сумму профессиональных знаний по управлению проектами. PMBoK является Американским национальным стандартом. В настоящем стандарте описываются суть процессов управления проектами в терминах интеграции между процессами и взаимодействий между ними, а также цели, которым они служат.

2. Agile-манифест разработки программного обеспечения – это серия подходов к разработке программного обеспечения, ориентированных на использование итеративной разработки, динамическое формирование требований и обеспечение их реализации в результате постоянного взаимодействия внутри самоорганизующихся рабочих групп, состоящих из специалистов различного профиля.

Основополагающие принципы Agile-манифеста [2]:

- изменение требований приветствуется, даже на поздних стадиях разработки. Agile-процессы позволяют использовать изменения для обеспечения заказчику конкурентного преимущества;
- на протяжении всего проекта разработчики и представители бизнеса должны ежедневно работать вместе;
- непосредственное общение является наиболее практичным и эффективным способом обмена информацией, как с самой командой, так и внутри команды;
- команда должна систематически анализировать возможные способы улучшения эффективности и соответственно корректировать стиль своей работы.

Из Agile-манифеста вытекает такая методология управления проектами как Scrum. Она активно применяется при разработке информационных систем для гибкой разработки программного обеспечения. Scrum чётко делает акцент на качественном контроле процесса разработки. Scrum – это набор принципов, на которых строится процесс разработки, позволяющий в жёстко фиксированные и небольшие по времени итерации, называемые спринтами (sprints), предоставлять конечному пользователю работающее ПО с новыми возможностями, для которых определён наибольший приоритет. Возможности ПО к реализации в очередном спринте определяются в начале спринта на этапе планирования и не могут изменяться на всём его протяжении. При этом строго фиксированная небольшая длительность спринта придаёт процессу разработки предсказуемость и гибкость.

Как мы видим, в этих методиках разработки проектов есть общие черты, такие как:

- планирование проектов;
- придание участникам определенных ролей;
- гибкость развития разработки проекта и др.

В этом проекте объединены общие принципы разработки и направлены на организацию управления проектными группами в учебной и исследовательской деятельности.

2. Постановка задачи

Цель работы: организовать деятельность проектных групп по научным и учебным проектам.

Систем для управления проектами существует много, но для учебных и исследовательских проектов нет. Этот проект направлен именно на организацию управления деятельностью проектных групп в области учебных и научных исследований.

3. Реализация проекта

Исходя из требований к системе, наиболее подходящим из языков программирования оказался Django framework(версия 1.6) [3], для базы данных – MySQL, а для оформления – Bootstrap 2.3.2 [4].

Основополагающим принципом является тот факт, что все пользователи разбиваются на различные группы, в которых уже создаются научные или исследовательские проекты. При этом один и тот же пользователь может состоять в нескольких группах и участвовать в нескольких проектах, выполняя определенные задачи, назначенные владельцем проекта.

На рисунок изображен пример распределения пользователей системы в зависимости от их ролей.

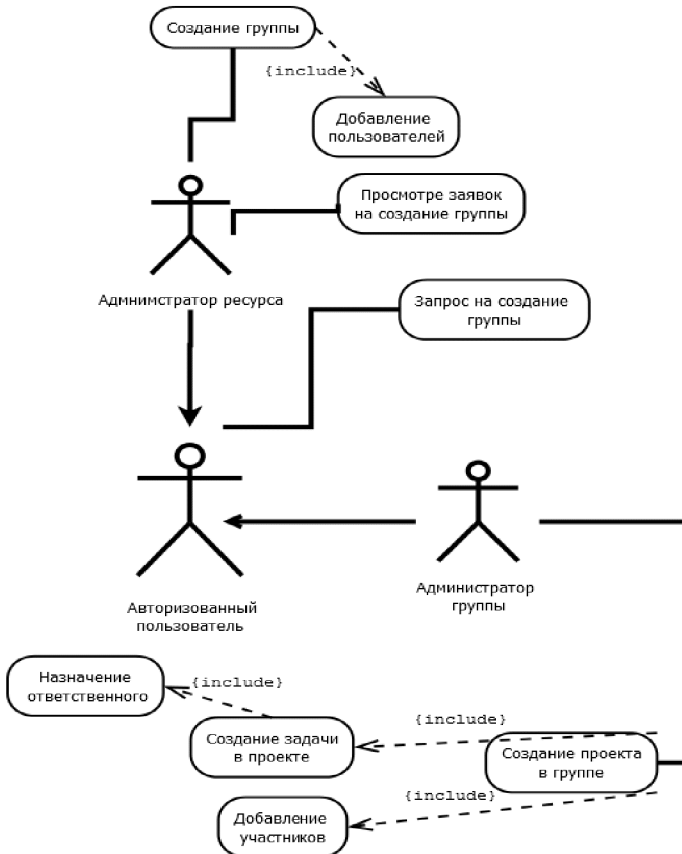


Рисунок. Use-Case диаграмма

Заключение

На данном этапе разработки реализованы следующие части:

– *Создание научных или исследовательских групп.* Пользователь отправляет запрос администратору ресурса, и администратор ресурса создает группу, в согласии с требованиями пользователя.

– *Создание проектов внутри группы.* Администратор группы, создает проект, в который добавляет пользователей из группы, при этом нет возможности добавить пользователя не из группы.

– *Создание задачи и назначение ответственного пользователя за ее выполнение.* Владелец проекта может создавать различные задачи, которые в совокупности приведут к конечной цели проекта. Так же он назначает пользователя, который должен будет решить задачу.

– *Добавление файлов и сообщений.* Любой участник проекта может комментировать и прикреплять файлы в проекте или задаче. Данные два действия реализованы при помощи металинков, что добавляет гибкости проекту.

– *События, которые произошли с пользователем.*

– *Раздел «Вопрос-ответ».* Любой авторизованный пользователь может задать вопрос администратору ресурса в этом разделе. Администратор ответит на вопрос и ответ будет виден любому посетителю ресурса, в том числе не зарегистрированному.

Список литературы

1. Кон М. Scrum: гибкая разработка ПО / М. Кон. – М. : «Вильямс», 2011. – С. 576.

2. Agile-манифест разработки программного обеспечения. – Режим доступа: <http://agilemanifesto.org/iso/ru/manifesto.html>

3. Django documentation. Models. – Режим доступа: <https://docs.djangoproject.com/en/1.6/topics/db/models>

4. Bootstrap. Components. – Режим доступа: <http://getbootstrap.com/2.3.2/components.html>

АФФИНО-ОДНОРОДНЫЕ ВЕЩЕСТВЕННЫЕ ГИПЕРПОВЕРХНОСТИ (0,0)-ТИПА В \mathbb{C}^3

С.Г. Косачев
Студент

А.В. Руденко,
Студент

А.В. Атанов
Ассистент

Введение

Задача описания поверхностей, однородных относительно действий различных групп преобразований (в частности, аффинных), известна довольно давно (см. [1]). Соответствующая классификация поверхностей в двумерном вещественном случае известна, относительно недавно получены списки всех аффинно-однородных гиперповерхностей в пространствах \mathbb{R}^3 и \mathbb{C}^2 . Более сложная задача таких гиперповерхностей в трёхмерном комплексном пространстве остаётся открытой, к настоящему времени получены лишь частичные результаты (см. [2], [3]). Данная работа посвящена изучению одного небольшого фрагмента этой задачи.

Введём определения.

Определение 1. Поверхность $M \subset \mathbb{C}^3$ называется однородной относительно группы преобразований G , если G транзитивно действует на M , то есть любую пару точек из M можно перевести друг в друга с помощью подходящего преобразования из группы G .

В ряде работ (см., например, [2]) установлено, что задачу описания однородных гиперповерхностей удобно решать, используя их канонические уравнения. Рассмотрим вещественно-аналитическую гиперповерхность $M \subset \mathbb{C}^3$, заданную вблизи начала координат уравнением

$$\begin{aligned} \Phi(z, \bar{z}, w) = & -v + |z_1|^2 + |z_2|^2 + \left(\varepsilon_1(z_1^2 + z_2^2) + \varepsilon_2(\bar{z}_1^2 + \bar{z}_2^2) \right) + \\ & + \sum_{k+l+2m \geq 3} F_{klm}(z, \bar{z}, u). \end{aligned} \quad (1)$$

Здесь z_1, z_2, w – комплексные координаты в \mathbb{C}^3 , $u = \operatorname{Re}(w)$, $v = \operatorname{Im}(w)$, F_{klm} – полиномы степени k по переменным $z = (z_1, z_2)$, степени l по переменным $\bar{z} = (\bar{z}_1, \bar{z}_2)$ и степени m по переменной u . Пара неотрицательных вещественных чисел $\varepsilon_1, \varepsilon_2$ образует аффинный инвариант поверхности (1). В зависимости от значений $\varepsilon_1, \varepsilon_2$ можно рассмотреть

несколько разных типов поверхностей. В нашей работе обсуждаются поверхности $(0,0)$ -типа, то есть в данном случае $\varepsilon_1 = \varepsilon_2 = 0$.

Мы обсуждаем непрерывные поверхности, поэтому и группы преобразований должны действовать здесь непрерывно. В такой ситуации естественным образом возникают группы Ли (см. [4]).

Определение 2. Под аффинной однородностью понимается существование в группе аффинных преобразований трёхмерного комплексного пространства некоторой подгруппы Ли $G(M)$, транзитивно действующей на поверхности M .

Мы изучаем однородность в локальном смысле, то есть свойство транзитивности должно выполняться лишь в малой окрестности фиксированной точки. В такой ситуации естественным является переход от самих преобразований к их инфинитезимальным (бесконечно малым) аналогам. Строго говоря, такое действие означает переход от группы преобразований к соответствующей алгебре векторных полей, касательных к рассматриваемой поверхности. Эти поля имеют следующий вид:

$$Z = (A_1 z_1 + A_2 z_2 + A_3 w + p) \frac{\partial}{\partial z_1} + (B_1 z_1 + B_2 z_2 + B_3 w + s) \frac{\partial}{\partial z_2} + (a z_1 + b z_2 + c w + q) \frac{\partial}{\partial w}, \quad (2)$$

где A_k, B_k ($k = 1, 2, 3$), a, b, c, p, s, q – комплексные константы.

Удобно также пользоваться матричным представлением каждого такого поля:

$$Z = \begin{pmatrix} A_1 & A_2 & A_3 & p \\ B_1 & B_2 & B_3 & s \\ a & b & c & q \\ 0 & 0 & 0 & 0 \end{pmatrix}. \quad (3)$$

Тогда линейное пространство \mathfrak{g} матриц, соответствующих линейным полям на поверхности M , образует матричную алгебру Ли, если оно замкнуто относительно операции коммутатора (скобки)

$$[Z_1, Z_2] = Z_1 \cdot Z_2 - Z_2 \cdot Z_1.$$

Здесь $Z_j \cdot Z_k$ – обычное матричное умножение, а замкнутость \mathfrak{g} относительно коммутатора означает, что для любых двух матриц из \mathfrak{g} их коммутатор также принадлежит \mathfrak{g} .

Факт касания линейным векторным полем рассматриваемой однородной поверхности M записывается в виде основного тождества

$$Re\{Z(\Phi)\} \equiv 0. \quad (4)$$

Для изучения однородности гиперповерхностей в ряде работ (см., например, [2], [3]) использовался «коэффициентный» подход, заключающийся в получении информации о связях элементов матриц (3)

друг с другом путём разложения аналитической функции, стоящей в левой части тождества (4), в ряд Тейлора в окрестности начала координат с последующим анализом младших компонент этого ряда. Применение данного подхода к рассматриваемому случаю поверхностей $(0,0)$ -типа позволило получить ряд важных результатов.

Основные результаты

Теорема 1. Для поверхностей $(0,0)$ -типа в уравнении (1) полином $F_3 \equiv 0$, а полином F_4 имеет вид

$$F_4 = \lambda(|z_1|^2 + |z_2|^2)^2 + \lambda u^2.$$

При этом алгебра векторных полей на однородной гиперповерхности M имеет вид

$$Z = \begin{pmatrix} m + im(A_1) & A_2 & 2i\lambda p & p \\ -\bar{A}_2 & m + im(B_2) & 2i\lambda s & s \\ 2i\bar{p} & 2i\bar{s} & 2(m + i\lambda q) & q \\ 0 & 0 & 0 & 0 \end{pmatrix}.$$

Здесь m, λ – вещественные, а все остальные коэффициенты комплексные.

Кроме того, размерность группы Ли аффинных преобразований, транзитивно действующей на аффинно-однородной вещественной гиперповерхности, допускает оценку

$$5 \leq \dim_R(g(M)) \leq 10.$$

Теорема 2. Если коэффициент λ в условиях теоремы 1 равен нулю, то размерность группы Ли аффинных преобразований, транзитивно действующей на аффинно-однородной вещественной гиперповерхности, равна 10, все слагаемые F_k ($k > 2$) уравнения (1) тождественно равны нулю, а интегральной поверхностью M является квадра

$$v = |z_1|^2 + |z_2|^2.$$

Список литературы

1. Cartan E. Sur la geometrie pseudoconforme des hypersurfaces de deux variables complexes/ E. Cartan // Ann. Math. Pura Appl. – 1932. – Vol. (4) 11. – P. 17–90.
2. Лобода А.В. Об одном семействе аффинно-однородных вещественных гиперповерхностей 3-мерного комплексного пространства / А.В. Лобода, А.С. Ходарев // Известия вузов. Математика. – 2003. – № 10. – С. 38–50.
3. Лобода А.В. Об аффинной однородности поверхностей трубчатого типа \mathbb{C}^3 / А.В. Лобода, Т.Т.З. Нгуен // Труды МИАН. – 2012. – Т. 279. – С. 93–110.
4. Постников М.М. Лекции по геометрии. Семестр 5. Группы и алгебры Ли / М.М. Постников – М. : Наука, 1982. – 447 с.

BUSINESS INTELLIGENCE И OLAP СЕГОДНЯ

Е.Д. Кочурова

Студент

И.В. Абрамов

Доцент

Введение

Business Intelligence (BI) – это совокупность технологий, программного обеспечения, практик, системного мышления и знаний, направленных на достижение целей бизнеса путём наилучшего использования имеющихся данных. В отличие от других классов делового ПО, актуальность во внедрении BI-систем в трудные с экономической точки зрения времена не ослабевает, а только усиливается. Многообразие представленных на рынке решений, от мощных платформ бизнес-анализа до простых систем аналитики и отчетности, позволяет выбрать решение, доступное любой организации. Развитие средств визуального представления данных, мобильных и облачных технологий сделали BI-инструменты массовыми всего за последние несколько лет. Сегодня теоретики, вендоры и исследователи рынка BI зачастую расходятся в определении того, что такое BI (системы бизнес-анализа). Приведём одно из многочисленных определений BI: «Business Intelligence — знания, добытые о бизнесе с использованием различных аппаратно-программных технологий. Такие технологии дают возможность организациям превращать данные в информацию, а затем информацию в знания».

Это определение четко разграничивает понятия «данные», «информация» и «знания». Данные понимаются как реальность, которую компьютер записывает, хранит и обрабатывает — это «сырые данные». Информация — это то, что человек в состоянии понять о реальности, а знания — это то, что в бизнесе используется для принятия решений. (Информация – осознание, знания – ОСМЫСЛЕНИЕ). В процессе организации информации для получения знания часто применяют хранилища данных, а для представления этого знания пользователям — инструменты бизнес-интеллекта [1]. В русскоязычной терминологии подобные системы называются также системами поддержки принятия решений – СППР (рис. 1).

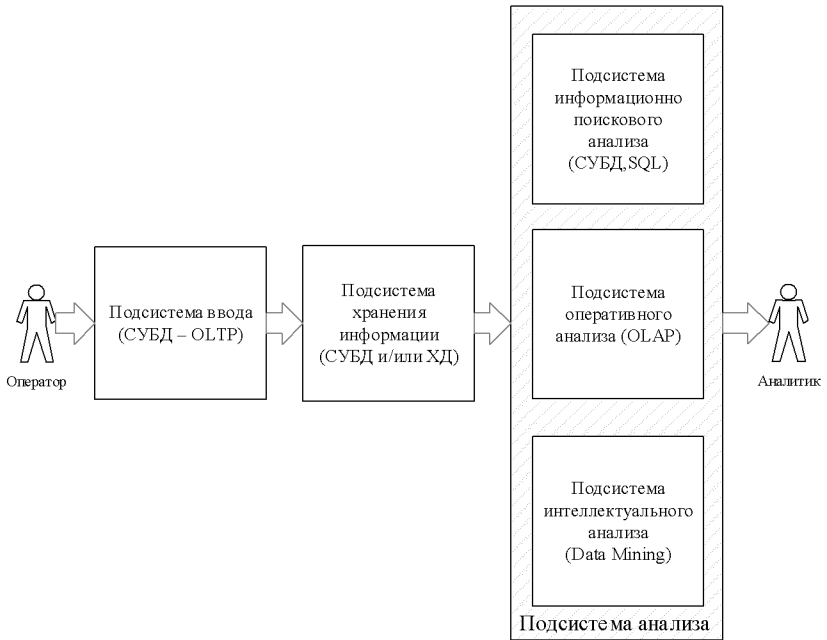


Рис. 1. Архитектура СППР

Каждый год количество данных в мире удваивается, но от этого мало пользы, хотя их можно превратить в полезную информацию и знания, но информация сама по себе не очень подходит для принятия решений в виду ее огромного объема. Средства BI и хранилищ данных призваны находить в кучах данных и информации то существенное, что реально прибавляется к нашим полезным знаниям.

1. Классификация продуктов business intelligence

Сегодня категории BI-продуктов включают: BI-инструменты и BI-приложения. Главная часть BI-инструментов делится на корпоративные BI-наборы и BI-платформы (рис. 2) [2]. Средства генерации запросов и отчетов в большой степени поглощаются и замещаются корпоративными BI-наборами. Многомерные OLAP-механизмы или серверы, а также реляционные OLAP-механизмы являются BI-инструментами и инфраструктурой для BI-платформ.

Большинство BI-инструментов применяются конечными пользователями для доступа, анализа и генерации отчетов по данным, которые чаще всего располагаются в хранилище, витринах данных или оперативных складах данных. Разработчики приложений используют BI-

платформы для создания и внедрения BI-приложений, которые не рассматриваются как BI-инструменты.

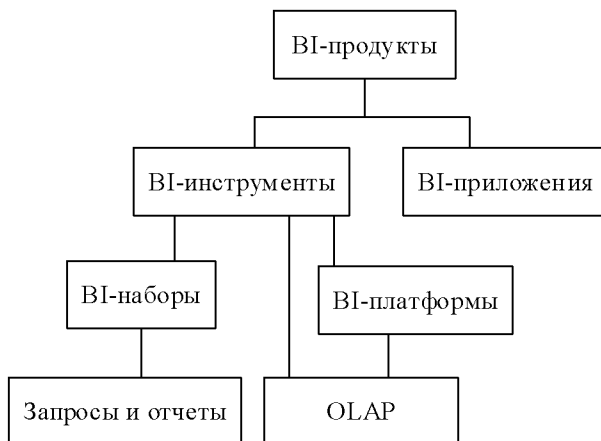


Рис. 2. Категории BI-продуктов

2. OLAP или развитие аналитические инструменты

Рассмотрим более подробно OLAP-системы.

OLAP (от англ. OnLine Analytical Processing — оперативная аналитическая обработка данных) — подход к аналитической обработке данных, базирующийся на их многомерном иерархическом представлении, являющийся частью более широкой области информационных технологий — бизнес-аналитики (BI). Это технология, использующая методы и средства для сбора, хранения и анализа многомерных данных в целях поддержки процессов принятия решений.

OLAP это также технология обработки информации, включающая составление и динамическую публикацию отчётов и документов. Успешно используется аналитиками для быстрой обработки сложных запросов к базе данных. Служит также для подготовки бизнес-отчётов по продажам, маркетингу, в целях управления, т.н. Data Mining — добыча данных (способ анализа информации в базе данных с целью отыскания аномалий и трендов без выяснения смыслового значения записей).

Инструменты OLAP являются аналитическими инструментами, которые первоначально были основаны на многомерных базах данных (МБД). МБД — это базы данных, сконструированные специально для поддержки анализа количественных данных с множеством измерений, содержат данные в «чисто» многомерной форме. Большинство приложений включают измерение времени, другие измерения могут касаться географии, организационных единиц, клиентов, продуктов и др.

OLAP позволяет организовать измерения в виде иерархии. Данные представлены в виде гиперкубов (кубов) — логических и физических моделей показателей, коллективно использующих измерения, а также иерархии в этих измерениях. Некоторые данные предварительно агрегированы в БД, другие рассчитываются «на лету».

Средства OLAP позволяют исследовать данные по различным измерениям. Пользователи могут выбрать, какие показатели анализировать, какие измерения и как отображать в кросс-таблице, обменять строки и столбцы, затем сделать срезы и вырезки, чтобы сконцентрироваться на определенной комбинации размерностей. Можно изменять детальность данных, двигаясь по уровням с помощью детализации и укрупнения, а также кросс-детализации через другие измерения.

Для поддержки МБД используются OLAP-серверы, оптимизированные для многомерного анализа и поставляемые с аналитическими возможностями. Они обеспечивают хорошую производительность, но обычно требуют много времени для загрузки и расширения МБД. Классический OLAP-сервер — Hyperion Essbase Server.

Сегодня реляционные СУБД применяются для эмуляции МБД и поддерживают многомерный анализ. OLAP для реляционных БД (ROLAP) имеет преимущество по масштабируемости и гибкости, но проигрывает по производительности многомерному OLAP (MOLAP), хотя существуют методы повышения производительности, наподобие схемы «звезда». Несмотря на то, что МБД являются по-прежнему наиболее подходящими для оперативной аналитической обработки, сейчас эту возможность встраивают в реляционные СУБД или расширяют их (например, MS Analysis Services или Oracle OLAP Services — это не то же самое, что ROLAP).

Также существует гибридная оперативная аналитическая обработка данных (HOLAP) для гибридных продуктов, которые могут хранить многомерные данные естественным образом и в реляционном представлении. Доступ к МБД осуществляется с помощью API для генерации многомерных запросов, тогда как к реляционным БД доступ производится посредством запросов на SQL. Примером ROLAP-сервера является Microstrategy 7i Server.

Настольные OLAP-инструменты (например, BusinessObjects Explorer, Cognos PowerPlay, MS Data Analyzer) облегчают конечным пользователям просмотр и манипулирование многомерными данными, которые могут поступать из серверных ресурсов данных ROLAP или MOLAP. Некоторые из этих продуктов имеют возможность загружать кубы, так что они могут работать автономно. Настольные инструменты по

сравнению с MOLAP-средствами имеют небольшую производительность и аналитическую мощь. Нередко обеспечивается интерфейс через Excel, например, MS Excel2000/OLAP PTS, BusinessQuery for Excel. Практически все OLAP-инструменты имеют Web-расширения (Business Objects WebIntelligence к примеру), для некоторых они являются базовыми.

3. Преимущества использования BI-системы

Ключевыми преимуществами от внедрения BI системы на данный момент являются повышение скорости и точности составления отчетов, аналитики и планирования, повышение эффективности процессов принятия решений, увеличение удовлетворенности клиентов. Гораздо реже BI-системы (системы бизнес-анализа) позволяют добиться сокращения операционных расходов или увеличения объемом выручки (рис. 3).

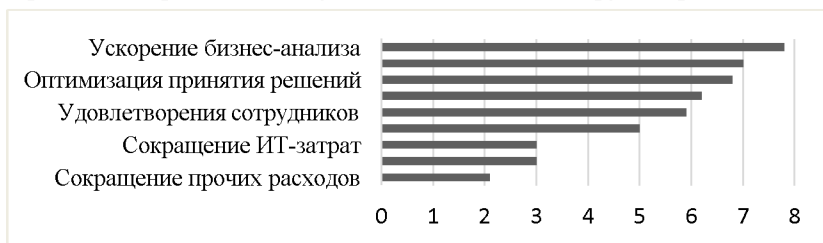


Рис. 3. Наиболее часто достигаемые в ходе внедрения BI-системы преимущества

4. Задачи, решаемые BI-системой

Системы для бизнес-анализа решают очень широкий спектр задач. Так, «ближним горизонтом» является мониторинг, анализ и корректировка оперативных целей:

- поддержка развития бизнес-процессов и структурных изменений предприятия;
- возможность моделирования различных бизнес-ситуаций в единой информационной среде;
- проведение оперативного анализа по нестандартным запросам;
- снижение рутинной нагрузки на персонал и высвобождение времени для более глубокой аналитической работы;
- устойчивая работа при увеличении объема обрабатываемой информации, возможность масштабирования.

В части поддержки стратегического развития предприятия BI-системы обеспечивают:

- оценку эффективности различных направлений бизнеса;
- оценку достижимости поставленных целей;

- оценку эффективности использования ресурсов, в том числе дочерними предприятиями;
- оценку эффективности операционной, инвестиционной и финансовой деятельности;
- бизнес-моделирование и оценку инвестиционных проектов;
- управление затратами, налоговое планирование, планирование капитальных вложений.

5. Оценки рынка BI в России

По данным TAdviser, Рынок BI-систем России и стран СНГ (лицензии и услуги) в 2012 году преодолел отметку в 43,125 млрд рублей, что на 25% больше, чем годом ранее (34,5 млрд рублей). Темп роста рынка несколько замедлился против 35% в 2011 году, но все еще остался довольно значительным, опережающим темпы роста рынка ИКТ в целом и других сегментов рынка делового ПО.

В 2012-2013 году в лидеры BI-внедрений опять выдвинулись компании финансового сектора. Это связано с большим числом проектов по внедрению новых аналитических технологий и хранилищ данных, которые были реализованы за последние 18 месяцев. В России же традиционно бизнес с высокой маржинальностью, такие как финансовые услуги, телеком и ритейл, первыми осваивают передовые технологии информатизации.

OLAP-системы входят в состав подавляющего большинства решений для бизнес-аналитики, «корпоративных» редакций СУБД основных поставщиков (IBM, Microsoft, Oracle). В той или иной мере технологии OLAP используются в существенной части современных ERP-систем. В государственном секторе РФ отдается предпочтение OLAP-инструментария, предложенному Группой компаний БАРС Груп [1].

Заключение

Сегодня во всём мире, в трудные с экономической точки зрения времена, актуальность BI-систем усиливается, число внедрения этих систем растёт, области применения BI-систем расширяются.

В конце 2012-2013 года начала активно формироваться «третья волна» внедрений BI, связанная с освоением российскими компаниями новых подходов к хранению и анализу данных. Так например в России началось активное освоение Big Data и вычислений в оперативной памяти.

С технологической точки зрения российский рынок BI не испытывает ни малейшего отставания от мирового в плане доступности передовых технологий, но освоение их только начинается.

Список литературы

1. Системы бизнес-аналитики 2013. TAdviser портал выбора технологий и поставщиков. – Режим доступа: <http://www.tadviser.ru/index.php/BI>

2. Барсемян А.А. Методы и модели анализа данных: OLAP и DataMining / А.А. Барсемян, М.С.Куприянов. – Санкт-Петербург : БХВ-Петербург, 2006. – 336 с.

ИССЛЕДОВАНИЕ МАРШРУТИЗАЦИИ ПРИ РАЗЛИЧНОЙ ПЛОТНОСТИ РАСПОЛОЖЕНИЯ УЗЛОВ

Е.О. Лукин
Студент

Ю.Б. Нечаев
Профессор

Введение

На сегодняшний день сотовая телефония продемонстрировала огромную востребованность рынка мобильных абонентов к передаче голосовых и информационных данных со скоростями от нескольких сотен килобит до нескольких мегабит в секунду. Создаваемые информационные системы призваны стать (в большей или меньшей степени) частью информационной сети, обеспечивающей абонентов глобальным роумингом. Решение этой задачи связывают с внедрением новых (3G, WiMAX) и совершенствованием уже существующих (Wi-Fi) технологий беспроводной передачи данных. Одним из вариантов решения подобных сетей, основанных на кластерной структуре, является технология Mesh.

1. Теоретический анализ

Информационные сети, организованные по топологии Mesh, получили за последние полтора-два года большое признание. Масштабы проектов выросли до тысяч точек доступа и десятков тысяч пользователей. Mesh-сети представляют наиболее интересные решения, интегрирующие различные сетевые и радиотехнологии, и потому в полной мере отвечают все более растущим требованиям абонентов (мобильность, QoS, безопасность).

Возможность организации с помощью Mesh-топологии локальных (LAN) и городских (MAN) сетей, легко интегрируемых в глобальные сети (WAN), является привлекательным фактором для муниципальных и персональных пользователей.

Существующие в настоящее время Mesh-сети построены с использованием наиболее распространенного беспроводного стандарта Wi-Fi. Преимущества такого решения очевидны – широкий спектр дешевых стандартных абонентских устройств определяет коммерческую успешность проектов.

Программа OpNet представляет собой комплекс средств, для создания моделирования и изучения сетей связи. Позволяет анализировать воздействия приложений типа клиент-сервер и новых технологий на работу сети; моделировать иерархические сети, многопротокольные локальные и глобальные сети с учетом алгоритмов маршрутизации; осуществлять оценку и анализ производительности смоделированных сетей. Также с помощью пакета можно осуществить проверку протокола связи, анализ взаимодействий протокола, оптимизацию и планирование сети [1].

На рис. 1 представлена главная страница программного продукта OpNet.

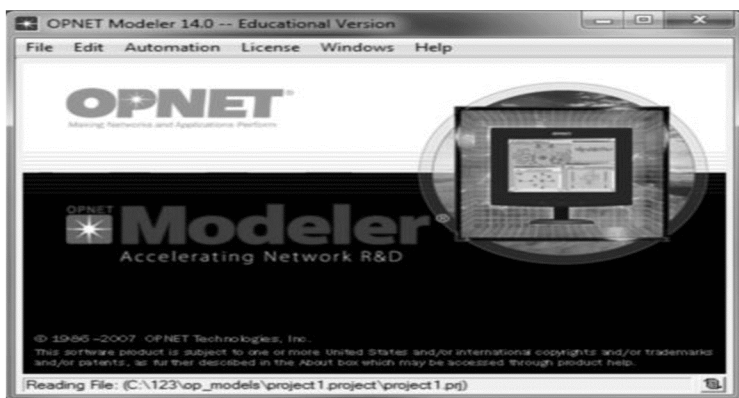


Рис. 1. Программа OpNet Modeler 14.0

Mesh – это сетевая топология, в которой устройства объединяются многочисленными (часто избыточными) соединениями, вводимыми по стратегическим соображениям. Определяет принцип построения сети, отличительной особенностью которой является самоорганизующаяся архитектура, реализующая следующие возможности:

- создание зон сплошного информационного покрытия большой площади;
- масштабируемость сети (увеличение площади зоны покрытия и плотности информационного обеспечения) в режиме самоорганизации;

- использование беспроводных транспортных каналов (backhaul) для связи точек доступа в режиме "каждый с каждым"
- устойчивость сети к потере отдельных элементов.

На рис. 2 представлен пример Mesh – сети. Протокол маршрутизации — сетевой протокол, используемый маршрутизаторами для определения возможных маршрутов следования данных в составной компьютерной сети. Применение протокола маршрутизации позволяет избежать ручного ввода всех допустимых маршрутов, что, в свою очередь, снижает количество ошибок, обеспечивает согласованность действий всех маршрутизаторов в сети и облегчает труд администраторов.

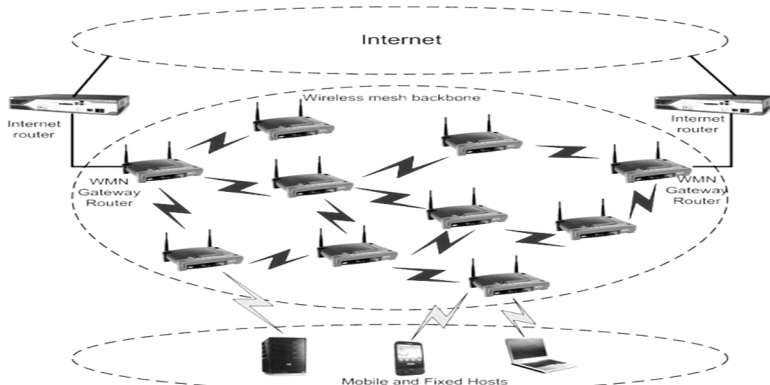


Рис. 2. Пример Mesh – сети

Протоколы маршрутизации делятся на два вида, зависящие от типов алгоритмов, на которых они основаны:

- Дистанционно-векторные протоколы, основаны на Distance Vector Algorithm (DVA);
- Протоколы состояния каналов связи, основаны на Link State Algorithm (LSA).

Так же протоколы маршрутизации делятся на два вида в зависимости от сферы применения:

- Междоменной маршрутизации;
- Внутridoменной маршрутизации.

2. Экспериментальная часть

Цель исследования заключается в выборе протокола маршрутизации при различной плотности расположении узлов в программном продукте OpNet на смоделированной Mesh – сети.

Для выбора оптимальной маршрутизации при различной плотности расположении узлов, я рассмотрел четыре протокола:

1. **AODV** (англ. Ad hoc On-Demand Distance Vector) — протокол динамической маршрутизации для мобильных ad-hoc сетей (MANET) и других беспроводных сетей. Является реактивным протоколом маршрутизации, то есть устанавливает маршрут до адресата по требованию. В отличие от классических протоколов маршрутизации Интернета являющихся превентивными, то есть находящих пути маршрутизации независимо от использования маршрутов. Как следует из названия, для вычисления маршрутов используется дистанционно-векторный алгоритм маршрутизации. В AODV с помощью применения порядковых номеров при обновлениях маршрута исключена возможность возникновения проблемы «счета до бесконечности», присущая другим протоколам, использующим этот алгоритм маршрутизации.

2. **DSR** (англ. Dynamic Source Routing) — протокол маршрутизации для MANET с топологией mesh. Схож с AODV в том, что также формирует маршрут по требованию. Однако, он использует явную маршрутизацию, не полагаясь на таблицы маршрутизации на каждом промежуточном устройстве. Кроме того, в DSR было внесено множество последовательных конкретизаций, включая DSR-Flow (гибрид явной маршрутизации и маршрутизации по таблицам).

3. **OLSR** (англ. Optimized Link-State Routing) — протокол маршрутизации для MANET, который также может использоваться в других беспроводных сетях. OLSR — проактивный протокол маршрутизации, использующий обмен сообщениями приветствия и контроля для получения информации о топологии сети. Выполняет следующие задачи: обнаружение соседей, распространение по сети маршрутной информации о звеньях между узлами для построения графа сети, поиск кратчайшего маршрута на построенном графе.

4. **TORA** (англ. Temporally Ordered Routing Algorithm) адаптивный алгоритм маршрутизации, основными свойствами которого являются:

- распределенное исполнение
- исключение закливания
- многопутевая маршрутизация
- реактивное или активное создание и поддержание маршрута
- быстрая реакция на топологические изменения

Распределенность TORA заключается в том, что маршрутизаторы хранят информацию только о прилегающих узлах, в то же время, как и в дистанционно-векторных протоколах (основанные на Distance Vector Algorithm (DVA)) сохраняются маршруты для каждого направления. Однако в данном протоколе кратчайшие пути обновляются не постоянно, а для оценки оптимальности маршрута используются весовые метрики узлов, созданные при установлении маршрута [2].

В таблице представлены основные параметры смоделированной сети.

Таблица

Основные параметры сети

Количество узлов	20
Физический уровень	WI-FI
Протокол доступа к среде	CSMA/CA

Рассмотрим два случая построения сети:

I. *Топология сети при низкой плотности распределения узлов 900×900 (рис. 3).*

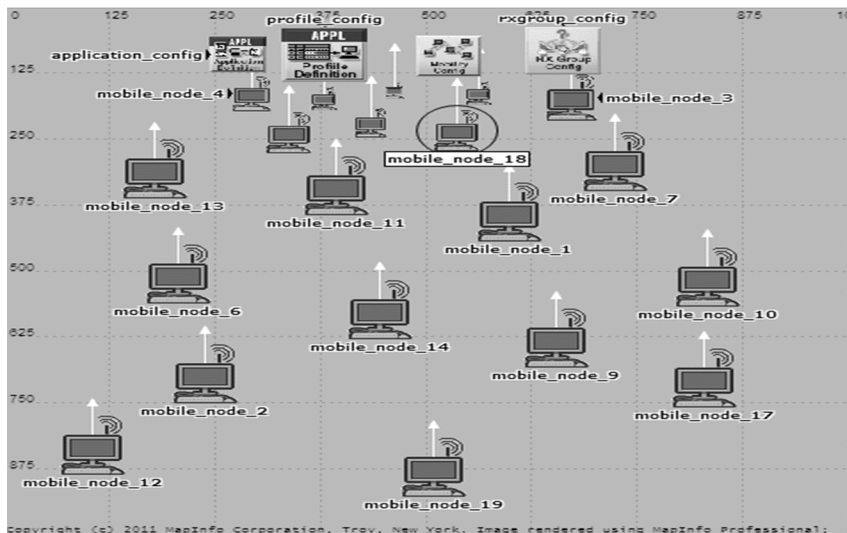


Рис. 3. Низкая плотность распределения узлов в сети

Всем узлам присваиваем по очереди один протокол маршрутизации и выполняем симуляцию. На рис. 4 представлен выбор протокола маршрутизации в узле.

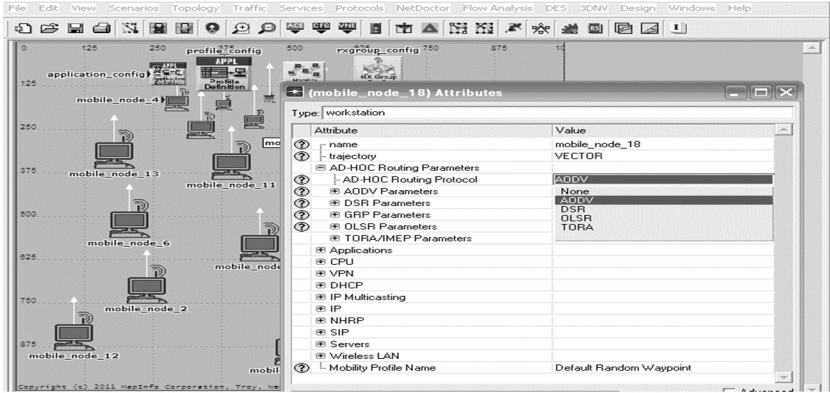


Рис. 4. Выбор протокола маршрутизации

После чего, по графикам общего объема служебной информации протоколов маршрутизации, общего объема потерянной пользовательской информации в канале, среднее время построения маршрутов, среднее количество ретрансляторов на маршрутах, общий объем потерянной пользовательской информации в буферах узлов, сравнивая протоколы по показателям, выбираем наиболее подходящий к данному виду сети. На рис. 5 представлен график общего объема служебной информации протоколов маршрутизации.

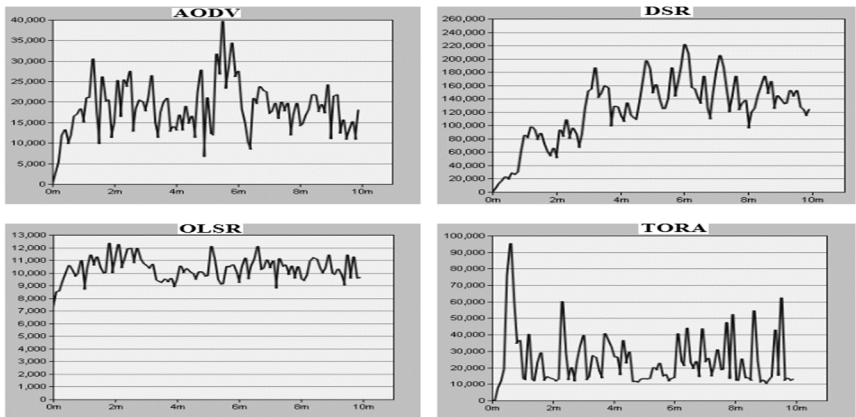


Рис. 5. Общий объем служебной информации протоколов маршрутизации при низкой плотности распределении узлов

На рис. 6 представлен график общего объема потерянной пользовательской информации в канале.

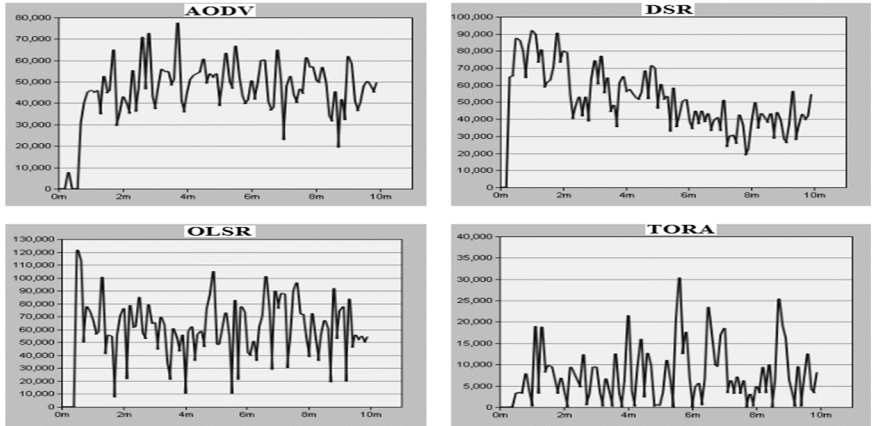


Рис. 6. Общий объем потерянной пользовательской информации в канале при низкой плотности распределении узлов

При низкой плотности расположения узлов в сети протокол OLSR создает наименьшую нагрузку на сеть за счет собственной служебной информации протокола маршрутизации, однако использование данного протокола приводит к существенным потерям пользовательской информации при передаче трафика чувствительного к задержкам, что обусловлено использованием проактивной стратегии построения маршрутов. Минимизировать объем потерь пользовательской информации позволяет построение сети на основе реактивного протокола AODV.

II. Топология сети при высокой плотности распределении узлов 125×125 (рис. 7).

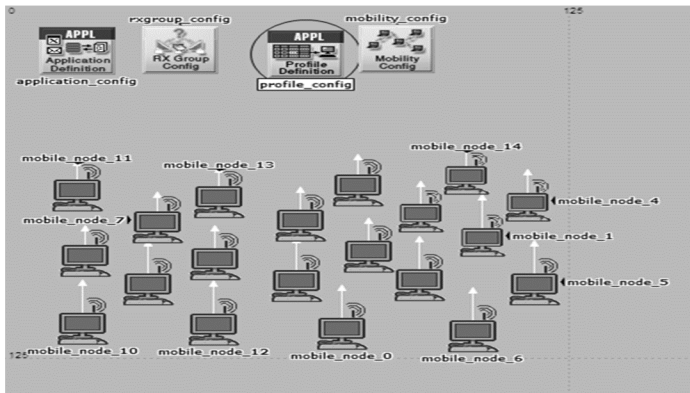


Рис. 7. Высокая плотность распределения узлов в сети

Аналогичным образом, меняем протоколы маршрутизации в каждом узле и проводим четыре симуляции. На рис. 8 представлен график общего объема служебной информации протоколов маршрутизации.

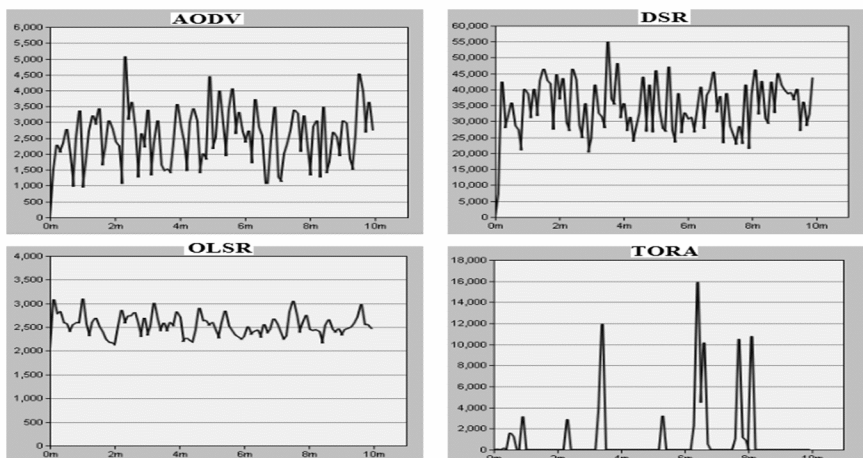


Рис. 8. Общий объем служебной информации протоколов маршрутизации при высокой плотности распределении узлов

На рис. 9 представлен график общего объема потерянной пользовательской информации в канале.

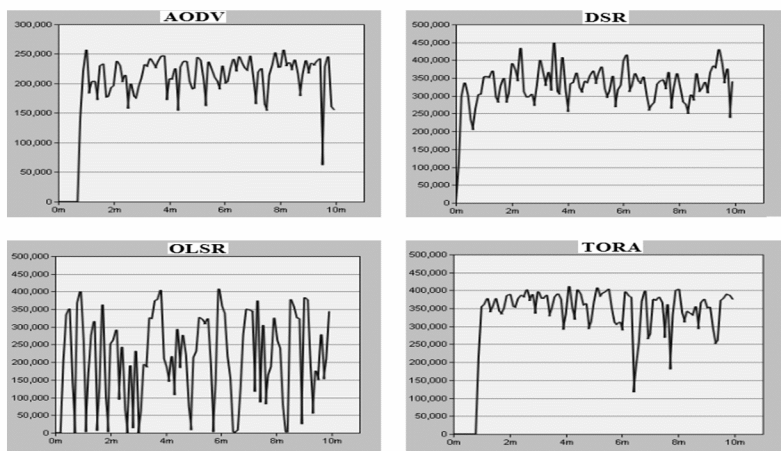


Рис. 9. Общий объем потерянной пользовательской информации в канале при низкой плотности распределении узлов

При высокой плотности расположения узлов в сети протокол OLSR создает наименьшую нагрузку на сеть за счет собственной служебной информации, однако использование данного протокола приводит к существенным потерям пользовательской информации при передаче трафика чувствительного к задержкам, что обусловлено использованием проактивной стратегии построения маршрутов. Минимизировать объем потерь пользовательской информации позволяет построение сети на основе гибридного протокола TORA.

Заключение

Сравнение протоколов маршрутизации при высокой плотности расположения узлов в сети показало, что при передаче информации чувствительной к задержкам в сети с мобильными абонентами целесообразно использовать протокол TORA. В сети с низкой плотностью расположения узлов высокую эффективность демонстрируют реактивные протоколы маршрутизации, оптимальным является использование протокола AODV.

Список литературы

1. Маколкина М.А. Моделирование сетей связи в программном продукте OpNet. / М.А. Маколкина. – СПб : СПбГУТ, 2009. – 31 с.
2. Свободная энциклопедия. – Режим доступа: <http://ru.wikipedia.org>

РАЗРАБОТКА СИСТЕМЫ ЛОКАЛИЗАЦИИ WI-FI УСТРОЙСТВ В ПОМЕЩЕНИИ

А.Д. Мигаль
Студент
И.В. Китаев
Ассистент

Введение

Благодаря развитию технологий стало уделяться большое внимание системам слежения за перемещениями людей внутри зданий. Варианты использования этих систем могут быть совершенно различны. Например, это задачи навигации, отслеживание трафика людей, контроль безопасности объектов или учет статистики посещаемости торговых центров.

Определение местоположения в закрытых помещениях является обособленной задачей, где невозможно применение GPS. Это требует иных подходов: использование видеокамер, ИК-датчиков, радиооборудования и т.п. Подавляющее большинство таких систем являются сложными и дорогими в реализации.

Новые возможности предлагают беспроводные радиосети семейства Wi-Fi. Они широко распространены, просты в установке и эксплуатации, дешевы и надежны. Мобильные устройства с возможностью доступа к таким сетям повсеместно распространены. Все это позволяет использовать эти две составляющие как единую систему. Обычные беспроводные точки доступа можно рассматривать как сенсоры, а мобильные устройства посетителей как радиометки, позволяющие, не нарушая приватности, идентифицировать отдельного человека.

Всё это делает технологию Wi-Fi многообещающим инструментом для определения местоположения внутри помещений, обладающим значительным рядом достоинств по сравнению с аналогичными системами.

1. Принцип работы системы локализации

Основная идея работоспособности системы заключается в одной из особенностей работы Wi-Fi сетей [1]. В соответствии со стандартом мобильные устройства периодически рассылают широковещательные пакеты с опросом окружающих сетей (probe-request). Эти пакеты можно зарегистрировать на обычной беспроводной точке доступа и получить MAC-адрес устройства, по которому в дальнейшем будет происходить идентификация, и мощность принятого сигнала. Путем оценки мощности сигнала, используя большую сеть связанных между собой точек доступа, становится возможным вычислить примерное расположение устройства.

2. Основные сущности и создание событий

Для удобства работы с исходными данными и перехода непосредственно к процедуре локализации, было разработано несколько специализированных сущностей (рис. 1).

- Record («запись»). Представляет единичный пакет, отправленный устройством. Каждая такая сущность содержит значение мощности принятого сигнала и время фиксации.

- Short Event («короткое событие»). Событие отправки устройством одного широковещательного запроса. По стандарту для обеспечения надежности устройство рассылает несколько идентичных пакетов в рамках одного запроса. «Короткое событие» объединяет несколько «записей» в рамках одной секунды между ними.

- Common Short Event («общее короткое событие»). Событие регистрации широковещательного запроса всеми беспроводными точками доступа, находящимися в радиусе действия, состоит из множества «коротких событий». Используя эти сущности, возможна локализация устройства путём использования информации о мощности сигнала, содержащейся в «записях».

- Long Event («длинное событие»). Событие нахождения мобильного устройства в области действия точек доступа. Состоит из множества «общих коротких событий», объединенных по правилу: интервал времени между двумя последовательно идущими широковещательными запросами не должен превышать среднестатистического значения для данного производителя устройства. Для «длинного события» осуществляется расчет продолжительности нахождения внутри помещения.

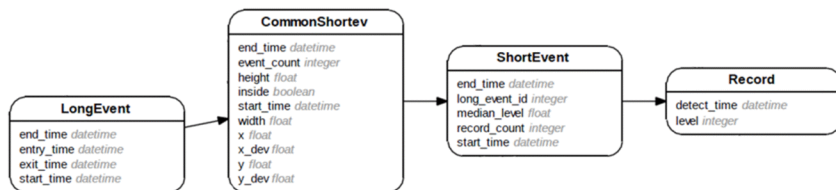


Рис. 1. Абстрактные понятия, используемые при локализации

Расчет и создание перечисленных событий происходит постоянно на сервере по мере приема данных с беспроводных точек доступа. Хранение сущностей происходит в соответствующих таблицах БД. Рассмотрение полной диаграммы таблиц БД не представляется целесообразным в рамках данной работы.

3. Этапы работы системы

1. Регистрация точками доступа широковещательных пакетов, отправленных мобильными устройствами, и загрузка данных на сервер.
2. Сохранение в БД информации о MAC-адресах устройств и мощности принятого сигнала.
3. Расчет специальных сущностей-событий («короткое событие», «короткое общее событие»)
4. Определение местоположения устройства для каждого «короткого общего события».
5. Вычисление «длинных событий», которые представляют собой информацию об отслеживании конкретного посетителя (например, время нахождения в помещении), которую можно использовать для дальнейших прикладных задач.

4. Методики локализации

Среди всех подходов к определению местоположения на основе использования радиосигналов, можно выделить две большие группы [2]:

1. Методы, основанные на мощности принятого сигнала (RSSI-подход);
2. Методы, использующие заранее составленную карту распространения сигнала («Signal fingerprinting»).

Рассмотрим первую группу. Уровень мощности принятого сигнала можно рассматривать как зависимость от длины пути, который пришлось преодолеть сигналу. Таким образом, можно определить по мощности расстояние от конкретной точки доступа, которая зафиксировала событие, и мобильным устройством. Для решения этой задачи существует довольно большое количество различных теоретических моделей распространения радиосигнала в пространстве. Для определения конечного положения может использоваться триангуляция, либо какие-то другие усовершенствованные подходы.

5. Модели распространения сигнала

Рассмотрим несколько наиболее простых и распространенных подходов к определению расстояния, пройденного сигналом [2-4].

- Модель Фрииса (для свободного пространства)

$$P_r = P_t \left(\frac{\lambda}{4\pi d} \right)^2 G_r G_t,$$

$$P_r = P_t + G_t + G_r + 20 \log_{10} \left(\frac{\lambda}{4\pi d} \right),$$

где P_t – мощность переданного сигнала, P_r – мощность принятого сигнала, G_r, G_t – коэффициенты усиления приемника и передатчика, d – расстояние между приемником и передатчиком.

- Статистическая модель «One-Slope»

$$L(d) = L_{fs} + 10n \log \left(\frac{d}{d_0} \right) + X_\sigma,$$

где $d_0 = 1\text{м}$, L_{fs} – потери в свободном пространстве на расстоянии d_0 , n – коэффициент, учитывающие характеристики помещения, X_σ – нормальная случайная величина.

- Двухлучевая модель. Является улучшенной версией простой модели распространения в открытом пространстве [3]. Учитывается не только прямое расстояние между приёмником и передатчиком, но и отражение от поверхности пола.

$$d < d_{th}: P_{r,TR} = \frac{G_r G_t \lambda^2 P_t}{(4\pi d)^2} = P_{r,FS},$$

$$d \geq d_{th}: P_{r,TR} = \frac{G_t G_r h_t^2 h_r^2 P_t}{d^4},$$

где d_{Th} – граничное расстояние, в зависимости от которого применяется другое соотношение.

После проведения экспериментов было установлено, что достаточно высокую точность обеспечивает простейшая логарифмическая модель, получаемая из модели Фрииса (рис. 2). Такая модель является наиболее простой в реализации и не требует дополнительной настройки параметров и знаний о помещении.

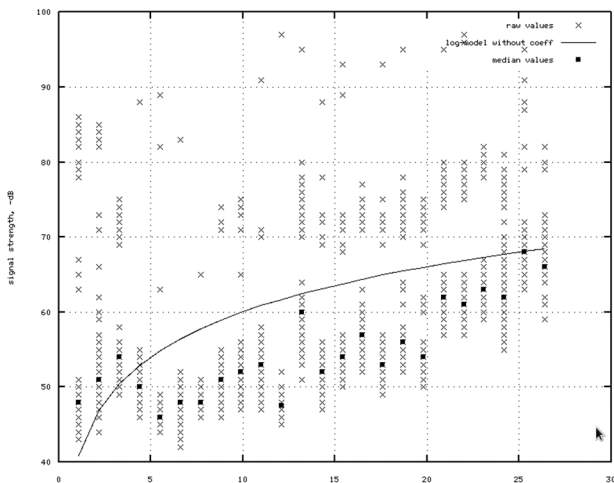
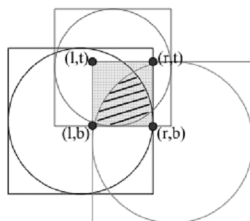


Рис. 2. Сравнение теоретической модели с реальными данными

При определении местоположения устройства удобно использовать один из простейших методов «min-max» [5], который, как было показано, несмотря на простоту реализации, демонстрирует высокую точность.

Метод представляет обычную триангуляцию с небольшим изменением. Область нахождения устройства расширяется до прямоугольного вида, что лучше соответствует зашумлению сигнала в реальных ситуациях (рис. 3)



$$\begin{aligned} \text{left} &= \max_{i=1}^{N_{anc}} \{a_i\} \\ \text{right} &= \min_{i=1}^{N_{anc}} \{a_i\} \\ \text{top} &= \min_{i=1}^{N_{anc}} \{a_y\} \\ \text{bottom} &= \max_{i=1}^{N_{anc}} \{a_y\} \end{aligned}$$

Рис. 3. Метод определения местоположения устройства

6. Метод «Signal fingerprinting»

Другим подходом к локализации является метод построения карты распространения сигнала [6,7]. Для этого требуется два этапа, первый является предварительным.

1. Этап построения радиокарты. В различных областях помещения, обычно с шагом около 1 метра, проводят измерения мощности принятого сигнала от мобильного устройства. Полученные значения сохраняются, образуя «отпечаток» (fingerprint, вектор значений мощностей).

2. Этап определения местоположения. Алгоритм, получая на вход мощности сигнала, определяет местоположение устройства (источника сигнала) в соответствии с картой, построенной на первом этапе. Метод должен быть устойчивым к шумам и искажениям значений, которые могут присутствовать в карте. Например, это может быть алгоритм ближайшего соседа, который демонстрирует достаточную точность при локализации.

К достоинствам этого подхода относятся простота реализации и возможность адаптации к различным помещениям. Несмотря на необходимость трудоёмкого этапа построения карты, этот метод на практике оказывается лучше использования теоретических моделей, так как позволяет неявным образом учесть все особенности конкретного помещения.

7. Результаты

Система локализации была реализована путем использования RSSI подхода. Тестирование проводилось в лабораториях кафедры ФКН, а также в одном из магазинов города. Было получено, что в условиях открытого пространства и небольших помещений возможно достичь достаточно хорошей точности. В среднем ошибка локализации составила около 3 метров, и зачастую снижалась до 1 метра. Также всегда можно было установить, находилось ли устройство в помещении или снаружи него (рис. 4).

Но данный подход сильно деградирует в условиях наличия различного рода помех. Стены, предметы интерьера, отражающие поверхности значительно искажают и зашумляют радиосигнал [8]. Зачастую становится невозможным установить факт нахождения в помещении. Из-за этого RSSI методы плохо применимы в условиях сложных помещений, так как в математической модели на практике не представляется возможным учесть все факторы, влияющие на распространение сигнала, либо же этап создания этих моделей является излишне трудоёмким. Поэтому наиболее перспективным методом для реализации является «signal fingerprinting», который универсален и

применим к различным помещениям, требуя минимальных затрат на построение карты сигнала.



Рис. 4. Пример результата успешной локализации

Система была реализована с использованием базы данных PostgreSQL и фреймворка Ruby on Rails для удобной работы с сущностями и проведению необходимых расчетов.

Заключение

В результате выполнения данной работы была реализована система пространственной локализации мобильных WiFi-устройств в помещениях. В результате проведенных экспериментов была подтверждена эффективность её применения. Полученная система использует обычное сетевое оборудование, не требует дорогостоящей инфраструктуры и проста во внедрении. Подобные реализации могут с успехом использоваться для задач учета количества посетителей различных торговых центров и магазинов для оценки успешности ведения бизнеса, а также для выявления причин, влияющих на продажи.

Список литературы

1. Gast M. 802.11ac: A Survival Guide / M. Gast. – Sebastopol : O'Reilly Media, 2013. – 152 p.
2. Поникар А.В. Исследование возможности локального позиционирования в беспроводных сетях IEEE 802.15.4 / А.В. Поникар, О.В. Евсеев, В.Е. Анциперов // IV Всероссийская конференция «Радиолокация и радиосвязь». – ИРЭ РАН. – 2010. – С. 914–918.
3. Ezzine R. A New Generic Model for Signal Propagation in Wi-Fi and WiMAX Environments / R. Ezzine, Ala AL-Fuqaha, R. Braham, A. Belghith and etc. – Dubai : Wireless Days, 2008.

4. Tummala D. Indoor propagation modeling at 2.4 GHz for IEEE 802.11 networks / D. Tummala. – University of North Texas, 2005. – 119 p.
5. Will H. The Membership Degree Min-Max Localization Algorithm/ H. Will, T. Hillebrandt, Yang Yuan // AG Computer Systems & telematics, Berlin, 2010.
6. Rai A. Zee: Zero-effort Crowdsourcing for indoor localization / A. Rai, K. Kant, V. Padmanabhan // Microsoft Research, India, 2012.
7. Farshad A. A Microscopic look at Wi-Fi Fingerprinting for indoor mobile phone localization in diverse environments/ A. Farshad, L. Jiwei, K. Mahesh // International Conference on Indoor Positioning and Indoor Navigation, 2013.
8. Никольский В.В. Электродинамика и распространение радиоволн / В.В. Никольский, Т.И. Никольская. – М. : Наука, 1989. – 537 с.

МОДИФИКАЦИЯ ДЕТЕКТОРА ОБНАРУЖЕНИЯ КРАЕВ CANNY С ИСПОЛЬЗОВАНИЕМ СТАТИСТИКИ И ПРИЗНАК ХААРА

Х. Мирзай
Студент

Н.А. Тюкачев
Доцент

Введение

В статье рассматриваются алгоритмы обнаружения краев на основе градиентных методов. Кратко изложен принцип работы детектора обнаружения краев Canny [1], а также предложена модификация этого детектора с использованием статистики и признаков Хаара и интегральных изображений предложенным Виолом и Джонсом [2].

Как правило, края в изображении распознаются как сильный разрыв в сигнале (интенсивности или цвета). Кажется, что легко измерить такие разрывы, но такое определение неясно, поскольку не известно что значит "сильный разрыв".

Многие алгоритмы обнаружения края оставляют эту проблему пользователям в виде порогового значения, которое должно быть указано вручную. Таким образом, пользователь может настроить порог согласно его потребностям, но этот подход не удовлетворяет требованию к обнаружению краев, что должно надежно работать при различных

обстоятельствах, поскольку порог, который хорошо работает в одной ситуации может не работать в другой.

Независимо от того, что квалифицирует "сильный разрыв", с семантической точки зрения ясно, что не все края в изображении образуют контуры объекта. Контуры объектов несет достаточно информации для обнаружения некоторого объекта в изображении. Обнаружение края осуществляется на более фундаментальном уровне, чем обнаружение контура. В задаче обнаружения краев сталкиваемся с проблемой идентификации различных видов краев, таких как ступенчатые края, линии, переходы и т.д. Важно отметить, что обнаружение края не является необходимым условием для обнаружения контура. Обнаружение контура само по себе не занимается вопросами, какие края присутствуют или как они ориентированы друг к другу. Оно направлено на определение границ между объектами, которые также могут быть получены с помощью других средств, производящих идеально закрытые контуры.

1. Градиентные методы обнаружения краев

Использование разрывов яркости при обнаружении контуров объектов имеет много преимуществ. Одним из них является тот факт, что, как правило, это очень просто и поэтому быстро реализуется. Помимо этого, метод обычно может быть легко адаптирован к цветным изображениям: каждый канал может рассматриваться отдельно и можно получить отклики от фильтров для всех каналов, а затем объединить в одно изображение. Учет цвета означает, что нет никакой потери информации: когда цветные изображения преобразуются в черно-белые, много цветов проецируются на те же значения серого. Это означает, что границы объектов, которые явно были обнаруживаемыми в исходном изображении могут не отображаться больше в преобразованном изображении.

Класс фильтров, которые могут быть использованы для обнаружения краев, являются направленными фильтрами производной первого порядка, такие как оператор Собеля (Sobel) [3], Prewitt, Roberts и т.д. . Ниже показан фильтры S_v , S_h ядра Собеля для обнаружения вертикальных и горизонтальных краев соответственно.

$$S_v = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}, \quad S_h = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

Если существует сильный разрыв яркости в том же направлении, что и указывает фильтр, результат фильтра будет иметь высокое абсолютное значение. Полезным свойством оператора Собеля является его

сепарабельность. Это позволяет заменить 2D-свертку двумя 1D-сверток, которые более эффективнее. Еще одной особенностью фильтра Собеля является то, что можно использовать его в качестве фильтра высоких частот: в связи с антисимметричной конструкцией фильтра, медленные изменения сигнала сбалансированы, и поэтому будут подавляться. Поскольку края имеют резкие отклонения, то они представлены в виде высокочастотной составляющей в сигнале. К сожалению, детали изображения и шумы также являются высокочастотными компонентами. Это означает, что шумы на изображении могут также проходить через фильтр Sobel.

2. Модификация детектора обнаружения Canny

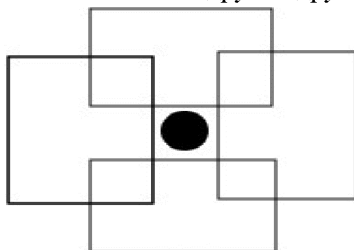
Канни (John F. Canny; 1953 г.) изучил математическую проблему получения фильтра, оптимального по критериям выделения, локализации и минимизации нескольких откликов одного края. Это означает, что детектор должен реагировать на границы, но при этом игнорировать ложные, точно определять линию границы (без её фрагментирования) и реагировать на каждую границу один раз, что позволяет избежать восприятия широких полос изменения яркости как совокупности границ. Канни ввел понятие Non-Maximum Suppression (подавление не-максимумных значений), которое означает, что пикселями границ объявляются точки, в которых достигается локальный максимум градиента в направлении вектора градиента. Несмотря на то, что его работа была проведена на заре компьютерного зрения (1986 г.), детектор границ Canny до сих пор является одним из лучших детекторов.

Алгоритм состоит из пяти отдельных шагов:

1. **Сглаживание.** Размытие изображения для удаления шума.
2. **Поиск градиентов.** Границы отмечаются там, где градиент изображения приобретает максимальное значение.
3. **Подавление не-максимумов.** Только локальные максимумы отмечаются как границы.
4. **Двойная пороговая фильтрация.** Потенциальные границы определяются порогами.
5. **Трассировка области неоднозначности.** Итоговые границы определяются путем подавления всех краев, несвязанных с определенными (сильными) границами.

Среди причисленных шагов, шаг 2 является значительным важным, поскольку именно на этом шаге обнаруживаются края и все качество работы зависит от успеха этого шага. Связи с этим в этой статье предложен алгоритм для улучшения качества детектора на шаге вычисления градиентов.

Как правило, для вычисления градиента используется фильтр или операция свертки. Главная идея модификация операции вычисления градиента заключается в том, что можно получить среднее значение (яркости, цвета) из некоторого прямоугольной областей (в 4 направлениях, право, лево, верх и вниз, рис. 1) из каждого канала изображения, затем составить некоторый вектор признаков и с помощью статистики получить некоторую вероятность, определяющую то, насколько две противоположные области отличаются друг от друга.



Кружок по середине — текущий пиксель, прямоугольные — области из которых вычисляются среднее значение

Рис. 1. Разбиения при операции вычисления градиента

Очевидно, что относительно размера окон время работы алгоритма меняется прямо пропорционально. Для решения этой проблемы и повышения эффективности работы алгоритма использован следующий инструмент так называемый прямоугольные признаки Хаара [2].

Признаки Хаара — признаки цифрового изображения, используемые в распознавании образов. Своим название они обязаны интуитивным сходством с вейвлетами Хаара. Признаки Хаара использовались в первом детекторе лиц, работающем в реальном времени.

Исторически сложилось так, что алгоритмы, работающие только с интенсивностью изображения (например, значение RGB в каждом пикселе), имеют большую вычислительную сложность. В работе Папагеоргиу [4], была рассмотрена работа с множеством признаков, основанных на вейвлетах Хаара. Виола и Джонс [2] адаптировали идею использования вейвлетов Хаара и разработали то, что было названо признаками Хаара. Признак Хаара состоит из смежных прямоугольных областей. Они позиционируются на изображении, далее суммируются интенсивности пикселей в областях, после чего вычисляется разность между суммами. Эта разность и будет значением определенного признака, определенного размера, определенным образом позиционированного на изображении.

Ключевой особенностью признаков Хаара является наибольшая, по сравнению с остальными признаками, скорость. При использовании

интегрального представления изображения, признаки Хаара могут вычисляться за постоянное время (примерно 60 процессорных инструкций на признак из двух областей).

Простейший прямоугольный признак Хаара можно определить как разность сумм пикселей двух смежных областей внутри прямоугольника, который может занимать различные положения и масштабы на изображении. Такой вид признаков называется 2-прямоугольным. Виола и Джонс так же определили 3-прямоугольные и 4-прямоугольные признаки. Каждый признак может показать наличие (или отсутствие) какой-либо конкретной характеристики изображения, такой как границы или изменение текстур. Например, 2-прямоугольный признак может показать, где находится граница между темным и светлым регионами.

Одним из вкладов Виола и Джонса было использование суммированных прямоугольных регион, которые они называли интегральные изображения. Интегральные изображения могут быть определены как двумерных справочных таблиц в виде матрицы с тем же размером исходного изображения. Каждый элемент целостного образа содержит сумму всех пикселей, расположенных на верх-влево области исходного изображения (по отношению к положению элемента). Это позволяет вычислить сумму прямоугольных областей на изображении, в любом положении или масштаба, используя только четыре поисков:

$$\text{sum} = I(C) + I(A) - I(B) - I(D),$$

где точки A, B, C, D принадлежат интегральному изображению I , как показано на рис. 2.

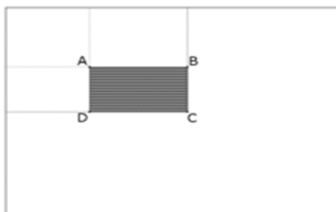


Рис. 2. Вычисление суммы штрихованной прямоугольной области

Как следует из названия, значение в любой точке (x, y) в суммарной табличной области это просто сумма всех вышеперечисленных точек и точек слева от (x, y) , включительно:

$$I(x, y) = \sum_{\substack{x' \leq x \\ y' \leq y}} i(x', y'),$$

Кроме того, суммированная табличная область может быть вычислена эффективно за один проход над изображением, используя тот

факт, что значение в суммарной табличной зоны на (x, y) вычисляется просто:

$$I(x, y) = i(x, y) + I(x-1, y) + I(x, y-1) - I(x-1, y-1).$$

Таким образом, можно считать сумму в области прямоугольника любого размера только четырьмя поисками по интегральному изображению. Следовательно, для использования этого инструмента, вычисляем интегральное изображения для каждого канала цвета (RGB).

Вводим еще один инструмент, который назовем вектором признаков с размером равным количеству каналов (3 канала, RGB). Этот вектор заполняется из суммы прямоугольных областей, которые получаем с помощью интегрального изображения.

$$V_i, V_r, V_u, V_d = \frac{1}{m \cdot n} V_i,$$

где m, n — размер прямоугольника, $V_i = \begin{pmatrix} 2, v \\ v, 3 \end{pmatrix}$ — вектор из трех компонентов.

Каждый вектор признак представляет собой некоторой случайной величины распределенному по некоторому закону. Для сравнения двух векторов используем критерий согласия Пирсона.

Критерий согласия Пирсона, или критерий согласия χ^2 (Хи-квадрат) — наиболее часто употребляемый критерий для проверки гипотез о принадлежности наблюдаемой выборки x_1, x_2, \dots, x_n объемом n некоторому теоретическому закону распределения $F(x, \theta)$.

Критерий может использоваться при проверке простых гипотез вида

$$H_0 : F_n(x) = F(x, \theta),$$

где θ — известный вектор параметров теоретического закона.

В соответствии с заданным разбиением подсчитывают число n_i выборочных значений, попавших в i -й интервал, и вероятности попадания в интервал

$$p_i(\theta) = F(x_i, \theta) - F(x_{i-1}, \theta)$$

соответствующие теоретическому закону с функцией распределения $F(x, \theta)$, при этом

$$n = \sum_{i=1}^k n_i \quad \text{и} \quad \sum_{i=1}^k P_i(\theta) = 1.$$

Критерий χ^2 вычисляется следующим образом:

$$\chi^2 = \sum_{i=1}^n \frac{(H_1 - H_2)^2}{H_1},$$

где χ^2 – тестовая статистика Пирсона, который асимптотически приближается χ^2 распределению, $\frac{2}{H}$ – гистограммы или в данном случае векторы признаков, n – длина гистограммы.

Хи-квадрат статистика может быть использована для вычисления Р-значения, путем сравнения величины статистики с хи-квадрат распределением. Для данной случаи степень свободы $k = 2$ (рис. 3). Здесь, Р-значение (от нуля до единицы) показывает на сколько гистограммы похоже распределены или похоже друг на друга. Мы исходим из того, что граница объектов находится между двумя разными областями, где цвета по разному распределены.

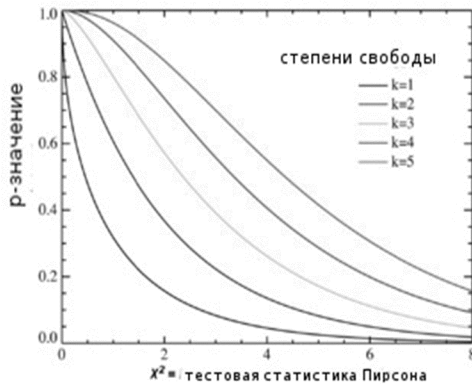


Рис. 3. Хи-квадрат распределение, показывает χ^2 на горизонтальной оси и Р-значение на вертикальной оси.

Таким образом, при вычислении градиента, можно использовать эту вероятность следующим образом:

$$G_x = G_{rl} \cdot (1 - p_{rl}),$$

$$G_y = G_{ud} \cdot (1 - p_{ud}),$$

где G_x – градиент по горизонтальному направлению, G_y – градиент по вертикальному направлению, G_{rl} – градиент по горизонтальному направлению полученный в результате применения фильтра, G_{ud} – градиент по вертикальному направлению полученный в результате

применения фильтра, $(1 - p_{rl})$ – вероятность того насколько сильно отличается области слева и справа от данного пикселя, $(1 - p_{ud})$ – вероятность того насколько сильно отличается области сверху и снизу от данного пикселя.

Рисунок 4 показывает результат применения статистики вместе с фильтром Собеля, а также результат без применения статистики.

Анализ тестового изображения показывает, с использованием статистики получаем лучший результат. В этом примере для вычисления градиента использовался фильтр Собеля с размером прямоугольных областей для статистики равным (8×8) .

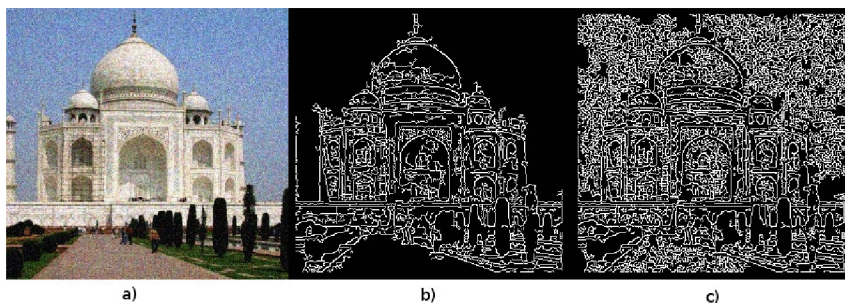


Рис. 4. (а) Исходное зашумленное изображение гауссовским шумом, (б) результат детектора края с применением статистики (критерий χ^2 Пирсона), (с) результат детектора без применения статистики.

Заметно, что улучшение получено в тех областях изображения, где область является однородной. Из-за чувствительности к шуму, что характерно для всех градиентных методов особенно на зашумленных изображениях, получаем много ложных краев, но с использованием статистики видно, что ложные края в этих регионах хорошо подавлены.

Так как мы используем интегральное изображение при вычислении суммы в прямоугольных областях, то можем задать практически любой размер окна, который не будет влиять на скорость работы. Это очень полезно в тех случаях, когда деталей в изображении мало или этими деталями можно пренебречь.

3. Результат тестирования модифицированного детектора Canny

Рассмотрим результаты тестов детектора Canny и его модификации с использованием статистики (критерий хи-квадрат Пирсона). В таблице приведено сравнение результата реализаций двух детектора.

В каждом примере время вычисления указано в миллисекундах. На вход детектора передается два порога, минимальное и максимальное(min,

max) значения для операции гистерезиса. Оригинальное тестовое изображение приведено на рис. 5.

Сравнение результатов реализаций детекторов Канни и его модификации

Таблица

Детектор Канни

Детектор Канни с использованием статистики



Время вычисления = 5391ms
min = 10, max = 40



Время вычисления = 17421ms
min = 10, max = 40



Время вычисления = 5359 ms
min = 50, max = 100



Время вычисления = 17345 ms
min = 50, max = 100



Рис. 5. Оригинальное тестовое изображение, размер = 286x176

Заключение

Градиентными методами являются самые популярные и первые методы обнаружения краев. Не смотря на многолетие работы в разработке методов обнаружения краев, до сих пор ни один из них не удовлетворяет всем требованиям для различных условий. Детектор краев Санну является лучшим между детекторами краев на основе градиентных методов по отношению скорости и качества.

Модификация детектора Санну, предложенной в данной статье, показывает некоторое улучшение в подавлении шумовых краев особенно в однородных областях изображения.

Из преимуществ модифицированного алгоритма можно сказать об эффективности вычисления вектора признака с помощью интегрального изображения. Тем не менее, при некотором улучшении по сравнению с оригинальным детектором Санну можно отметить увеличение общего времени вычисления детектора почти в три раза. Связано это с дополнительными вычислениями интегральных изображений для 3-х каналов (RGB) и значений вероятности.

Список литературы

1. Canny J. A computational approach to edge detection / J. Canny. – IEEE Trans. Pattern Anal. Mach. Intell. – 1986. – June. – P. 679–698.
2. Viola R.. Rapid object detection using a boosted cascade of simple features / R. Viola, K. Jones // Computer Vision and Pattern Recognition. – 2001. – P. 785–792.
3. Gonzalez R. Digital Image Processing / R. Gonzalez, R. Woods. – Boston: Addison Wesley, 1992. – 516 p.
4. Papageorgiou O. A general framework for object detection / O. Papageorgiou, S. Poggio. – International Conference on Computer Vision. – 1998. – P. 198–204.

АВТОМАТИЗАЦИЯ ПОИСКА ФЕНОМЕНОВ, СВЯЗАННЫХ С РАЗЛИЧНЫМИ СОСТОЯНИЯМИ ГОЛОВНОГО МОЗГА

П.М. Михин

Студент

Я.А. Туровский

Доцент

Введение

В современном мире цифровые технологии активно внедряются в повседневную жизнь. Компьютеризация не обошла стороной и систему здравоохранения. Результаты исследований, полученные в цифровой форме при помощи современных медицинских диагностических приборов, предоставляют богатые возможности для диагностики, т.к. цифровая информация удобна для обработки компьютером. Не менее важно, что крупные базы данных для проведения исследований в цифровой форме гораздо удобнее хранить, нежели существующие «бумажные» архивы.

Одной из перспектив развития медико-информационных технологий, решающих проблему совершенствования операторской деятельности формирования технологий общения людей с ограниченными возможностями, является разработка новых методов человеко-машинного общения: человеко-машинных, нейрокомпьютерных интерфейсов.

1. Анализ проблемы

В связи с тем, что современные медико-диагностические приборы обладают высокой точностью и эффективностью, они требуют для обработки результатов специфического и дорогого программного обеспечения [1]. В лаборатории информационных технологий в медицине кафедры цифровых технологий ФКН ВГУ можно решать уникальные задачи, что, приводит к необходимости разработки собственных программных решений.

Одним из методов реализации нейрокомпьютерного интерфейса является анализ электроэнцефалограммы (ЭЭГ).

ЭЭГ позволяет детально исследовать активность мозга человека при решении различных задач. Основная проблема при поиске различных феноменов с использованием ЭЭГ заключается в том, что изначальная группа исследуемых должна быть достаточно большой. Человеческий мозг обладает множеством индивидуальных особенностей, и для достижения максимально объективных результатов необходимо всех испытуемых разбить на подгруппы, по различным дополнительным признакам, таким как пол, возраст, темперамент и т.п. Причем чем больше признаков будет

использовано, тем объективнее результаты будут получены [1]. Но расширение признакового пространства неминуемо ведет к резкому росту необходимого числа вычислений.

Исходя из вышеописанных особенностей выбранного метода исследований, актуальной задачей становится разработка программного обеспечения для автоматизации анализа накопленных экспериментальных данных.

2. Реализация

Разработанное программное обеспечение позволяет проводить массовый статистический анализ и оценку экспериментальных данных, с использованием самых распространенных методик параметрического и не параметрического анализа. К таким следует отнести следующие критерии:

- критерий Уилкоксона (Wilcoxon signed-rank test);
- критерий Краскела-Уоллиса (Kruskal–Wallis one-way analysis);
- критерий Хи-квадрат Пирсона (Pearson's chi-squared test),

Кроме того, используются коэффициенты корреляции:

- коэффициент корреляции Спирмена (Spearman's rank correlation coefficient);
- коэффициент корреляции Пирсона (Pearson correlation coefficient).

Критерий Уилкоксона – относится к непараметрическим статистическим критериям, применяемым для проверки двух выборок парных измерений на различия. Этот критерий эффективно используется для сопоставления экспериментально полученных значений, которые характеризуют одну группу испытуемых, но в различных состояниях. Он позволяет констатировать не только выраженность, но и направленность изменений, определяет относительную интенсивность сдвигов показателей [2].

Критерий Краскела-Уоллиса относится к непараметрическим статистическим критериям, используемым для выявления различий двух выборок. Выборки берутся из закона распределения, отличного от нормального, либо получаются путем измерения с использованием порядковой шкалы [2, 3].

Критерий Хи-квадрат Пирсона применяется для проверки гипотезы, что наблюдаемая случайная величина подчиняется некому теоретическому закону распределения. В большинстве задач, возникающих на практике, точный закон распределения заранее неизвестен [3]. Применяя данный критерий, можно проверить гипотезу о соответствии эмпирического закона, построенного с учетом наблюдений, предполагаемому теоретическому закону. По результатам проверки гипотеза будет либо подтверждена, либо опровергнута.

Коэффициенты корреляции описывают статистическую связанность нескольких случайных, либо считаемых случайными с допустимой степенью точности, величин. При этом изменения одних величин приводят к прогнозируемому изменению других [2].

Помимо уже подготовленных статистических методов анализа, разрабатываемое программное обеспечение способно расширить набор выполняемых функций за счет использования дополнительных, подключаемых модулей.

Главной особенностью разработанного программного обеспечения является возможность описать весь процесс исследования (выбор данных, разбиение данных на отдельные блоки, выбор методики анализа для каждого блока) до начала вычислений. Это экономит время и позволяет за один запуск обработать большой объем данных. Блок-схема одного из возможных процессов изображена на рисунке.

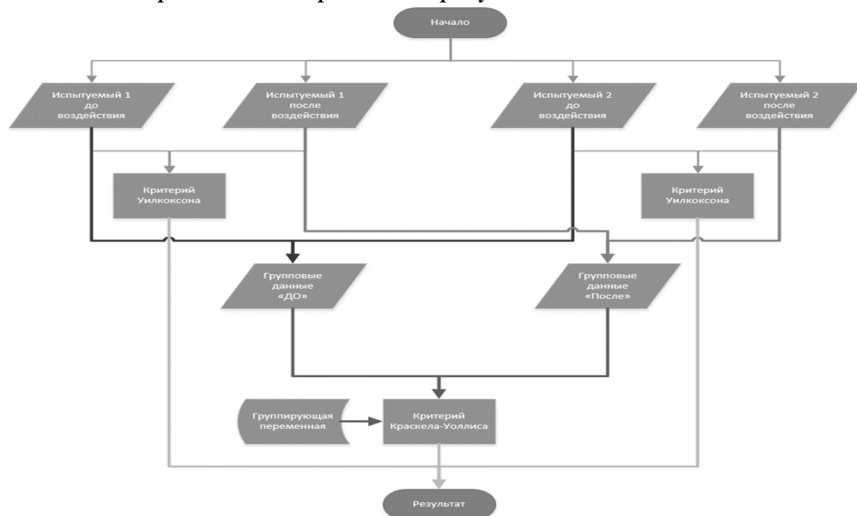


Рисунок. Блок-схема исследования

Первым шагом происходит вычисление критерия Уилкоксона для результатов одного испытуемого до и после воздействия. Далее происходит объединение результатов всех испытуемых до и после воздействия, добавляется группирующая переменная-признак, и вычисляется групповой критерий Краскела-Уоллиса. В качестве результата выводятся значения всех проводимых вычислений, с выделением достоверных результатов.

Заключение

Таким образом, автоматизация проведения исследований, направленных на поиски феноменов, связанных с различными состояниями головного мозга средствами разработанного программного обеспечения, позволяет не только использовать время оптимальным образом, но и провести целый комплекс исследований и получить достоверные данные за один запуск. А возможность подключать дополнительные модули дает возможность применять самые современные методики анализа без существенной переработки программных средств.

Список литературы

1. Гнездицкий В.В. Вызванные потенциалы мозга в клинической практике / В.В. Гнездицкий. – Таганрог : Издательство ТРТУ, 1997. – 252 с.
2. Кобзарь А.И. Прикладная математическая статистика / А.И. Кобзарь. – М. : Физматлит, 2006. – 458 с.
3. Рунион Р. Справочник по непараметрической статистике / Р. Рунион. – М. : Финансы и статистика, 1982. – 198 с.

МОДЕЛИРОВАНИЕ И РАСЧЕТ ОПТИМАЛЬНЫХ ПАРАМЕТРОВ ДЛЯ ЭКЗОСКЕЛЕТА ВЕРХНЕЙ КОНЕЧНОСТИ

В.Г. Михина
Студент

Я.А. Туровский
Доцент

Введение

Роль цифровых технологий в повседневной жизни переоценить трудно. Упрощение расчетов, совершенствование способов обработки информации, создание универсальных информационных систем – все это позволяет одновременно снизить расходы на эксперименты с реальными объектами и расширить объем получаемых данных путем работы с математическими и компьютерными моделями. Это особенно важно в сфере медицины, где моделирование является одним из самых незаменимых инструментов в любом исследовании, будь то разработка новой вакцины или изучение механизмов поведения человека. Компьютерные науки предоставляют широкий спектр возможностей для диагностики. Полученные в цифровой форме результаты экспериментов удобно хранить, компьютерная обработка позволяет быстрее

анализировать полученные данные, структурировать их, выявлять особенности и вычислять оптимальные значения необходимых параметров.

Одним из направлений медико-информационных технологий является разработка оборудования для помощи травмированным людям и людям с инвалидностью, пожилым людям, которые в силу своего возраста имеют проблемы с опорно-двигательным аппаратом. Возможным решением данной проблемы является экзоскелет.

Экзоскелет – это механизм, который за счет внешнего каркаса способен пропорционально увеличивать силу, скорость движений, повышать прочность человеческого тела. Данные разработки важны не только в медицине, но и в других областях – например, могут применяться в спасательных операциях, связанных с ликвидацией последствий завалов, крушений зданий, когда необходимо приложить все усилия для безопасности пострадавших. Свести риск к минимуму очень сложно, пользуясь лишь громоздкой техникой, которая не способна оперативно помочь и не всегда применима. Экзоскелеты для спасателей в таких ситуациях просто незаменимы [1].

С помощью экзоскелета люди станут выносливее, смогут преодолевать большие дистанции. Также станет возможным перемещение тяжелых грузов в труднодоступную местность. Кроме того, они смогут чинить тяжелое оборудование, которое иначе было бы невозможно привести в рабочее состояние.

1. Анализ проблемы

Исследования в данной области являются актуальными по нескольким причинам. Экзоскелет состоит из 5 элементов, без которых его работа немыслима: структура, энергия, управление, исполнительные механизмы и биомеханика. Но для достижения результата необходимо преодолеть несколько фундаментальных проблем при его разработке. Во-первых, важно выбрать правильный материал, который одновременно будет прочным и достаточно легким, а также сможет обеспечить защиту человека, например, от воздействия разрушающих факторов внешней среды. Во-вторых, источники питания для данных механизмов на сегодняшний день не обладают достаточной емкостью, чтобы обеспечить работу в течение 24 часов. В-третьих, экзоскелет руки должен повторять биомеханику человека и быть максимально приближенным по размерам к человеческим, так как если пользователь будет ограничен в движениях, то для него эти ограничения в некоторых случаях будут фатальны. Все эти условия необходимо учитывать при построении механизма. Но без соответствующих расчетов сделать это невозможно. Именно поэтому было принято решение разработать информационную систему, которая бы

позволила на основе математической модели рассчитать оптимальные размеры звеньев экзоскелета, а также необходимую мощность мотора, приводящего его в движение [2, 3].

2. Реализация

При разработке моделей необходимо включать только те свойства реальных объектов, которые играют основную роль в исследуемом явлении. В ряде случаев достоверно неизвестно, какие факторы и свойства имеют определяющее значение. Тогда используются результаты тех экспериментов, в которых имеется достаточная информация о других подобных явлениях. Выбор определяющих параметров является главной частью формулировки исследуемой проблемы.

Опорно-двигательная система человека состоит из костей, суставов, связок и мышц. Это основная система, которая оформляет структуру человека и дает ему возможность выполнять основное свойство – двигаться. Движение совершается в местах соединений костей – в суставах. Мышцы обладают основным свойством – сокращаться и приводить, таким образом, в движение рычаги костей.

Разбиение тела человека на звенья позволяет представить эти звенья как механические рычаги и маятники, потому что все эти звенья имеют точки соединения, которые можно рассматривать либо как точки опоры (для рычага), либо как точки отвеса (для маятника).

Рычаг – твердое тело, которое может вращаться вокруг оси под воздействием определенных сил (таких сил должно быть не меньше двух). Кости, к которым мышцы прилагают свою силу, с точки зрения механики рассматриваются как рычаги, приводимые во вращательные движения.

Основной закон рычага: рычаг находится в равновесии в случае равенства моментов приложенных к нему сил. При неравенстве моментов происходит вращение в сторону большего момента.

Рука человека представляет собой рычаг третьего рода. Рис. 1 представлен в виде принципиальной схемы данного рычага.

Рычаг скорости (рычаг третьего рода) отличается тем, что активная сила приложена к короткому плечу, из-за чего происходит выигрыш в скорости и расстоянии, но проигрыш в силе. Экзоскелет сможет восполнить данный проигрыш с минимальной потерей в скорости.

В рамках поставленной задачи была построена модель экзоскелета верхней конечности, для которой учитываются такие параметры как длина звеньев, их вес, угол расположения относительно координатных осей и друг друга.

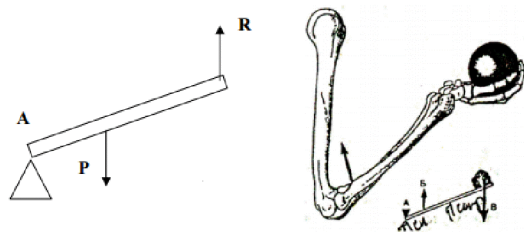


Рис. 1. Рука человека как рычаг третьего рода

На рис. 2 изображена построенная модель с учетом всех необходимых параметров.

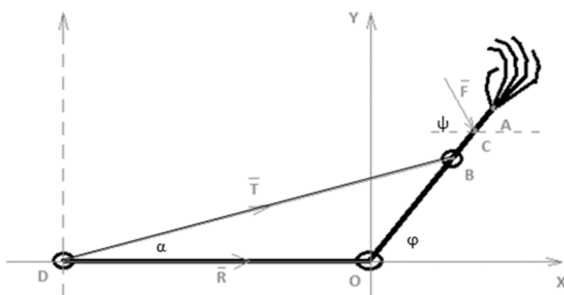


Рис. 2. Визуализация модели экзоскелета руки

Разработанная информационная система способна на примере сгибания и разгибания локтевого сустава в плоскости рассчитать необходимый крутящий момент двигателя, чтобы привести предплечье в движение из любого начального положения в любое конечное с подъемом полезного груза.

Также были рассчитаны оптимальные значения длин звеньев в пределах человеческих параметров, необходимый крутящий момент для поднятия груза весом в заданном интервале. Расчеты проводились по трем разным критериям:

1. минимальное энергопотребление двигателя;
2. минимальный крутящий момент;
3. максимальная скорость реакции.

Заключение

Актуальность работы определяется широкими перспективами использования данной системы при конструировании экзоскелета для людей различной комплекции, а также для принятия оптимального решения по выбору двигателя для механизма.

Разработанную информационную систему в дальнейшем можно расширить в плане области проводимых расчетных экспериментов с компьютерной моделью экзоскелета руки. Полученные результаты будут использованы при разработке прототипа и, в итоге, действующего механизма. Исследования проводятся в лаборатории информационных технологий в медицине кафедры цифровых технологий ФКН ВГУ.

Список литературы

1. Hirose S. Biologically Inspired Robots: Snake-Like Locomotors and Manipulators / S. Hirose. – Oxford : Oxford University Press, 1993. – 220 с.
2. Бранков Г. Основы биомеханики / Г. Бранков. – М. : Мир, 1981. – 254 с.
3. Макаров И.М. Робототехника: История и перспективы / И.М. Макаров, Ю. И. Топчсев. – М. : Издательство "МАИ", 2003. – 348 с.

РАЗРАБОТКА МОДЕЛЕЙ И АЛГОРИТМОВ ФОРМИРОВАНИЯ АНАЛИТИЧЕСКОЙ ОТЧЕТНОСТИ В СИСТЕМЕ SAP BW НА ОСНОВЕ СТАТИСТИЧЕСКИХ ДАННЫХ, АГРЕГИРУЕМЫХ В ВИДЕ OLAP-КУБОВ

Д.Н. Молодых
Студент
А.А. Сирота
Профессор

Введение

Развитие современных предприятий происходит в направлении автоматизации их деятельности на базе корпоративных информационных систем (КИС), позволяющих оптимизировать организационные бизнес-процессы по времени и ресурсам. КИС может быть реализована в форме ERP-системы, системы workflow, с некоторыми допущениями – системы класса groupware и др. Общепризнанным инструментом управления всеми ресурсами в масштабе всего предприятия является система класса ERP – она сокращает временные затраты на получение и обработку информации и позволяет достигнуть интеграции различных подразделений и функциональных областей организации за счет использования единой базы данных и стандартных процессов обработки деловых операций. В результате интеграции современных ERP-систем со средствами бизнес-интеллекта, средствами автоматизации коллективной работы, расширяющих их функционал, пользователи системы получают

возможность работать как с неструктурированной, так и со структурированной информацией, обмениваться знаниями между собой, производить глубокий анализ информации. В этой связи в качестве КИС в настоящей работе выбрана современная ERP-система.

КИС накапливают огромные массивы данных и информации и ориентированы на удовлетворение ежедневных информационных потребностей сотрудников организации, связанных с текущими бизнес-функциями. В то же время в процессе их выполнения у сотрудников возникает потребность в дополнительных знаниях, инициируемая возможностью и необходимостью выбора способа выполнения функции, входных ресурсов, установкой требований к результату функции и его дальнейшему использованию, а в конечном итоге направленная на повышение качества выполнения функции.

На определенном этапе некоторые компании попадают в ситуацию, когда возможности имеющихся ERP систем уже не позволяют получать качественную аналитическую информацию и необходимую скорость для принятия управленческих решений. Поэтому возникает потребность в использовании различных хранилищ данных и дополнительных аналитических инструментов [1].

Широкое распространение различных информационных систем связано с увеличением числа решаемых прикладных задач и значительным возрастанием нагрузок на вычислительные системы. Поставщики сетевых сервисов, опираясь на крупные консолидированные центры обмена данных, особое внимание стали уделять совершенствованию методов планирования задач обработки данных.

Требования сокращения временных издержек на решение прикладных задач, упрощения процедуры сопровождения систем обработки данных в существующих условиях обосновывают актуальность разработки новых подходов построения информационных систем.

1. Информационная структура

Проектное решение базируется на подходе, в основе которого лежит принцип построения информационной пирамиды, определяющей связи между стратегическими целями компании, ее бизнес-процессами и задачами ключевых участников бизнес-процессов (рис. 1).

Ключевые индикаторы процессов (уровень 1 – Стратегический уровень управления бизнес-процессами) отражают эффективность бизнес-процессов и позволяют руководству оперативно контролировать и принимать решения, направленные на достижения стратегических целей компании.

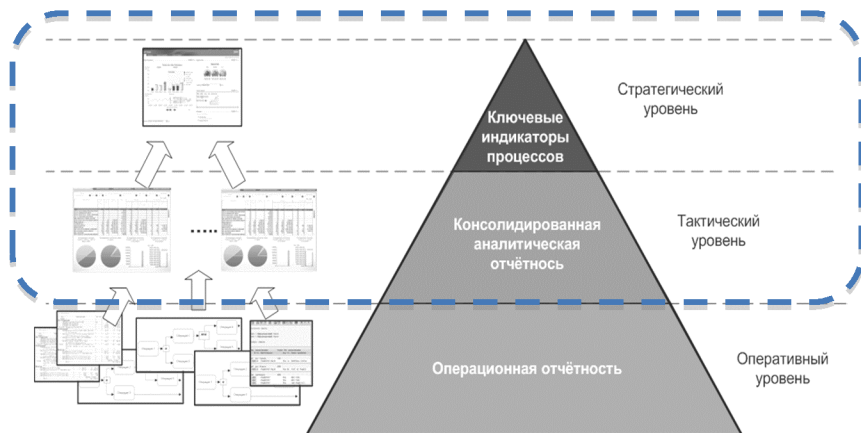


Рис. 1. Структура информационной пирамиды

Консолидированная аналитическая отчётность (уровень 2 – Тактический уровень управления бизнес-процессами) является инструментом анализа причин и тенденций того или иного ключевого индикатора процесса. Инструмент аналитической отчётности данного уровня позволит:

- Владельцам бизнес-процессов оперативно принимать решения, направленные на достижение показателей эффективности процессов;
- Аналитикам проводить исследования данных с целью обнаружения «скрытых» (неочевидных) взаимосвязей и трендов, влияющих на ключевые индикаторы процессов;
- Ключевым участникам процессов ориентироваться на достижение стратегических целей компании в решении своих производственных задач [2].

Уровень ключевых индикаторов процессов и уровень консолидированной аналитической отчётности являются уровнями аналитической информационной системы отчётности компании.

Операционная отчётность (уровень 3 – Оперативный уровень управления бизнес-процессами) реализуется средствами транзакционной системы SAP ERP 6.0 и обеспечивает оперативную отчётность в части операционной деятельности участников бизнес-процессов.

2. Архитектура решения

Аналитическая информационная система отчетности реализована с использованием платформы бизнес-аналитики SAP Business Objects BI Platform в части визуализации и анализа данных и центрального

хранилища SAP Business Information Warehouse (SAP BW) в части хранения исторических данных (хранилище информации).

Применение SAP BW в решении по построению аналитической информационной системы позволяет построить центральное хранилище исторической информации для анализа и осуществлять периодическую загрузку данных путем бесшовной интеграции с основными источниками данных шаблонных решений, реализованных в SAP ERP 6.0.

Использование двухуровневого ландшафта SAP BW и архитектуры LSA в решении по построению аналитической информационной системы будет обеспечивать:

- Улучшенную масштабируемость решения;
- Разделение витрин данных и источников;
- Сбор, преобразование, хранение исторических данных и управление ими;
- Трансформацию данных, расширение набора аналитических признаков для целей анализа;
- Многопользовательский режим работы;
- Авторизацию пользователей и регламентацию доступа к данным.

Для визуализации и анализа данных предназначена аналитическая платформа SAP Business Objects, которая предоставляет средства построения интерактивных, основанных на Flash технологии, информационных панелей, а также средства формирования нестандартных аналитических запросов и поиска бизнес-данных.

При построении аналитической информационной системы отчетности «Газпром нефть» планируется использовать следующие компоненты аналитической платформы SAP Business Objects:

SAP BusinessObjects Dashboards – инструмент визуализации бизнес-данных, позволяющий создавать интерактивные информационные панели с большим набором графических элементов (спидометры, графики, диаграммы, интерактивные карты и т.д.);

BusinessObjects Web Intelligence – инструмент построения расширенных аналитических отчетов и формирования нестандартных запросов с возможностью для интерактивного анализа через математические формулы, встроенные функции ранжирования и сортировки, имеет мощный математический инструментарий для работы с полученными данными, настройки различного рода уведомлений об отклонении значений от заданных параметров.

Direct Access(BICS) – протокол прямого подключения, позволяющий связываться инструментам SAP BO BI непосредственно с BEx-запросом SAP BW.

Архитектура решения должна быть утверждена на Техническом совете Компании. Общая архитектура решения приведена на рис 2.

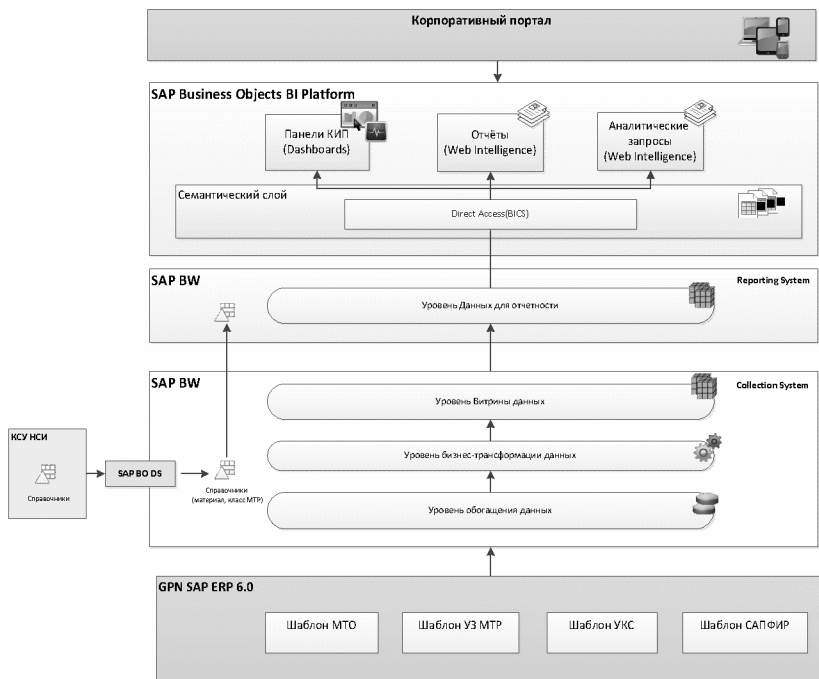


Рис. 2. Архитектура решения

Заключение

В результате выполненной работы:

1. Проведен анализ построения корпоративных информационных систем.
2. Разработан общий подход, теоретические положения и формализованная модель формирования аналитической отчетности.
3. Разработана трехслойная архитектура аналитической информационной системы, реализованной при взаимодействии SAP ERP и SAP BI систем.

Рассмотренные в работе задачи являются лишь небольшими фрагментами широкого поля задач комплексной автоматизации управления предприятием. Для массового внедрения таких задач в системы управления необходимы:

- глубокая проработка универсальных математических моделей и алгоритмов;
- разработка архитектуры и программная реализация

тиражируемых систем, решающих поставленные задачи;

– внедрение разработанных тиражируемых систем в качестве дополнительных компонентов систем управления предприятиями [3].

Имеется множество задач, в таких областях, как логистика (снабжение, производство и сбыт), финансы, маркетинг, управление персоналом, управление качеством и других, которые пока что ещё не реализованы в ERP-системах. Даже если эти задачи реализованы на отдельных предприятиях, то представляют собой уникальные разработки для конкретных систем управления.

Список литературы

1. Беккер И. Менеджмент процессов / И. Беккер, Л. Вилкова, В. Таратухина, М. Кугелер и др. – М. : Эксмо, 2007. – 178 с.
2. Дикерсбах И.Т. Планирование и управление производством с помощью SAP ERP / И.Т. Дикерсбах, Г. Келлер. – С.-Петербург : Эксперт РП, 2011. – 211 с.
3. Hilgefort I. SAP Business objects Profitability and Cost Management: learning course / I. Hilgefort. – Waldorf : SAP AG, 2007. – 129 с.

ПОСТРОЕНИЕ ИМИТАЦИОННОЙ МОДЕЛИ УПРАВЛЕНИЯ ЗАПАСАМИ НА СКЛАДЕ СРЕДСТВАМИ BUSINESS STUDIO

В.А. Назарьева

Студент

М.Г. Матвеев

Профессор

Введение

Современные представления об управлении предприятием базируются на том, что бизнес необходимо четко определять, измерять, анализировать и улучшать. Создание модели бизнес-процессов предприятия позволяет рассмотреть и осуществить первоначальный анализ его деятельности. Более глубокий анализ бизнес-процессов на предприятии возможно осуществлять, используя метод "имитационного моделирования" [1]. Механизм "имитационного моделирования" позволяет оптимизировать бизнес-процессы предприятия, не проводя эксперименты над компанией и сотрудниками.

Современный крупный склад – это сложная система, которая состоит из многочисленных взаимосвязанных элементов, имеет определенную структуру и выполняет ряд функций по преобразованию

материальных и информационных потоков. Во время слаженной работы склада осуществляется прием, хранение, переработка и распределение грузов и сопроводительной документации между потребителями [2].

Бизнес-процессы склада, для эффективной его работы, необходимо анализировать и улучшать. Проблема оптимизации бизнес-процессов склада возникает у многих предприятий и является весьма актуальной, поскольку излишки запаса на складе замораживают денежные средства и снижают финансовую активность, а их недостаток вызывает перебои в работе предприятия [3].

1. Постановка задачи

В качестве примера рассмотрим небольшой склад. Для обслуживания транспорта на складе предусмотрено два разгрузочных поста, два грузчика, два кладовщика, бухгалтер и четыре дополнительных парковочных места для машин на территории склада. Машины с грузом на склад прибывают не в определенное время, а в случайные моменты времени, т.е. интервал между прибытием машин – случайная величина. Количество груза, которое находится в машине – тоже величина случайная, следовательно, время разгрузки товара для каждой прибывшей на склад машины является случайным. Из-за хаотичного прибытия машин на склад, возникает проблема увеличения времени простоя транспорта и неравномерной загрузки персонала склада.

Цель работы: проверить, проведя имитационное моделирование, хватит ли пропускной способности склада для того, чтобы обслужить все машины, приезжающие на склад.

2. Задание параметров модели для имитации

Рассматриваемый в качестве примера склад осуществляет прием машин с 9:00 до 15:00. Машины, прибывшие позже, в этот день не обслуживаются. Интервал между прибытием машин составляет 30 минут. Но, принимая во внимание всю сложность прогнозирования точного времени, в течение которого машина будет находиться в пути на склад, машины могут подъехать как раньше назначенного времени, так и позже. Поэтому для моделирования интервала между прибытием машин используем нормальный закон распределения с матожиданием 30 минут, стандартным отклонением 5 минут и возможным диапазоном значений от 5 минут до 1 часа (рис. 1.).

Количество разгруженных машин на складе напрямую зависит не только от времени разгрузки одной машины, а также от времени выполнения каждого шага бизнес-процесса. Время выполнения шага процесса может быть фиксированной или случайной величиной. Длительности шагов бизнес-процесса «Разгрузка и обслуживание машин на складе» приведены в таблице 1.

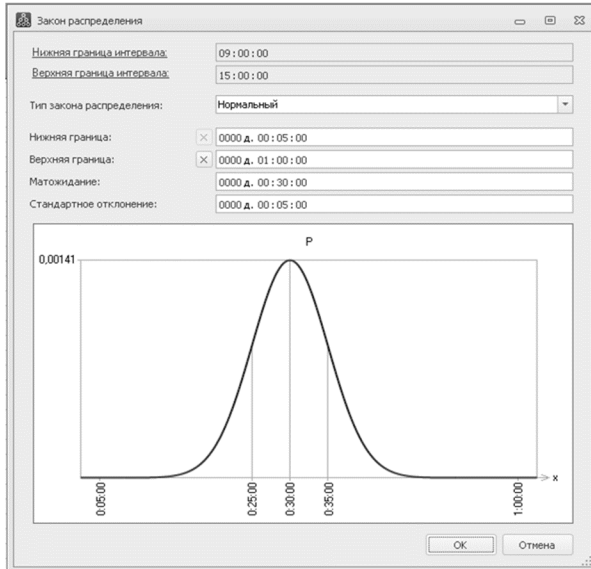


Рис. 1. Параметры закона распределения

Таблица 1

Длительности шагов процесса

Шаг процесса	Длительность
1	2
Проверка наличия свободных мест разгрузки	1 минута
Проезд на разгрузку	2 минуты
Проверка наличия мест на парковке	1 минута
Проезд на парковку	2 минуты
Ожидание на парковке	0 (время моделируется системой автоматически)
Проезд с парковки на разгрузку	2 минуты
Ожидание на дороге	0 (время моделируется системой автоматически)
Проезд с дороги на парковку	2 минуты
Прием документов	5 минут
Проверка соответствия товара документации	Равномерный закон распределения: — нижняя граница — 1 минута — верхняя граница — 5 минут
Выезд машины со склада	2 минуты

Окончание табл. 1

1	2
Разгрузка товара	Нормальный закон распределения: — нижняя граница — 10 минут — верхняя граница — 1 час — матожидание — 40 минут — стандартное отклонение — 5 минут
Приходование товара	Нормальный закон распределения: — нижняя граница — 5 минут — верхняя граница — 30 минут — матожидание — 10 минут — стандартное отклонение — 1 минута
Размещение товара на хранение	Нормальный закон распределения: — нижняя граница — 10 минут — верхняя граница — 1 час — матожидание — 30 час — стандартное отклонение — 5 минут
Выезд машины со склада	2 минуты

Длительность шага процесса зависит не только от нормативного времени его выполнения, но и от доступности ресурсов (временных и материальных), необходимых для его выполнения. Если необходимых ресурсов в нужный момент времени нет, то выполнение процесса останавливается.

Шаги по обслуживанию машины на складе выполняют сотрудники склада. Перечень и количество ресурсов, необходимых для выполнения каждого шага процесса, приведены в таблице 2.

Таблица 2

Перечень ресурсов, необходимых для выполнения процесса

Шаг процесса	Ресурсы	Количество ресурсов
Прием документов	Бухгалтер	1
Проверка соответствия товара документации	Кладовщик	1
Разгрузка товара	Грузчик	1
Приходование товара	Кладовщик	1
Размещение товара на хранение на складе	Грузчик	1
	Кладовщик	1

Всего на складе работают следующие сотрудники, их список приведен в таблице 3.

Таблица 3

Временные ресурсы

Ресурсы	Название смены	Календарь	Количество экземпляров	Ставка в час
Бухгалтер	Смена 1	Календарь 1	1	100
Кладовщик	Смена 1	Календарь 1	2	100
Грузчик	Смена 1	Календарь 1	2	50

3. Результаты имитации

Проведя имитацию работы склада с заданными параметрами в течение одного рабочего дня, получаем следующие показатели процесса (рис. 2).

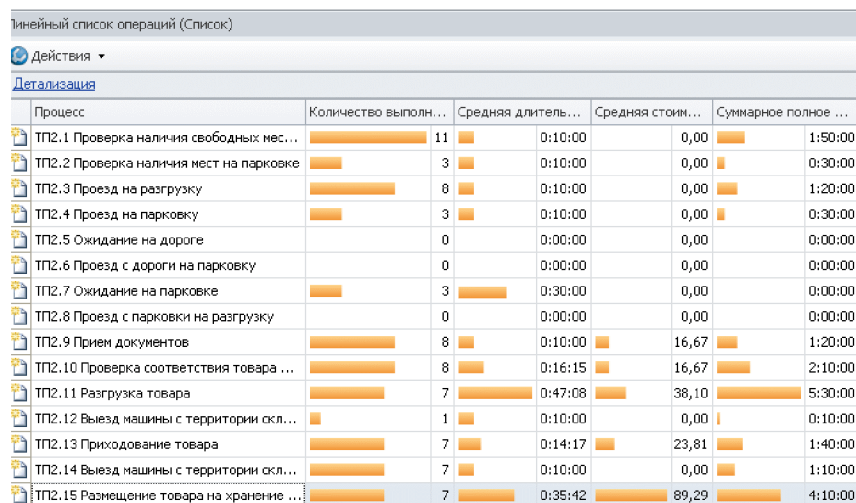


Рис. 2. Параметры шагов процесса, полученные в результате имитации

Анализируя результаты имитации можно сделать следующий вывод: существующей пропускной способности склада недостаточно для обслуживания машин, приезжающих на склад:

- из 11 машин обслужено всего 7 (7 — это количество выполнений процесса "Разгрузка товара");
- товар, привезенный только 7-мью машинами, был размещен на хранение на складе;
- из 11 машин 3 ожидали на парковке (среднее время ожидания 30 минут).

В результате, машины в течение длительного времени ожидают обслуживания, занимая парковочные места и затрудняя движение на складе, при этом одно из 4 парковочных мест практически не используется.

Выходом из сложившейся ситуации может быть решение: переоборудовать одно парковочное место под место разгрузки.

Кроме того анализируя использование временных ресурсов в существующих условиях, можно увидеть, что "бутылочным горлышком" (ресурсом, время ожидания которого при выполнении процесса наибольшее) является кладовщик. Поэтому для повышения пропускной способности склада увеличим количество кладовщиков до 3.

Нельзя не заметить, что размещен на хранение был только товар, привезенный 7 машинами. Очевидно, что оставлять недоделанную работу на следующий рабочий день нецелесообразно, учитывая интенсивность ежедневного потока поступления машин на склад. Поэтому изменим рабочее время сотрудников склада на график: с 9:30 до 19:00.

4. Оптимизация бизнес-процесса.

В результате повторной имитации процесса с параметрами, измененными в соответствии с принятым после анализа решением, показатели работы склада значительно улучшились, и пропускной способности склада стало достаточно, чтобы обслуживать такое количество приезжающих машин:

- из 12 подъехавших машин было обслужено 11, товар 1 машины не соответствовал документации;
- весь товар, привезенный машинами, был размещен на складе;
- снизилось время ожидания машины на парковке (до 15 минут).

Заключение

Таким образом, имитация работы склада в различных условиях позволила предусмотреть возникновение возможных проблем и проверить работоспособность склада.

После проведения первой имитации выяснилось, что пропускной способности склада не хватает для обслуживания такого количества приезжающих машин. Было принято решение переоборудовать место парковки под место разгрузки, добавить еще одного кладовщика на склад и увеличить рабочее время сотрудников склада. В результате, после проведения второй имитации с новыми параметрами, склад стал работать эффективнее, пропускная способность склада повысилась, все приезжающие машины были разгружены, и весь товар был размещен на хранение на складе.

Список литературы

1. Методика: Функционально-стоимостной анализ и имитационное моделирование. – Режим доступа: http://www.businessstudio.ru/procedures/business/fsa_imitacia
2. Никифоров В.В. Логистика. Транспорт и склад в цепи поставок / В.В. Никифоров. – М. : ГроссМедиа, 2008. – 192 с.
3. Быков А. Оптимизация запасов на основе имитационного моделирования / А. Быков // Логистика. – 2004. – №1. – С. 19–21.

ПРИМЕНЕНИЕ МЕТОДА ЧАСТИЦ ДЛЯ МОДЕЛИРОВАНИЯ МЕХАНИЧЕСКОГО ВЗАИМОДЕЙСТВИЯ ТЕЛ

Д.Ю. Новичихин

Студент

В.Г. Хлебостроев

Доцент

Введение

В данной работе рассматривается один из широко известных методов агентного имитационного моделирования – метод частиц, используемый для построения моделей механического взаимодействия тел [1].

Идея метода частиц состоит в том, что моделируемое физическое тело представляется совокупностью взаимодействующих между собой частиц (материальных точек или твёрдых тел), поведение которых описывается законами классической механики. В качестве частиц выбираются лишь наиболее значимые для моделирования точки макроскопических объектов. В качестве потенциалов взаимодействия выступают так называемые связи, которые можно представить себе как упругие стержни переменной длины или пружины, соединяющие по две частицы. Эти связи так же отражают наиболее важные взаимодействия частиц. Построение моделей объектов с использованием данного метода похоже на каркасное моделирование. Метод частиц позволяет создавать физические объекты совершенно различных форм, моделировать как статическое, так и динамическое взаимодействия (деформирование, разрушение).

1. Описание модели

Основными свойствами частиц являются векторные величины:

- мгновенное положение в пространстве (позиция) – $\vec{p}(t)$;

© Новичихин Д.Ю., Хлебостроев В.Г., 2014

- скорость – $\vec{v}(t)$;
- ускорение – $\vec{a}(t)$.

Частицы могут быть зафиксированными на одной позиции для имитации различных креплений. Массы всех частиц полагаются равными единице, поэтому ускорение можно отождествить с силой, действующей на частицу.

Основными свойствами связей являются скалярные величины:

- длина в ненапряжённом состоянии (нормальная длина) – L_{nm} ;
- длина при максимальном сжатии (минимальная длина) – L_{min} ;
- длина при максимальном растяжении (максимальная длина) – L_{max} ;
- жёсткость – S .

Если текущая длина связи отличается от нормальной длины, то частицы приближаются друг к другу или удаляются друг от друга. Чем меньше жёсткость связи, тем медленнее происходит этот процесс. Если текущая длина связи меньше минимальной длины или больше максимальной длины, то связь разрушается.

Для реализации метода частиц используется алгоритм Position-Based. Он весьма прост и имеет всего два основных этапа:

- перемещение частиц в соответствии с их скоростями в текущий момент времени, независимо от связей;
- разрешение связей с учётом лишь парных взаимодействий частиц; при этом подразумевается корректировка их позиций таким образом, чтобы все связи имели длины, наиболее удовлетворяющие состоянию равновесия системы на текущий момент времени.

Расчет траектории перемещения частиц осуществляется методом интегрирования Верле [2]. Его особенность заключается в том, что для вычисления следующей позиции точки вместо скорости используются её текущая и предыдущая позиции:

$$\vec{p}(t + \Delta t) = 2\vec{p}(t) - \vec{p}(t - \Delta t) + \vec{a}(t)(\Delta t)^2. \quad (1)$$

При необходимости на этом этапе происходит ограничение пространства моделирования, которое в трёхмерном случае можно представить в виде произвольного параллелепипеда, плоскости граней которого ортогональны прямым координатных осей. Это ограничение аналогично разрешению связей: если частица вышла за пределы ограничения, то её позиция корректируется.

Операцию разрешения связи продемонстрируем на примере. Пусть в некоторый момент времени две связанные частицы имеют позиции \vec{P}_1 и \vec{P}_2 . Их связь имеет текущую длину L и нормальную длину L_{nm} , причем

$L > L_{nm}$ по условию (Рис. 1). Обозначим также $\bar{D} = \bar{P}_2 - \bar{P}_1$, S – жёсткость связи (принимает значения от 0 до 1 включительно).

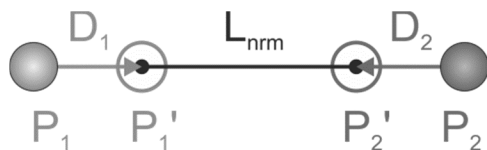


Рис. 1. Разрешение связи

Формулы для вычисления новых позиций \bar{P}'_1 и \bar{P}'_2 имеют вид:

$$\bar{P}'_1 = \bar{P}_1 + \bar{D} \times \left(\frac{L - L_{nm}}{2 \times L} \times S \right). \quad (2)$$

$$\bar{P}'_2 = \bar{P}_2 - \bar{D} \times \left(\frac{L - L_{nm}}{2 \times L} \times S \right). \quad (3)$$

Очевидно, что они справедливы и в случае, когда $L < L_{nm}$.

После разрешения связей система переходит в новое состояние, которое на текущий момент времени является наиболее приближённым к состоянию её равновесия, и алгоритм повторяется для следующего шага по времени. Главный недостаток данного алгоритма заключается в том, что обновлённое состояние системы может быть различным в зависимости от порядка разрешения связей, поскольку разрешение каждой последующей связи может изменить длины предыдущих.

Для исключения подобной зависимости предлагается вести отложенную корректировку позиций частиц. Корректная позиция какой-либо частицы вычисляется несколько раз (если частица имеет более одной связи), после чего усреднённое значение этой позиции заменяет текущую позицию частицы (Рис.).

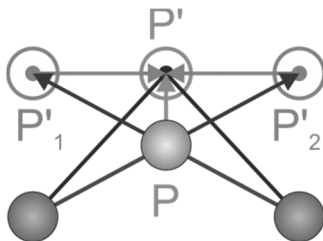


Рис. 2. Отложенная корректировка позиции частицы

Как уже отмечалось, в методе Верле не используются явно заданные

скорости, что даёт ряд преимуществ. Однако на практике использование этих скоростей оказывается более удобным и наглядным. Поэтому удобно модифицировать логику метода Верле, представив формулу (1) в следующем виде:

$$\bar{p}(t + \Delta t) = \bar{p}(t) + \bar{v}(t) \times \Delta t + \bar{a}(t) \times (\Delta t)^2, \quad (4)$$

где
$$\bar{v}(t) \times \Delta t = \bar{p}(t) - \bar{p}(t - \Delta t).$$

Для сокращения объема вычислений удобно произвести следующие замены:

$$\bar{v}_{dt}(t) = \bar{v}(t) \times \Delta t, \quad \bar{a}_{dt}(t) = \bar{a}(t) \times (\Delta t)^2.$$

После этого модифицированная формула метода Верле примет более простой и удобный вид:

$$\bar{p}(t + \Delta t) = \bar{p}(t) + \bar{v}_{dt}(t) + \bar{a}_{dt}(t). \quad (5)$$

Такое изменение формулы приводит к тому, что теперь при любой корректировке позиций частиц необходимо изменять на ту же величину и их «скорости», поскольку слагаемое $\bar{p}(t)$ входит в выражение, определяющее $\bar{v}_{dt}(t)$.

2. Реализация модели

В рамках данной работы было создано приложение-симулятор для демонстрации возможностей метода частиц. В качестве инструментальной среды использовалась среда программирования Delphi, а вывод трёхмерной графики осуществляется средствами библиотеки OpenGL. Приложение содержит следующие модули, выражающие логику работы программы:

- **UFormMain.** Главный модуль, содержащий описание класса окна приложения TFormMain. В нём объединяется функциональность интерфейса и всех остальных модулей;

- **UGraph3D.** Этот модуль содержит описание класса TGraph3D, который отвечает за вывод трёхмерной графики и всё, что с этим связано. Он позволяет реализовывать прорисовку в одном или нескольких контекстах устройств, объединяет функции OpenGL для упрощения кода прорисовки, содержит функции для управления (перемещения, вращения) точкой обзора (камерой). Также этот класс позволяет создавать интерфейс приложения, дающий пользователю возможность выбирать отображаемые объекты каким-либо способом (в разрабатываемом приложении с помощью мыши осуществляется выбор частиц и связей);

- **UPartsSystem.** В этом модуле описывается наиболее значимый класс приложения – TPartsSystem (система частиц). Помимо реализации алгоритма метода частиц, он содержит также функции для различных

манипуляций с системой частиц и связями между ними, например, добавление и удаление частиц и связей, установка их свойств, копирование объекта (выбранных частиц с их возможными связями), объединение нескольких частиц в одну и некоторый другие;

- **UParticle**. Модуль с описанием вспомогательного класса TParticle, который содержит все необходимые свойства частицы. Используется в классе TParticleSystem;

- **UJoint**. Модуль с описанием аналогичного вспомогательного класса TJoint, включающего в себя свойства связи. Так же используется в классе TParticleSystem;

- **UVector3D**. Здесь описывается класс TVector3D, содержащий функции для работы с векторами в трёхмерном пространстве, которые широко используются при создании приложения.

Таким образом, каждый из шести описанных модулей содержит в себе реализацию единственного одноимённого класса. Это удобно, поскольку существует возможность легко устанавливать различные зависимости между классами, распространять их и использовать в любых приложениях и в любых интерпретациях.

Интерфейсная часть приложения включает средства задания пространственного положения частиц и связей между ними. На рис. 3 представлена главная форма приложения, содержащая окно графического вывода и панель управления.

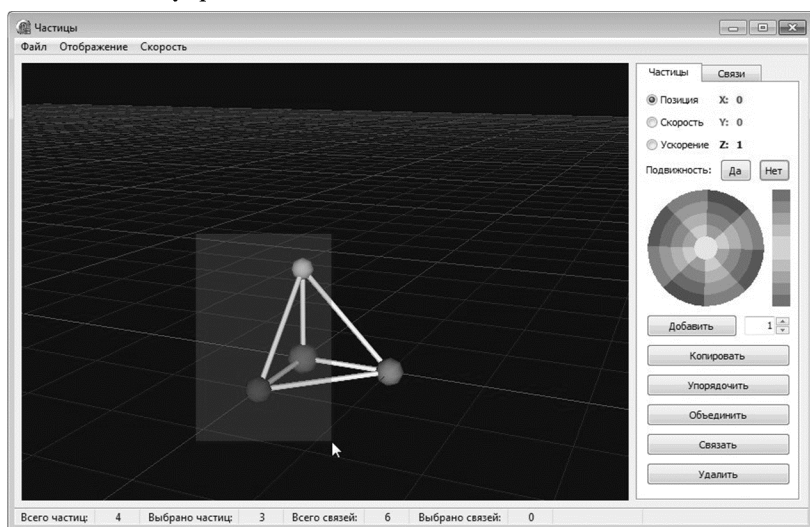


Рис. 3. Главная форма приложения с открытой вкладкой «Частицы»

Приведём пример моделирования разрушения объекта при соударении с поверхностью земли (нижней ограничивающей плоскостью) под действием силы тяжести. Связи разрушаются при растяжении, и частицы разлетаются по сторонам (рис. 4- 6).

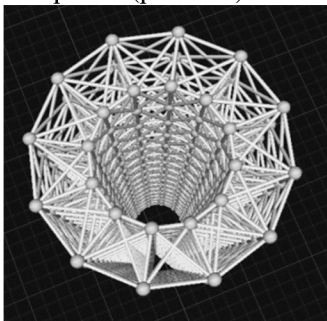


Рис. 4. Моделирование разрушения объекта (начальное состояние, вид сверху)

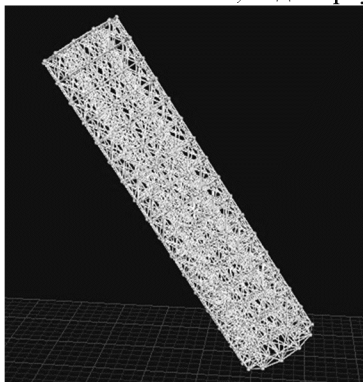


Рис. 5. Моделирование разрушения объекта (начальное состояние, вид сбоку)

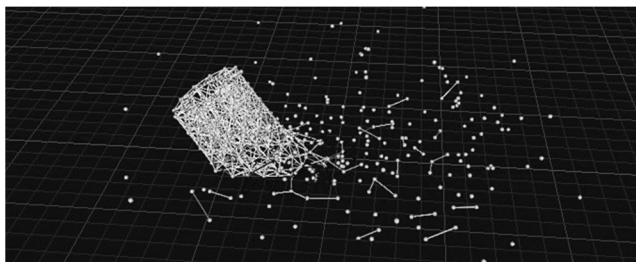


Рис. 6. Моделирование разрушения объекта (связи разрушаются при растяжении)

Заключение

В рамках данной работы было создано демонстрационное приложение, позволяющее моделировать механическое взаимодействие тел с использованием метода частиц. Тестирование и визуальная оценка моделей позволяют сделать вывод о том, что их поведение при определённых условиях достаточно правдоподобно. Таким образом, метод частиц является эффективным и перспективным способом моделирования.

Список литературы

1. Хокни Р. Численное моделирование методом частиц / Р. Хокни, Дж. Иствуд. – М. : Мир, 1987. – 640 с.
2. Пестряев, Е.М. Молекулярная динамика на персональном компьютере: учеб. пособие / Е.М. Пестряев; под. ред. В.Д. Скирды и др. // Сб. статей. «Наука-фундамент решения проблем технологического развития России». – Ч.1. – Казань: Казанский университет. – 2012. – С. 3–104.

ПРОГРАММНО-АППАРАТНЫЙ КОМПЛЕКС АНАЛИТИКИ ПОВЕДЕНИЯ ЛЮДЕЙ НА ОСНОВЕ ТЕХНОЛОГИИ WI-FI

А.В. Нужных
Студент

И.С. Губарев
Ассистент

Введение

Владельцам и менеджерам торговых объектов важно иметь возможность объективно оценивать качество работы персонала, эффекты от проведения маркетинговых акций и оформления витрин. Одним из способов получения такой информации является наблюдение за поведением посетителей. Основными количественными характеристиками поведения могут быть отношение количества посетителей к общему числу прохожих, время, проведённое в магазине и количество постоянных посетителей. Также немалый интерес представляют сравнения описанных выше показателей за разные промежутки времени. Для получения этой информации было решено разработать описываемый комплекс.

1. Аналитическая часть

Для получения исходной информации было решено использовать возможности технологии WiFi. Важным этапом работы рассматриваемой

технологии является поиск доступных беспроводных локальных сетей. Этот процесс подробно описан в наборе стандартов для беспроводной связи IEEE 802.11 и заключается в рассылке специальных probe request пакетов. Получая эти пакеты, можно оценить уровень сигнала и выявить, находится ли человек в помещении в этот момент.

По результатам исследований [1] и [2] более трети людей активно используют устройства с WiFi модулем. Это даёт возможность оценивать тенденцию поведения людей. Авторы статьи [3] пришли к выводу, что большая часть современных смартфонов рассылает необходимые probe request пакеты.

2. Реализация

Состав комплекса схематично изображён на рис. 1. В качестве устройств, принимающих probe request пакеты, используются обычные роутеры со встроенными WiFi адаптерами. Все собранные данные периодически отсылаются на сервер в сети Интернет. После получения данные сохраняются в базу данных (БД) и запускается подсистема локализации, позволяющая определить, в какое время обнаруженные устройства находились внутри помещений. По результатам работы рассчитывается статистика, на основе которой конечному пользователю генерируется итоговая информация. Доступ к такой информации в виде графиков пользователь получает через Web-интерфейс.

Система состоит из следующих компонентов:

- Маршрутизаторы;
- Сервер приёма данных;
- Анализ данных;
- Представление результатов анализа в виде графиков.

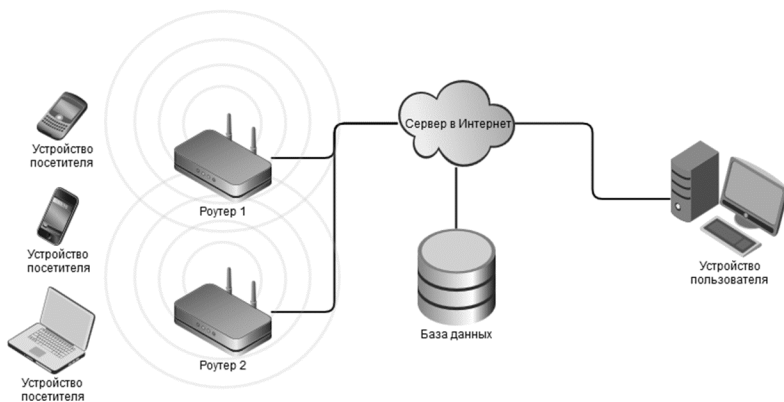


Рис. 1. Общая схема комплекса

Для корректной работы маршрутизаторов была собрана собственная версия прошивки OpenWRT, включающая в себя все необходимые утилиты и скрипты. Для сбора данных используется свободно распространяемая утилита tcpdump, позволяющая, переведя сетевой интерфейс в режим мониторинга, отслеживать передаваемые пакеты. Результаты работы этой утилиты передаются в программу processing.exe, которая выделяет необходимую информацию и в сжатом виде сохраняет на долговременную память. Используя стандартную утилиту cron, происходит периодическая отправка собранных данных на сервер. При этом происходит их шифрование алгоритмом RSA с использованием библиотеки OpenSSL.

После приёма данных на сервере и успешного дешифрования происходит их импорт в БД. При этом выделяются следующие сущности:

- AccessPoint – маршрутизатор, с которого пришли данные;
- Dump – набор двоичных данных за определённый промежуток времени;
- DumpRequest – набор нескольких дампов, а также информация о текущем состоянии точки доступа;
- Device – обнаруженное устройство;
- Record – информация об одном probe request пакете.

Так как устройства обычно посылают несколько probe request пакетов одновременно, то записи об этих пакетах объединяются в короткие события (ShortEvent). На их основе создаются общие короткие события (CommonShortev), которые являются аналогом коротких событий в рамках одной локации. Общие события в свою очередь объединяются в длинные (LongEvent). Каждое длинное событие описывает промежуток времени, в течение которого устройство находится в зоне видимости.

Далее запускается подсистема локализации, по результатам работы которой для каждого длинного события определяется момент входа и выхода из помещения. По полученным данным заполняются таблицы распределения количества прохожих и посетителей за каждый час (EntryIgnoreDistribution), а также таблица распределения по времени пребывания внутри помещения (IndoorDistribution). На рис. 2 представлена общая схема получившейся базы данных. По этим данным формируется графики, выводимые конечному пользователю.

На рис. 3 показан пример графика распределения по времени пребывания посетителей на разных торговых объектах.

Из этого графика можно получить информацию о том, насколько хорошо работают продавцы в отдельных магазинах. На рис. 4 показано распределение количества прохожих и посетителей в зависимости от времени суток.

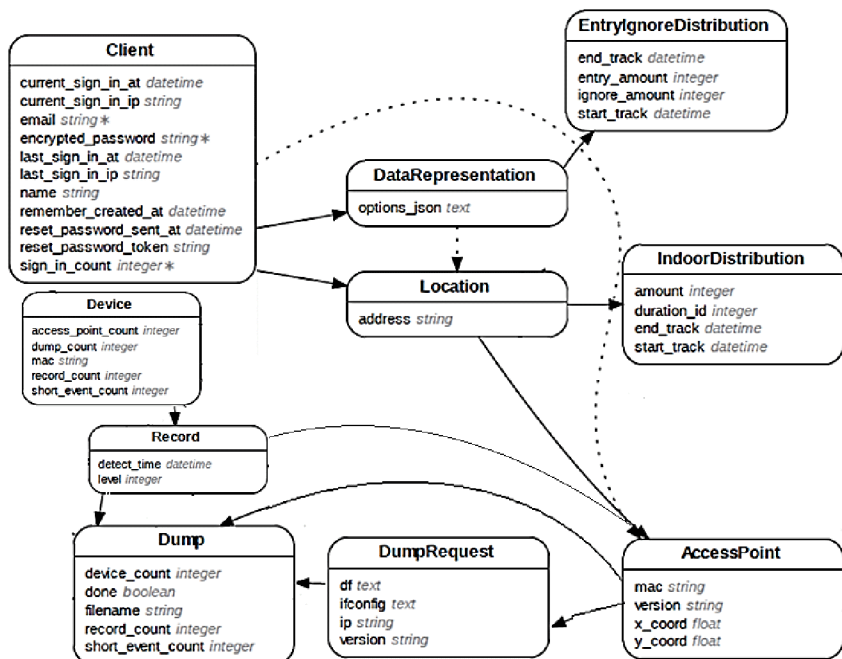


Рис. 2. Схема базы данных

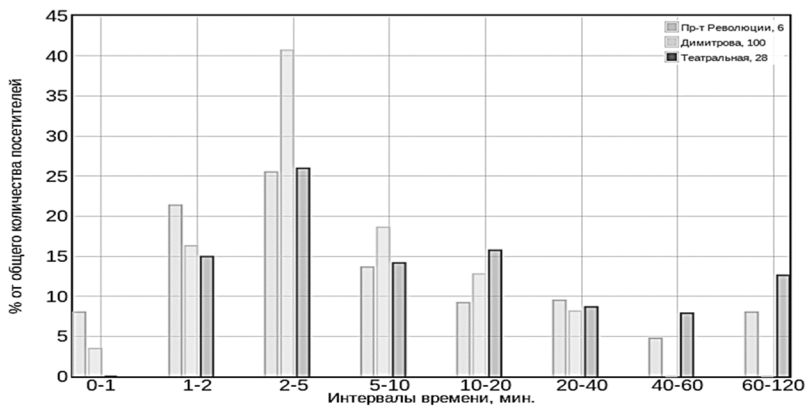


Рис. 3. Распределение по времени пребывания посетителей

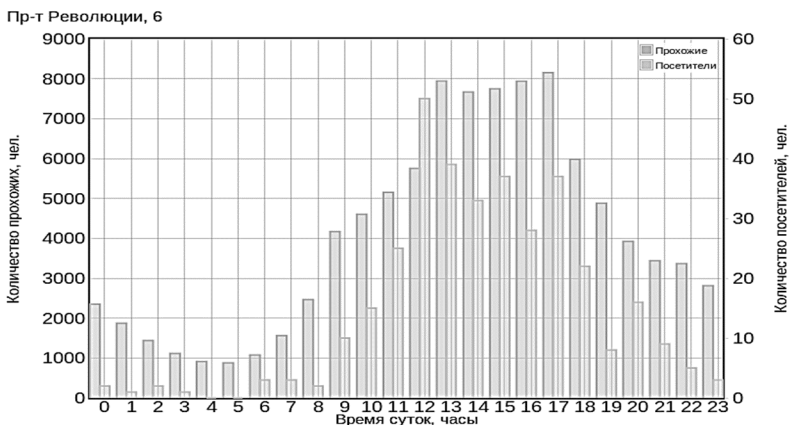


Рис. 4. Распределение количества прохожих и посетителей по времени суток за месяц

По этому графику можно определить оптимальные часы работы магазина. Кроме графиков система позволяет представлять информацию в виде индикаторов. Пример индикатора представлен на рис. 5.

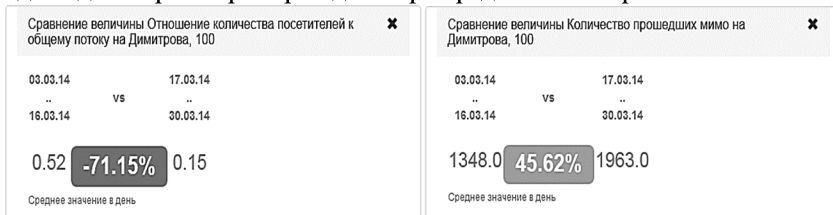


Рис. 5. Примеры индикаторов

Этот инструмент позволяет оценивать изменение определённой величины за разные промежутки времени. Например, его можно использовать для оценки эффекта от проведения маркетинговых акций или оформления витрин, когда сравниваются значения показателей до изменения и после.

Заключение

В итоге был разработан программно-аппаратный комплекс, позволяющий оценивать поведение посетителей различных торговых объектов. Наиболее востребована данная система для торговых объектов, ориентированных на людей, которые активно используют в своей повседневной жизни мобильные гаджеты. В настоящее время проходит тестирование комплекса в реальных условиях.

Список литературы

1. Крекнина А. Как смартфоны помогут ретейлерам. – Режим доступа: http://www.vedomosti.ru/companies/news/13467551/pokupatel_golosuet_smartfonom?full#cut
2. До 36% пользователей предпочитают смартфоны. – Режим доступа: <http://kapital.kz/tehnology/19012/do-36-polzovatelej-predpochitayut-smartfony.html>
3. Sidiropoulos N. Open Wifi SSID Broadcast vulnerability. – Режим доступа: https://www.os3.nl/_media/2012-2013/courses/ssn/open_wifi_ssid_broadcast_vulnerability.pdf

ONLINE-СЕРВИС ДЛЯ СКРЫТОГО АННОТИРОВАНИЯ ФАЙЛОВ ГРАФИЧЕСКИХ ФОРМАТОВ

А.С. Пашенцева
Студент

Е.Ю. Митрофанова
Ассистент

Введение

В настоящее время, в связи с увеличением объема различной информации увеличивается доля сведений, которые необходимо держать в тайне. Кроме того, с появлением глобальных компьютерных сетей стало легко получить доступ к информации, что значительно повысило угрозу нарушения безопасности данных, если не применять меры относительно их защиты. В связи с интенсивным развитием и распространением технологий, которые позволяют с помощью компьютера интегрировать, обрабатывать и синхронно воспроизводить различные типы сигналов вопрос защиты информации, представленной в цифровом виде, является чрезвычайно актуальным.

1. Теоретический анализ

Стеганография - наука, которая изучает способы и методы скрытия конфиденциальных сведений. Задача извлечения информации при этом отстывает и является второстепенной и решается в большинстве известных случаев стандартными криптографическими методами. При этом под скрытием существования информации подразумевается не только невозможность обнаружения в перехваченном сообщении наличия скрытого сообщения, но и вообще сделать невозможным возникновение любых подозрений на этот счет, поскольку в последнем случае проблема

информационной безопасности возвращается к стойкости криптографического кода.

Подавляющее большинство методов стеганографии базируется на двух ключевых принципах:

1) файлы, которые не требуют абсолютной точности (например, файлы с изображением, звуковой информацией и т.д.), могут быть видоизменены до определенной степени без потери своей функциональности;

2) органы чувств человека не способны надежно различать незначительные изменения в модифицированных таким образом файлах или отсутствует специальный инструментарий, который был бы способен выполнять данную задачу.

В основе базовых подходов к реализации методов стеганографии в рамках той или иной информационной среды лежит выделение малозначительных фрагментов этой среды и замена существующей в них информации информацией, которую необходимо скрыть. То есть скрываемое сообщение встраивается в объект, который затем открыто транспортируется [1].

Существует довольно много программных продуктов, которые применяются для различных целей и реализующих, как правило, средства внедрения секретных данных в графические, звуковые и видео файлы. Но для их использования пользователю необходимо устанавливать такие программные продукты на свой компьютер, поэтому предпочтительнее и удобнее пользоваться online-сервисами. На основе вышесказанного, можно выделить следующие ограничения, накладываемые на стеганографические сервисы, а именно, ограничения на формат файла, неудобный интерфейс, ограничение на размер исходного изображения, ограничение на размер встраиваемого файла и вводимое сообщение. Таким образом, разработка online-сервиса для скрытого аннотирования файлов представляется актуальной.

Представляется целесообразным разработать online-сервис, позволяющий пользователям:

- 1) загружать изображение;
- 2) встраивать информацию в загруженное изображение в виде текстового сообщения и/или файлового архива;
- 3) получать скрываемую информацию из изображения, если таковая имеется;
- 4) иметь возможность аутентификации для хранения истории изображений и использования более стойкого стеганографического метода;
- 5) осуществить обратную связь для повышения качества сервиса.

В качестве средств реализации данного online-сервиса был выбран фреймворк Ruby on Rails. В результате в данном online-сервисе реализованы следующие методы:

- 1) метод замены наименее значащих бит;
- 2) метод относительной замены величин коэффициентов ДКП (метод Коха и Жао).

Метод замены наименее значащих бит является доступным всем пользователям сервиса, а метод относительной замены величин коэффициентов ДКП (метод Коха и Жао) только зарегистрированным и авторизованным пользователям.

2. Реализация

Данное приложение выполнено в виде шести разделов. Главная страница представлена в виде статической страницы, которая содержит в себе описание сервиса. На рис. 1 представлено изображение главной страницы.

Раздел скрытия информации позволяет пользователю загрузить изображение, в которое необходимо скрыть информацию. На рис. 2 представлено изображение страницы загрузки изображения.

При успешной загрузке, пользователю представляется интерфейс для заполнения информации, в виде текстового поля и виджета загрузки файлового архива. На рис. 3 представлено изображение страницы скрытия информации.

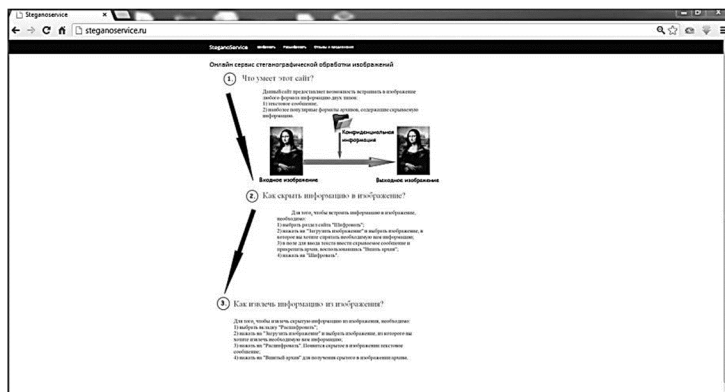


Рис. 1. Главная страница



Рис. 2. Загрузка изображения



Рис. 3. Скрытие информации

Клиентская сторона данного раздела реализована с помощью javascript – библиотеки *qq.FileUpload*. Она осуществляет асинхронную загрузку изображения на сервер, позволяющую вычислить максимальный объем скрываемых данных. После успешной загрузки изображения в ответ клиенту приходит адрес сохраненного изображения для отображения на форме, а также вычисленный предел в байтах по размеру скрываемых данных. Данное число необходимо для валидации формы при заполнении информации и деактивации поля ввода текста при превышении пользователем данного порога. После заполнения информации отправляется заполненная форма, в ответ на запрос клиенту посылается файл изображения со скрытой информацией. В случае, если пользователь аутентифицирован, ему предоставляется возможность выбора метода скрытия т.е. методов замены наименее значащих бит или Коха-Жао. Раздел получения информации практически повторяет функциональность вышеописанного раздела, за исключением того, что пользователю в ответ приходит скрытая информация в виде формы, содержащей скрытый текст и/или вшитый архив. На рис. 4 представлено изображение страницы получения скрытой информации. На рис. 5 представлено изображение страницы при успешном получении скрытой информации.



Рис. 4. Страница получения скрытой информации



Рис. 5. Успешное получение скрытой информации

Раздел «Отзывы о проекте» обладает минимальной функциональностью обратной связи. При заполнении и отправки формы информация сохраняется в базе данных. На рис. 6 представлено изображение страницы отзывов и предложений.

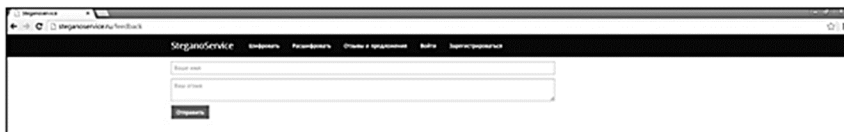


Рис. 6. Отзывы и предложения

Так же следует отметить, что разделы скрытия и получения информации были названы «Шифровать» и «Расшифровать» соответственно ввиду того, что для конечного пользователя эти термины наиболее понятны. Раздел регистрации пользователя позволяет пользователю осуществить регистрацию для получения дополнительных возможностей сервиса. На рис. 7 представлено изображение страницы регистрации приложения.

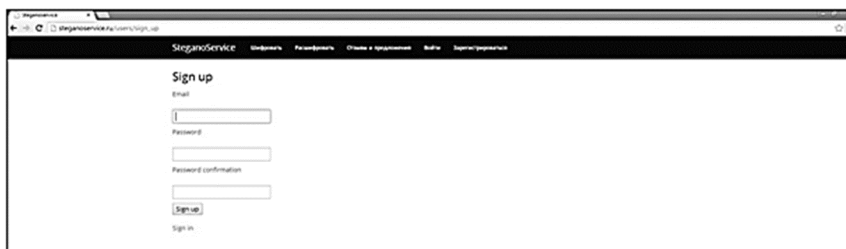


Рис. 7. Регистрация приложения

После регистрации пользователю становятся доступны следующие возможности.

1. **Мои изображения** – это страница, отображающая историю операций, выполненных на данном сервисе. Здесь находится информация о дате создания изображения со скрытой информацией, скрытое сообщение, исходное изображение и изображение с данными. Пользователю предоставляется возможность, без осуществления повторной загрузки изображения, изменять скрытую информацию в существующих изображениях. На рис. 8 представлено изображение личного кабинета пользователя.



Рис. 8. Мои изображения

2. Метод Коха-Жао. На рис. 9 представлено изображение страницы скрытия информации с выбором методов.



Рис. 9. Возможность использования метода Коха-Жао

Раздел аутентификации позволяет пользователю выполнить вход в систему, если ранее он был зарегистрирован на данном сервисе. На рис. 10 представлено изображение страницы аутентификации пользователя.



Рис. 10. Аутентификация пользователя

Основой для функциональности авторизации\аутентификации пользователей является библиотека devise, которая наиболее популярна сегодня в среде фреймворка *Ruby on Rails*. Она позволяет посредством генераторов создать необходимые представления, модели, миграции, тесты и, кроме того, имеет отличную совместимость с наиболее популярными библиотеками, в частности с *CanCan* служащей для управления правами пользователей. В данном проекте использована базовая инсталляция

библиотеки, служащая только для типовых вариантов использования пользователем:

- 1) «Войти»;
- 2) «Зарегистрироваться»;
- 3) «Восстановить пароль».

Заключение

В результате были проведены работы по исследованию и реализации таких стеганографических методов, как метод замены наименее значащих бит и метод относительной замены величин коэффициентов ДКП (метод Коха и Жао). А так же разработан и реализован online-сервис для скрытого аннотирования файлов графических форматов.

Список литературы

1. Конахович Г.Ф. Компьютерная стеганография. Теория и практика / Г.Ф. Конахович, А.Ю. Пузыренко. – Киев : МК-Пресс, 2006. – 288 с.

РЕШЕНИЕ ПРОБЛЕМЫ РАСПОЗНАВАНИЯ ПЕЧАТНЫХ ТЕКСТОВ С ИСПОЛЬЗОВАНИЕМ СРЕДСТВ БИБЛИОТЕКИ OPENCV

И.М. Перелётов
Студент

В.Г. Хлебостроев
Доцент

Введение

С развитием вычислительной техники и созданием форматов хранения данных в электронном виде, проблема перевода информации из одного формата в другой становится всё более актуальной. Большую долю информации составляют тексты, и существует способ переводить её в электронный вариант путём сканирования. Однако это создаёт другую проблему: отсканированный текст представляет собой графическое изображение, он не переводится сканером в текстовые форматы, а это ограничивает возможность работы с текстом, в частности, делает невозможным его редактирование.

В связи с этим возникает задача машинного распознавания текстовой информации, хранящейся в виде изображения. Это очень сложная задача, как с теоретической, так и с практической точки зрения.

Человек, использует для этого весь комплекс знаний и опыта. Он определяет текст из совокупности сигналов органов чувств, выделяет каждый символ, выделяет характерные признаки символов и на основании своего опыта приходит к выводу о значении символа и всего текста в целом.

Целью данной работы является разработка программы, которая должна обрабатывать изображения, полученные в результате сканирования текстов, обучать классификатор, который будет классифицировать символы с входного изображения и, в конечном итоге, представлять текст в обычном формате. Для работы с изображениями была выбрана библиотека OpenCV, а в качестве алгоритма для классификации символов был выбран метод к ближайших соседей [2].

1. Библиотека OpenCV

Библиотека OpenCV [1] имеет несколько примитивных типов данных, а именно:

✓ `CvPoint` – структура из двух целочисленных переменных x и y , используемая для представления точек на плоскости. Кроме того, существуют структуры для представления точки в трёхмерном пространстве, а также в вещественных координатах.

✓ `CvRect` – тип данных, содержащий четыре поля: x , y , $width$ и $height$. Представляет прямоугольную область с координатами (x, y) верхнего левого угла, высотой и шириной $height$ и $width$, соответственно.

✓ `CvScalar` – множество из четырёх чисел типа `double`, используемое для представления одного, двух или трёх действительных чисел.

✓ `CvMat` – структура для представления матрицы. Понятие "матрица" в OpenCV более абстрактное, чем понятие матрицы из линейной алгебры. В частности элементами могут выступать экземпляры структур `CvPoint` и `SvScalar`.

✓ `IplImage` – этот тип данных, фактически, является типом `CvMat`, но содержит некоторые дополнительные возможности, позволяющие интерпретировать матрицу как изображение. Первоначально эта структура была определена как часть библиотеки для обработки изображений от Intel [2].

✓ `CvSeq` – тип данных для представления последовательностей точек в общем случае и для представления контуров изображения в частности.

✓ `CvMoments` – тип данных для представления моментов контура. Момент – это характеристика контура, вычисляемая суммированием по всем пикселям контура. В общем виде мы определим момент (p, q) следующим образом:

$$m_{p,q} = \sum_{i=1}^n l(x,y) * x^p * y^q . \quad (1)$$

Здесь $l(x,y) = 0$ для пикселей черного цвета и 1 – для белого; p и q означают степени, в которые возводятся соответствующий компонент, а n – количество пикселей в контуре.

Однако, такие моменты не самые лучшие исходные данные для сравнения контуров в большинстве случаев. Например, часто используются нормализованные моменты для сравнения контуров, имеющих похожие формы но совершенно разные размеры. Аналогично, простые пространственные моменты зависят от системы координат, и их значение будет другим, если контур будет повернут. OpenCV предоставляет функции для вычисления нормализованных моментов, а также hu-инвариантных моментов [3].

Центральные моменты – это тоже, что и обычные моменты, за исключением того, что значения x и y смещены на среднее значение:

$$\mu_{p,q} = \sum_{i=1}^n l(x,y) * (x - x_{avg})^p * (y - y_{avg})^q , \quad (2)$$

где $x_{avg} = \frac{m_{10}}{m_{00}}$, $y_{avg} = \frac{m_{01}}{m_{00}}$.

Нормализованные моменты – это отношение центральных моментов и соответствующей степени момента m_{00} :

$$\eta_{p,q} = \frac{2 * \mu_{p,q}}{m_{00}^{p+q} + 2} . \quad (3)$$

И, наконец, hu-инвариантные моменты, являющиеся линейными комбинациями центральных моментов. Идея состоит в том, что, вычисляя разные нормализованные центральные моменты, можно создать инвариантные функции, представляющие разные аспекты изображения, которые будут инвариантны по отношению к масштабированию, вращению и отражению.

Функция `cvGetHuMoments()` вычисляет hu-моменты, исходя из центральных моментов. Для полноты картины покажем фактические определения некоторых hu-моментов:

$$\begin{aligned} h_1 &= \eta_{20} + \eta_{02}, \\ h_2 &= (\eta_{20} + \eta_{02})^2 + 4\eta_{22}^2, \\ h_3 &= (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} - \eta_{03})^2, \\ h_4 &= (\eta_{30} + \eta_{12})^2 + (\eta_{21} + \eta_{03})^2. \end{aligned}$$

2. Метод ближайших соседей

Итак, мы можем получить вектор значений h_u -моментов для любого контура, а в данном случае для контуров изображений букв, и теперь необходимо проанализировать, каким образом мы можем по входному вектору значений h_u -моментов определить, какому символу он соответствует.

Математически задача классификации ставится следующим образом. Пусть X – множество описаний объектов, Y – множество классов. Существует некоторое отображение $X \rightarrow Y$, результаты которого известны только для подмножества обучающей выборки $X_m = \{(x_1, y_1), \dots, (x_m, y_m)\}$. Требуется построить алгоритм, реализующий это отображение для произвольного объекта $x \in X$.

Для начала рассмотрим результат работы алгоритма k ближайших соседей в случае, когда k равно 1. В этом случае алгоритм предсказывает, что неизвестный объект относится к классу, который мы обозначили плюсом, так как первый ближайший сосед неизвестного объекта является представителем класса "+". Теперь увеличим число k до двух и рассмотрим пару ближайших соседей неизвестного объекта. В этом случае алгоритм не сможет классифицировать объект, так как второй ближайший сосед относится к классу "-" и в этом случае классы "+" и "-" набирают одинаковое количество голосов (по одному). Теперь увеличим число k до пяти. Область с пятью ближайшими соседями неизвестного объекта изображена на рисунке. В этой области находятся два экземпляра класса "+" и три экземпляра класса "-", а это означает, что неизвестный объект будет классифицирован как экземпляр класса "-".

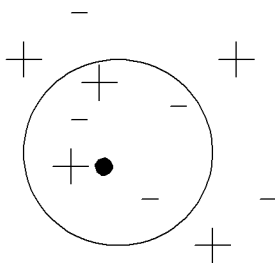


Рисунок. Пример работы алгоритма k ближайших соседей при различных значениях параметра k

Имеет смысл делать различия между ближайшими соседями при процессе классификации, чтобы результат классификации зависел от «расстояния» i -го ближайшего соседа до неизвестного объекта: чем ближе эта пара объектов, тем больше влияние на результат. Такая идея может быть достигнута путём введения набора весовых коэффициентов W , по

одному на каждого ближайшего соседа, который определяется мерой близости соседа к неизвестному объекту следующим образом:

$$W(x, p_i) = \frac{\exp(-D(x, p_i))}{\sum_{j=1}^k \exp(-D(x, p_j))}. \quad (4)$$

В формуле (4) использованы следующие обозначения: x – символ, который необходимо классифицировать; p_i – символ из обучающей выборки; D – «расстояние» между ними.

3. Структура приложения

Приложение содержит ряд классов, методы которых реализуют его функционал. Основными из них являются следующие классы.

Класс `Main`, содержащий точку входа (функцию `main`), а также абсолютный путь директории, в которой хранятся файлы обучающей выборки – изображения символов разных размеров и шрифтов.

Класс `ImageHelper`, обеспечивающий преобразование исходного изображения в черно-белое. Такое преобразование необходимо, поскольку, поиск контуров возможен только на бинарном изображении. Преобразование выполняется в несколько этапов:

1. функция `cvLoadImage()` возвращает указатель на трёхканальное изображение в формате RGB;
2. функция `cvCvtColor()` переводит изображение из трёхканального RGB в одноканальное изображение с оттенками серого;
3. функция `cvInRangeS()` преобразует это изображение в бинарное, используя следующее правило:

$$f(x) = \begin{cases} 1, & \text{left} < x < \text{right} \\ 0 - \text{иначе} \end{cases} \quad (5)$$

Здесь *left* и *right* границы интервала оттенков серого цвета, определяемые эмпирически.

Класс `SymbolsRecognizer` ищет контуры на получившемся бинарном изображении и вычисляет их *hu*-моменты. Получившийся вектор моментов подаётся на вход классификатору, который определит, какому классу-символу соответствует данный вектор.

Класс `KNNClassifier` – классификатор, непосредственно идентифицирующий символ. Вначале классификатор «просматривает» собственный ассоциативный массив `foundedEntities`, в котором он хранит значения "момент-класс" для таких векторов, для которых распознавание уже было произведено. Если вектора моментов нет в массиве уже классифицированных объектов, то происходит процесс распознавания с

помощью алгоритма ближайших соседей. Значение символа возвращается классу SymbolsRecognizer для записи в файл.

4. Результат работы приложения

Был проведён эксперимент по распознаванию символов с изображения текста «A beginning» при различных значениях k ближайших соседей. Результат эксперимента представлен в таблице.

Таблица

Результат работы алгоритма ближайших соседей

k	Процент верно распознанных символов
1	60%
2	73%
3	94%
4	96%
размер обучающей выборки	82%

Заключение

Рассмотрена задача распознавания текста с изображения. Было выяснено, что её можно разделить на последовательность подзадач, для решения каждой из которых существует несколько вариантов. Реализовано демонстрационное приложение, использующее библиотеку OpenCV и метод ближайших соседей для классификации образов.

Список литературы

1. Bradski G.R. Learning OpenCV / G.R. Bradski, A. Kaehler. – Sebastopol : O'Reilly, 2008. – 556 p.
2. Image processing library. Режим доступа: <http://www.comp.nus.edu.sg/cs4243/doc/ipl.pdf>
3. Hu M. Visual pattern recognition by moment invariants / M. Hu // IT (8). – 1962. – № 2. – P. 179–187.

СИСТЕМА ИНТЕГРАЦИИ WEB-САЙТОВ РЕКРУТМЕНТА

И.А. Петрушин

Студент

С.В. Сапегин

Доцент

Введение

Бурное технологическое развитие открыло миру огромное множество различных видов аппаратных и программных решений, которые в синтезе позволяют не только увеличить скорость обработки информации и сделать человека сверх коммуникабельным, но и оптимизировать затраты, связанные с эксплуатацией данных технологий.

На этом фоне появляются проблемы интеграции уже существующих продуктов, реализованных с помощью концептуально разных технологических средств и не имеющих общего, понятного всем протокола взаимодействия.

1. Анализ проблемной области

На данный момент существует много различных интернет ресурсов, на которых соискатель работы может выложить своё резюме в целях получить предложение о трудоустройстве от какой-либо заинтересованной в нём компании. Как правило, эти же ресурсы предоставляют аналогичную возможность публикации вакансий HR отдела компаний. Как правило, соискатели \ работодатели публикуют идентичные резюме \ вакансии, сразу на нескольких ресурсах, чтобы увеличить вероятность ответных предложений \ соглашений.

Таким образом, для управления своим резюме\вакансией пользователю необходимо заходить на каждый сайт, где был опубликован профиль и проделывать одну и ту же работу, в целях обновления, по каким-либо причинам не актуальной на текущий момент, информации. Самые простые примеры таких ситуаций:

а) Соискатель работы хочет добавить в своё резюме несколько новых навыков и изменить желаемую заработную плату;

б) Работодателю необходимо изменить требования, предъявляемые к какой-либо должности.

Количество времени, требуемое для обновления информации, прямо пропорционально количеству ресурсов. Очевидно, что это далеко не оптимальный с точки зрения производительности и не удобный, из-за монотонности действий, способ. Но другого выбора у пользователей нет. Данные интернет ресурсы не предоставляют никаких программных

интерфейсов для управления профилями и далеко не все поддерживают их импорт/экспорт.

2. Постановка задачи

Исходя из актуальности данной проблемы, было решено реализовать систему, предоставляющую пользователю:

- а) Программный интерфейс для взаимодействия с системой.
- б) Веб-интерфейс, посредством которого конечный пользователь приложения сможет воспользоваться функционалом системы.
- в) Правила формирования запроса к системе, включая формат представления данных экземпляра профиля-резюме.
- г) Возможность управлять пользовательским профилем (публикация/обновление резюме на указанных web-ресурсах).
- д) Надежный механизм хранения и синхронизации информации между системой и web-ресурсами, поддерживающий асинхронную модель передачи данных.

3. Анализ требований

В результате анализа требований были определены основные компоненты, из которых должна состоять система (рис. 1):

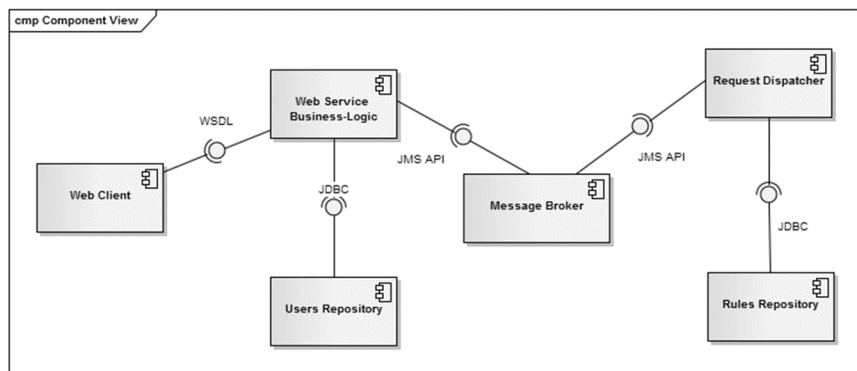


Рис. 1. Диаграмма Компонентов

- а) Web Client – пользовательский веб-интерфейс.
- б) WS Business-Logic – компонент, предоставляющий универсальный программный интерфейс к бизнес-логике системы.
- в) Message Broker – компонент системы, представляющий собой платформу для хранения, маршрутизации и трансформации запросов.

г) Request Dispatcher – компонент отвечающий за передачу обработанных системой запросов к сервисам назначения (web- сайтам рекрутмента).

д) Rules Repository – хранилище правил, в соответствии с которыми, система производит трансформацию запросов.

е) Users Repository – хранилище пользовательских учётных записей.

4. Анализ задачи

Для реализации поставленной задачи, в качестве основного архитектурного подхода, было решено использовать технологию Enterprise Service Bus (ESB), предназначенную для интеграции гетерогенных систем на уровне данных.

Суть подхода к интеграции на уровне данных заключается в следующем: приложения работают независимо друг от друга, каждое использует свой набор данных. В случае необходимости осуществляется обмен данными между приложениями. При этом, обмен данными осуществляется путем вызова сервисов или отправки/получения сообщений [1].

Основной принцип ESB — концентрация обмена сообщениями между различными системами через единую точку, в которой, при необходимости, обеспечивается транзакционный контроль, преобразование данных, сохранность сообщений. Все настройки обработки и передачи сообщений предполагаются также сконцентрированными в единой точке, и формируются в терминах служб, таким образом, при замене какой-либо информационной системы, подключённой к шине, нет необходимости в перенастройке остальных систем [2].

Важной особенностью сервисной шины является поддержка двух режимов передачи информации: синхронного и асинхронного.

В рамках поставленной задачи важно наличие именно второй характеристики, так её принцип обеспечивает надёжную передачу данных между приложениями, за счёт возможности использования сервисной шиной в качестве транспортного механизма технологии очередей сообщений (Message Queue). Таким образом, одно приложение посредством ESB может передать данные другому приложению без необходимости вызова процедуры получателя и определенно без ожидания результата. Отправитель не обязан знать, как найти получателя. Он может просто направить данные в ESB и быть уверенным, что они будут переданы.

Помимо возможности асинхронной передачи данных технология MQ обладает следующими важными свойствами [3]:

а) Отказоустойчивость — очереди сообщений позволяют отделить процессы друг от друга, так что если процесс, который обрабатывает сообщения из очереди, падает, то сообщения могут быть добавлены в очередь на обработку позднее, когда система восстановится.

б) Гарантированная доставка — использование очереди сообщений гарантирует, что сообщение будет доставлено и обработано в любом случае (пока есть хотя бы один обработчик).

в) Гарантированный порядок доставки — большая часть систем очередей сообщений способны обеспечить гарантии того, что данные будут обрабатываться в определённом порядке (чаще всего в том порядке в котором они поступили).

5. Реализация

Для реализации данной системы были выбраны технологии из стека Java, так как кроссплатформенность и масштабируемость разрабатываемого решения играет очень важную роль, в его развертывании в реальном окружении и в последующей поддержке.

За основу, реализующую концепцию ESB, было решено использовать open-source интеграционную платформу Apache ServiceMix. Данная технология распространяется под лицензией apache Licence 2.0. Это означает, что данное решение может быть использовано как в академических проектах, так и в коммерческих (единственное требование – прикрепить к дистрибутиву разработанного продукта файл лицензии Apache).

- Apache ServiceMix состоит из пяти компонентов:
- ApacheMQ – messaging платформа.
- Apache Camel – Фреймворк для конфигурации интеграции систем.
- Apache CXF – платформа для создания и развертывания веб-сервисов (SOAP, XML/HTTP, RESTful HTTP)
- Apache Karaf – легковесный OSGi – контейнер.
- Apache ODE – Фреймворк для обработки бизнес процессов в соответствии с WS-BPEL стандартом [4].

Технология Apache Camel была использована, как основной механизм трансформации и маршрутизации запросов к сторонним Internet ресурсам.

Для реализации общего формата представления запроса к системе были использованы технологии XML XSD.

Компонент “WS Business-Logic” было решено реализовать в виде Web-сервиса, а в качестве технологии разработки была выбрана платформа Apache CXF.

Apache CXF – реализация спецификации JAX-WS , что означает наличие возможности создания Web-сервиса на основе существующего кода Java или же генерации код Java на основе WSDL-описания для использования или реализации сервиса. Очевидно, что такой подход делает разработку сервисов удобной и быстрой.

Важным преимуществом выбранной технологии перед другими аналогичными технологиями, является то, что Apache CXF хорошо подходит для решения задач, в которых необходимо SOAP-компонент встроить в уже существующую платформу.

Компоненты “Users Repository” и “Users Repository” были реализованы в виде схем базы данных. В качестве платформы СУБД было решено использовать HSQLDB.

HSQLDB полностью написана на Java. Может использоваться и как отдельный сервер с поддержкой сетевых соединений по JDBC (Java Data Base Connectivity), и в виде библиотеки для использования непосредственно в коде программы.

Для реализации графического пользовательского интерфейса была выбрана технология Java Server Faces (JSF). За счёт следующих особенностей она существенно упрощает разработку:

- JSF предлагает компоненто-центрический, независимый от клиента подход к разработке пользовательских интерфейсов для Web, таким образом, увеличивая продуктивность разработчика и простой в использовании;

- Автоматически управляет состоянием пользовательского интерфейса между несколькими запросами и несколькими клиентами простым и ненавязчивым способом;

- Описывает стандартный набор архитектурных шаблонов для Web-приложений.

- JSF также предоставляет многочисленные невидимые компоненты, которые занимаются проверкой данных и выполняют задачи преобразования.

На рис. 2 изображена стартовая страница приложения, содержащая форму авторизации.

Авторизовавшийся в системе, пользователь попадает на главную страницу приложения, изображенную на рис. 3.

Спроектированная на основе выше описанных технологий архитектура представлена на диаграмме развертывания (рис. 4).

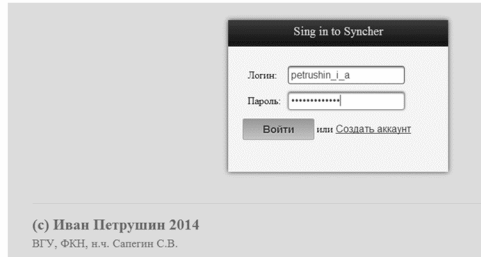


Рис. 2. Стартовая страница приложения

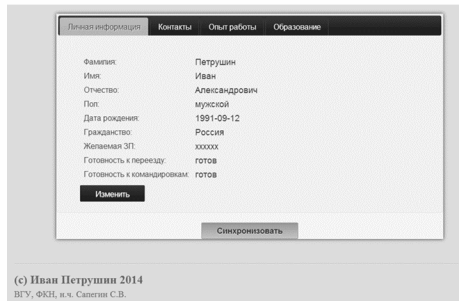


Рис. 3. Главная страница. Форма личной информации

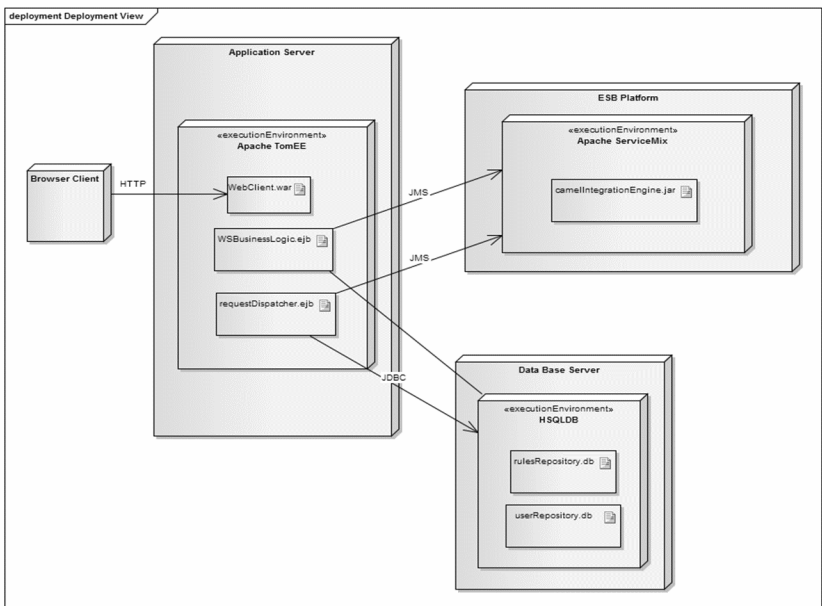


Рис. 4. Диаграмма компонентов и развёртывания

Заключение

В результате проделанной работы была разработана система, не имеющая аналогов на текущий момент, предоставляющая клиенту:

- а) Программный-WSDL и пользовательский Web-интерфейс для взаимодействия с системой.
- б) Правила формирования запроса к системе, включая формат представления данных экземпляра профиля-резюме.
- в) Возможности управления профилем резюме на основе реализованной интеграцией с сервисами “HeadHunter.ru” и “Monster.com”.
- г) Надежный механизм хранения и синхронизации информации между системой и web-ресурсами, поддерживающий асинхронную модель передачи данных.

Список литературы

- 1 Самольсов П. Три подхода к интеграции информационных систем. – Режим доступа: <http://samolisov.blogspot.ru/2012/03/blog-post.html>
- 2 Шаппел Д. ESB — Сервисная Шина Предприятия / Д. Шаппел. – СПб: БХВ-Петербург, 2008. – 370 с.
- 3 Свистунов А. Лекция 13: Использование JMS. – Режим доступа: <http://www.intuit.ru/studies/courses/633/489/lecture/11092?page=2>
- 4 Apache ServiceMix. Официальная документация. – Режим доступа: <http://servicemix.apache.org/documentation.html>

МАТЕМАТИЧЕСКОЕ И ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ДЛЯ ОБОСНОВАНИЯ СТРУКТУРЫ ИНФОРМАЦИОННОЙ СИСТЕМЫ С ИСПОЛЬЗОВАНИЕМ МЕТОДОВ СИСТЕМНОГО АНАЛИЗА

Н.Ю. Полухина

Студент

А.А. Сирота

Профессор

Введение

Люди постоянно принимают решения. Так было в прошлом, настоящем и, наверное, так будет всегда. Принимая сложные решения, мы пользуемся нашей памятью и знаниями, независимо от того, насколько связными и полными они являются. Результаты наших решений зависят от наших суждений и иногда от суждений других людей, мы часто

испытываем затруднения, решая, что имеет больший приоритет и является более важным для нас.

Морфологические методы являются мощным аппаратом исследования в различных областях знаний. Сущность морфологических методов заключается в том, что в результате морфологического анализа мы определяем пространство поиска, называемое морфологическим множеством, которое обязательно должно включать в себя искомое решение, а затем сужаем это пространство, осуществляя поиск этого решения, которое является элементом морфологического множества.

Структуру решения можно представить иерархией, включающей цель, критерии действующих лиц с их целями, людей, на которых влияет рассматриваемое решение и альтернативные варианты решения.

Метод анализа иерархий (МАИ) позволяет найти лучший из альтернативных вариантов или распределить ресурсы между альтернативами пропорционально их приоритетам. Метод состоит в декомпозиции проблемы на все более простые составляющие части и дальнейшей обработке последовательности суждений лица, принимающего решение (ЛПР), попарным сравнениям. В результате может быть выражена относительная степень (интенсивность) взаимодействия элементов в иерархии. Эти суждения затем выражаются численно. Метод анализа иерархии включает процедуры синтеза множественных суждений, получения приоритетности критериев и нахождения альтернативных решений [1].

Но в реальной жизни анализируемые варианты решений всегда имеют достоинства и недостатки. Принимая ответственные решения, как правило, требуется оценить выгоды и издержки для всех рассматриваемых альтернатив. В таких случаях полезно проводить функционально-стоимостный анализ, который позволяет выработать основные принципы распределения ресурсов, в то время как МАИ используется для структурирования проблемы распределения и изменения факторов, важных для решения.

Целью данной работы является разработка математического и программного обеспечения для обоснования структуры информационной системы с использованием методов системного анализа (на примере системы защиты информации).

1. Теоретический анализ

Рассмотрим конкретный пример генерации морфологического множества альтернативных вариантов системы защиты информации (СЗИ) для размещенной в помещениях одного здания корпоративной сети, подробно описанный в работе [2]. Объект информатизации (ОИ) или защищаемая ИС в [2] рассматривается как интеграция рабочих станций,

серверов, межсетевых мостов, элементов соединения нескольких сегментов сети и каналов связи. На основе анализа общих принципов построения СЗИ, возможных каналов несанкционированного доступа к информационной среде персонального компьютера (ПК), сегмента и объекта информатизации, состоящего из нескольких сегментов, предлагается трехрубежная модель СЗИ [2]. В ней первый рубеж – рубеж защиты периметра ОИ включает в себя комплекс функциональных подсистем защиты от внешних специальных деструктивных программно-технических воздействий злоумышленника и потенциальных воздействий удаленного пользователя; второй рубеж – рубеж защиты сегмента включает в себя комплекс функциональных подсистем защиты сегмента ОИ от внутренних деструктивных воздействий нарушителя (пользователей данного или соседнего сегмента), удаленного пользователя и злоумышленника; третий рубеж – рубеж защиты ПК и сервера реализует комплекс функциональных подсистем, обеспечивающих защиту информационной среды отдельного персонального компьютера и серверов сети.

Модель каждого рубежа защиты является многоуровневой, и формируется на основе морфологических таблиц, заполняемых с учетом уровней ограничения доступа к хранимой или обрабатываемой на объекте информации. Предварительно анализируются и обобщаются требования к функциональным подсистемам каждого рубежа СЗИ с использованием имеющихся руководящих элементов и внутренней нормативной документации. Списки требований к каждому комплексу выделенных рубежей составляются в различных вариантах для случаев обработки информации с различными уровнями ограничения доступа.

Альтернативные реализации (элементарные альтернативы) средств защиты выбираются с учетом этих требований. Таким образом, для каждого рубежа защиты создаются в общем случае несколько морфологических таблиц (но числу введенных уровней защищенности), в каждой из которых отражаются: перечень функциональных подсистем рубежа; требования к каждой подсистеме в зависимости от устанавливаемого уровня защищенности; набор элементарных альтернатив для каждой подсистемы. Для каждого рубежа защиты целостные альтернативные варианты генерируются с учетом совместимости программно-аппаратных средств.

Пример заполнения морфологической таблицы для выбора комплекса средств защиты периметра ОИ, детально рассмотренный в [2], представлен на рис. 1-4.

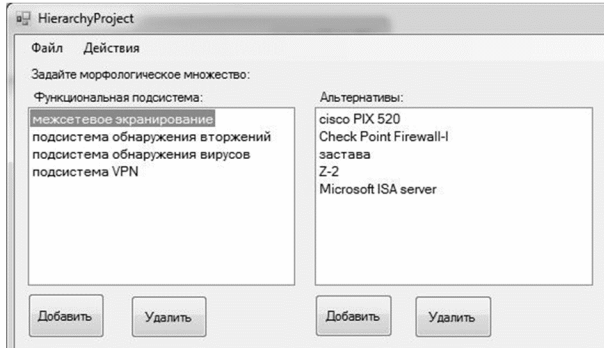


Рис. 1. Задание морфологической таблицы

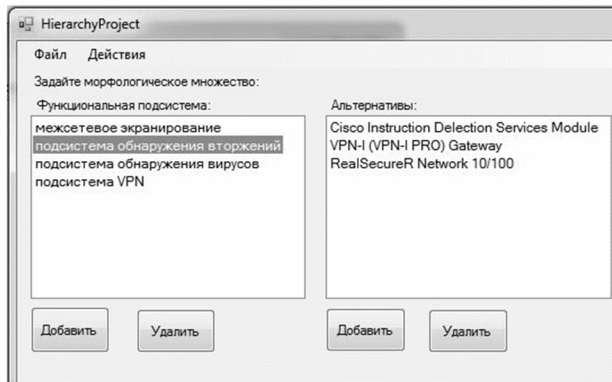


Рис. 2. Задание морфологической таблицы

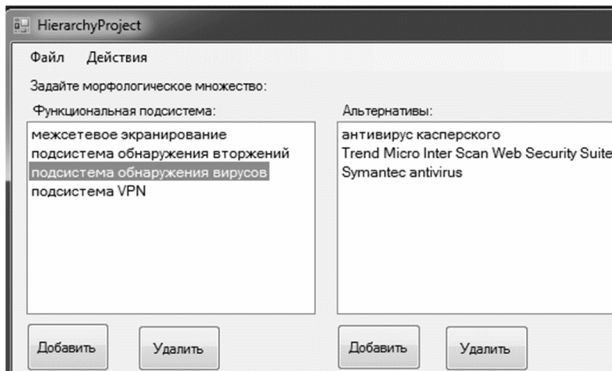


Рис. 3. Задание морфологической таблицы

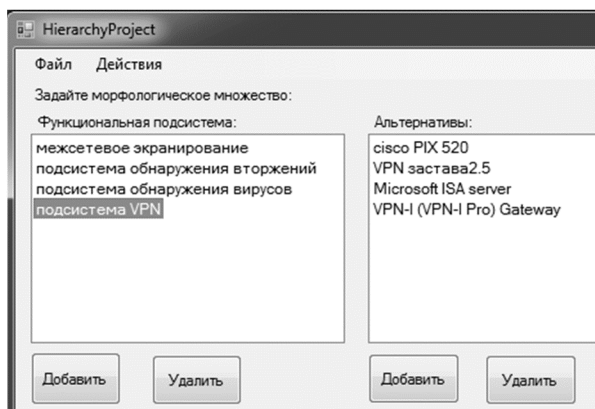


Рис. 4. Задание морфологической таблицы

Здесь выделены следующие функциональные подсистемы: подсистема межсетевого экранирования; подсистема обнаружения вторжений; подсистема обнаружения вирусов; подсистема, реализующая функции VPN (Virtual Private Network – виртуальная частная защищенная сеть – обеспечение безопасности и конфиденциальности данных, передаваемых по компьютерным сетям).

Требования для данной морфологической матрицы формировались исходя из первого (высшего) класса защищенности.

2. Анализ совместимости элементарных альтернатив

После заполнения подобным образом морфологических таблиц, определяющих исходное множество альтернативных вариантов, проводится его усечение исходя из соображений совместимости программно-аппаратных средств. Для этого заполняются вспомогательные матрицы совместимости элементарных альтернатив путем определения функций совместимости. Для этой функции определены два значения: 1 – если элементарные альтернативы совместимы, 0 – если элементарные альтернативы не совместимы. Объекты, принадлежащие одной строке морфологической таблицы, сравнению не подлежат. Пример вспомогательной матрицы совместимости для получения целостных вариантов СЗИ периметра ОИ приведен рис. 5.

Следует отметить, что при наличии некоторого комплексного решения, объединяющего реализацию альтернатив различных функциональных подсистем, составляющие этого решения вносятся в качестве элементарных альтернатив во все эти функциональные альтернативы, а стоимость интеграционного решения учитывается один раз.

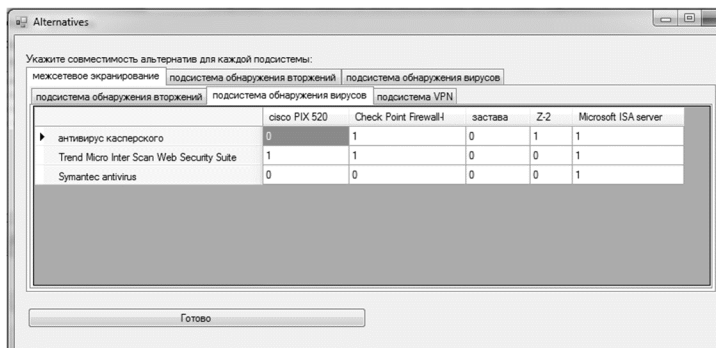


Рис. 5. Заполнение матрицы совместимости

Таким образом, выбор элементарных альтернатив из каждой строки морфологической матрицы (но одной из каждой строки) для генерации целостного альтернативного варианта системы осуществляется только из совместимых программ-продуктов.

3. Функционально-стоимостный анализ

Прежде чем перейти к подобным и весьма затратным исследованиям, можно выполнить еще один этап анализа, направленного на сужение подмножества вариантов системы. Данный этап предполагает проведение функционально-стоимостного анализа и прогнозирование возможной эффективности систем [3]. При этом предполагается, что исследуемая система улучшается по нескольким функциям и имеется более одной альтернативы по реализации каждой функции.

Для проведения предварительной оценки и отбора допустимых вариантов реализации подсистем и системы в целом проводится экспертный анализ выгод и издержек, определяемых по следующим формулам:

$$U^{(k)} = \sum_{i=1}^L U_{m_k}^{(i)}, V^{(k)} = \sum_{i=1}^L V_{m_k}^{(i)}, m_k \in \{1, \dots, L_i\},$$

где $U^{(k)}$, $V^{(k)}$ - выгоды и издержки от реализации k -го варианта системы; $U_{m_k}^{(i)}$, $V_{m_k}^{(i)}$ - выгоды и издержки от реализации элементарных альтернатив.

Определение этих показателей проводится методом анализа иерархий. Для оценки выгод от реализации каждой функциональной подсистемы составляется уникальная иерархия. В ее вершине определяется достигаемая эффективность – выгода.

На втором уровне определяются критерии (рис. 6), отражающие аспекты этой эффективности.

Выгоды от подсистемы	централизованное администрирование	система обнаружения вторжений	число активных элементов	наличие трансляции сетевых адресов	прозрачность для приложений	число протоколов	сертификат	уровни модели OSI	Вектор приоритетов
централизованное администрирование	1	7	8	6	4	7	3	3	0,40319534...
система обнаружения вторжений	0,1428571...	1	0,1666666...	0,1428571...	0,11111111...	5	3	7	0,06521882...
число активных элементов	0,125	6	1	0,2	0,33333333...	2	7	9	0,12287558...
наличие трансляции сетевых адресов	0,1666666...	7	5	1	0,16666666...	0,1428571...	4	0,125	0,06994436...
прозрачность для приложений	0,125	9	3	6	1	7	0,333333...	0,14285...	0,13515737...
число протоколов	0,1428571...	0,2	0,5	7	0,142857142...	1	4	7	0,08705736...
сертификат	0,3333333...	0,3333333...	0,1428571...	0,25	3	0,25	1	0,33333...	0,04112590...
уровни модели OSI	0,3333333...	0,1428571...	0,11111111...	8	7	0,1428571...	3	1	0,07542524...

L max: 14,4230187367459 ИС: 0,917574105249409 ОС: 0,632809727758213

Вычислить

Далее

Рис. 6. Второй уровень иерархии

На третьем уровне располагаются элементарные альтернативы, которые сравниваются по отношению к критериям, т.е. всего на этом уровне 8 матриц. На рис. 7 представлен пример одной из матриц третьего уровня.

централизованное администрирование	система обнаружения вторжений	число активных элементов				
централизованное администрирование	cisco PIX 520	Check Point Firewall	застава	Z-2	Microsoft ISA server	Вектор приоритетов
централизованное администрирование	1	7	8	9	3	0,5485584...
система обнаружения вторжений	0,142857...	1	0,5	0,11111...	4	0,0636283...
число активных элементов	0,125	2	1	0,16666...	0,25	0,0509164...
	0,111111...	9	6	1	7	0,2678936...
	0,333333...	0,25	4	0,14285...	1	0,0690031...

L max: 7,0009331921066 ИС: 0,500233298026649 ОС: 0,403413950021491

Рис. 7. Матрица третьего уровня иерархии

Необходимость использования уникальной иерархии выгод для элементарных альтернатив каждой строки морфологической таблицы определяется особенностями применения и описания различных функциональных подсистем. Напротив, иерархия для издержек для первых двух уровней может иметь одинаковую структуру.

В разбираемом примере [2] для первого рубежа системы защиты информации и подсистемы межсетевое экранирования иерархия выгод приведены на рис. 8.

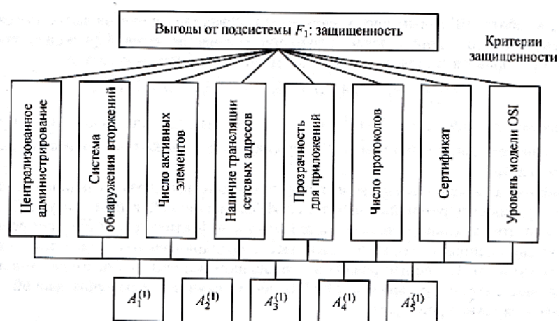


Рис. 8. Схема иерархии выгоды

При составлении иерархии издержек критерии второго уровня обычно группируются в три группы: экономические, связанные с затратами на реализацию и эксплуатацию альтернативных вариантов подсистем; социальные, связанные как с внутренними (сокращение персонала, переобучение и т. п.), так и с внешними последствиями; функциональные, определяющие ограничения, возникающие при реализации альтернативного варианта функциональной подсистемы (рис. 9).

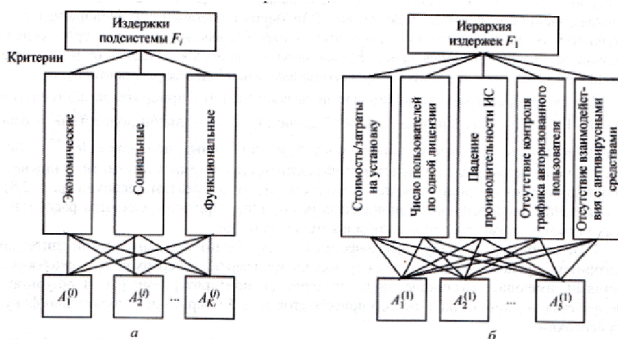


Рис. 9. Общий вид иерархии издержек (а); пример ее реализации для подсистемы межсетевого экранирования (б)

В иерархии выгод и издержек проводятся оценки глобальных приоритетов альтернатив $U_{m_k}^{(i)}, V_{m_k}^{(i)}, i = \overline{1, L}$ по каждой из функциональных подсистем либо на основе традиционного способа проведения подобных оценок в МАИ, либо модифицированным методом стандартов и копирования (рис. 10).

FormHierarchy

	0.4031953414...	0.0652188235...	0.1228755845...	0.0699443625...	0.13515737530	0.0870573668...	0.04112590441	0.0754252412...	Глобальные приоритеты
▶ cisco PIX 520	0.548558451...	0.162030358...	0.531540242...	0.40718507...	0.157336542...	0.560176023...	0.555402479...	0.201632003...	0.433619475...
Check Point Firewall	0.063628317...	0.425876428...	0.216798130...	0.31337518...	0.091540093...	0.187780353...	0.082092446...	0.163185275...	0.146392222...
застава	0.050916472...	0.084637241...	0.039282719...	0.11218214...	0.071252111...	0.060502268...	0.050636023...	0.104300577...	0.063569386...
Z-2	0.267893644...	0.157760402...	0.069345483...	0.04609990...	0.286898123...	0.095109938...	0.146338197...	0.041326318...	0.186239468...
Microsoft ISA server	0.069003114...	0.169695569...	0.143033424...	0.12115770...	0.392973129...	0.096431416...	0.165530851...	0.489555825...	0.170179447...

Назад Готово

Рис. 10. Глобальные приоритеты

После этого осуществляется выбор допустимых и предлагаемых для последующего исследования вариантов подсистем, удовлетворяющих, например, ограничениям вида (рис. 11):

$$U^{(k)} \geq U_0, V^{(k)} \leq V_0,$$

где U_0, V_0 – пороговые значения выгод и издержек.

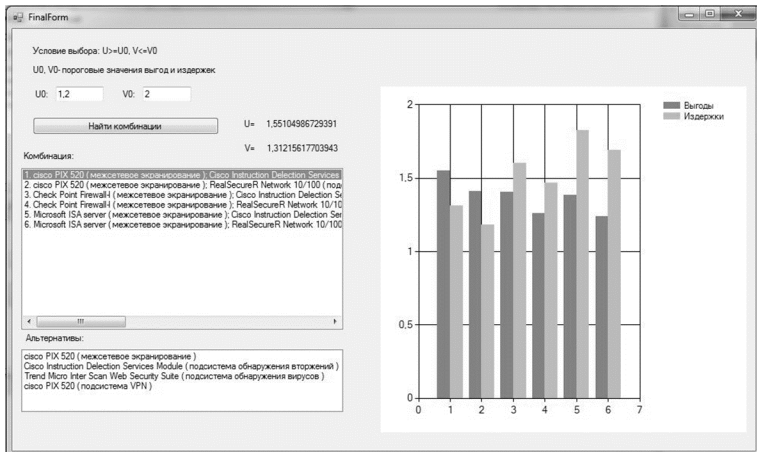


Рис. 11. Вывод допустимых вариантов

Функциональные подсистемы, из которых состоит некоторая функциональная система, могут вносить различный вклад в ее эффективность. Для таких ситуаций, помимо оценки относительного вклада альтернатив в эффективность и новизну решений, необходимо оценивать относительный вклад функциональных подсистем, как это предложено в [3].

Заключение

В результате выполнения работы реализовано математическое и программное обеспечение для обоснования структуры информационной системы с использованием методов системного анализа.

Реализован морфологический метод, позволяющий систематизировать достаточно большой объем знаний о морфологии систем в компактном виде, метод анализа иерархии, который используется в качестве вычислений выгод и издержек, анализ совместимости, позволяющий провести усечение множества альтернатив исходя из соображений совместимости программно-аппаратных средств и функционально-стоимостный анализ, позволяющий пользователю определить наилучший вариант.

Список литературы

1. Саати Т.Л. Принятие решений при зависимостях и обратных связях / Т.Л. Саати. – М. : Издательство ЛКИ, 2008. – 360 с.
2. Машкина И.В. Проектирование системы защиты информации объекта информатизации / И. В. Машкина, В. И. Васильев, Е. А. Рахимов // Информационные технологии. – 2006. – №10. – С. 17–26.
3. Андрейчиков А.В. Анализ, синтез, планирование решений в экономике / А.В. Андрейчиков. – М. : Финансы и статистика, 2004. – 464 с.

МОБИЛЬНЫЕ БЕСПРОВОДНЫЕ СЕТИ ЯЧЕЙСТОЙ ТОПОЛОГИИ

А.И. Рагуцкий
Студент

Ю.Б. Нечаев
Профессор

Введение

Беспроводные сети являются одним из самых перспективных направлений развития современных телекоммуникационных технологий.

Перспективы их использования связаны, во-первых, с заменой кабельной инфраструктуры на радиоэфир; во-вторых, с новыми возможностями коммуникаций между различными устройствами. При этом наряду с построением централизованных сетей, интерес представляет использование ЭЛЕМЕНТОВ децентрализации, которые присутствуют в ячеистых (mesh) сетях.

1. Ячеистая сеть

Широкое распространение беспроводных технологий и появление большого количества разнообразных мобильных устройств привело к необходимости создания новой технологии, которая обеспечивала бы надежную связь в условиях постоянно изменяющейся архитектуры сети.

В рамках IEEE по инициативе Intel, Cisco и ряда других компаний была сформирована рабочая группа по стандартизации новой технологии беспроводных локальных сетей, базирующихся на архитектуре 802.11. Технология получила название MESH. Остановимся более подробно на особенностях mesh-технологии [1].

Беспроводные mesh-сети являются динамически самоорганизующимися и самоконфигурируемыми, с узлами в сети автоматически порождающими ad-hoc сеть и поддерживающими mesh-связность. Mesh-сеть основана на децентрализованной схеме организации сети и включает в себя два типа узлов: mesh-роутеры и mesh-клиенты. Каждый узел по радиоканалу соединен со всеми соседями в зоне видимости. Сеть на основе этих узлов самоорганизуется и способна восстанавливаться при выходе из строя некоторых узлов. Mesh-роутеры имеют минимальную мобильность и формируют mesh-магистраль для mesh-клиентов. Таким образом, хотя mesh-клиенты также могут работать как роутеры в mesh-сетях, аппаратная платформа и программное обеспечение для них может быть более простыми, чем для mesh-роутеров. Например, коммуникационные протоколы для mesh-клиентов могут быть «легкими», функций шлюза и моста может не быть, может быть только один беспроводный интерфейс и т.д. Архитектуру mesh-сетей можно разделить на три типа: инфраструктура mesh, клиентская mesh-сеть и гибридная mesh-сеть.

В первой архитектуре mesh-роутеры формируют самоконфигурируемые, самовосстанавливающие mesh-связи между собой, а также создают инфраструктуру для клиентов. Mesh-роутеры с функциями шлюза могут подсоединяться к сети Internet. Данная архитектура может быть построена с использованием различных типов радио-технологий, что позволяет обеспечить интеграцию с

существующими беспроводными сетями при помощи функций шлюза\моста mesh-роутеров.

В клиентской mesh-сети узлы образуют текущую сеть, которая выполняет функции маршрутизации и конфигурации. Поэтому mesh-роутер не требуется для этого типа сети. Клиентские mesh-сети обычно формируются с использованием одного типа радиоустройств. Таким образом, это, фактически, то же самое, что и обычные ad-hoc сети. Однако требования к конечным устройствам повышаются по сравнению с инфраструктурой mesh, так как в архитектуре клиентских сетей конечные устройства должны обеспечивать дополнительные функции, такие как маршрутизация и самоконфигурация.

В сравнении с ad-hoc, Mesh-сеть имеет следующие особенности, получившие в стандарте название сервисов:

- *сервис выбора оптимального пути.* Представляет собой протокол HWMP на MAC уровне вместо MANET в ad-hoc. HWMP (Hybrid Wireless Mesh Protocol) соединяет в себе два алгоритма выбора оптимального пути: проактивный и реактивный. Проактивный выбор подразумевает построение пути заранее в процессе работы, реактивный – построение по запросу.

- *сервис обзора/поиска меш сетей.* Сервис подразумевает два вида обзора: активный и пассивный. При пассивном поиске, происходит анализ beacon-пакетов с поиском в них поля MeshID, в котором содержится название сети, например "MyMesh". В Ad-hoc сети данное поле называется BSSID.

- *beaconing и синхронизация.* В Mesh-сети все узлы в специально отведенный интервал отправляют beacon-пакеты, сообщая о своем существовании. Beacon-пакеты в любом случае всегда отправляются всеми узлами. В тоже время в ad-hoc сети происходит "соревнование" на то, кто первый отправит beacon пакет, по которому остальные узлы синхронизируют свое время и уже не будут отправлять свои beacon-пакеты в этом временном интервале.

- *сервис управления энергосбережением.* В стандарте IEEE 802.11s для узлов введено понятие энергосбережения. Узел имеет три состояния: активный, легкий сон и глубокий сон. В активном состоянии он ничем не отличается от обычного ad-hoc узла. При легком сне узел должен предоставлять сервис передачи кадров и прослушивать все beacon-пакеты от своих соседей. В глубоком сне узел имеет выбор и может не прослушивать beacon-пакеты соседей.

В обычной беспроводной LAN многие пользователи получают доступ к сети через прямую связь с точкой доступа (ТД); это сеть с одним центром. В сети со многими центрами любое устройство с радиосвязью

может служить маршрутизатором или ТД. Если ближайшая ТД перегружена, данные перенаправляются к другому узлу с более низкой загрузкой. Данные продолжают “перескакивать” от одного узла к другому до тех пор, пока они не достигнут точки своего окончательного назначения.

Ячеистая сеть более устойчива, чем сеть с единственным центром, поскольку она не зависит от функционирования каждого узла. В сети с одним центром, когда единственная ТД выходит из строя, с ней перестает функционировать и вся сеть. В архитектуре ячеистой сети, даже если ТД отказывает или возникает высокий уровень интерференции, сеть продолжает работать – данные будут просто перенаправляться по другому маршруту.

Еще один путь достичь устойчивости сети – использование многочисленных маршрутов для доставки данных. Хороший пример – электронная почта, которая делится на пакеты данных, пересылаемые через Интернет по многим маршрутам, а затем собираемые в первоначальное письмо, приходящее к адресату. Использование многочисленных маршрутов для доставки данных увеличивает эффективную полосу пропускания.

Физика беспроводных коммуникаций говорит о том, что полоса пропускания возрастает при уменьшении радиуса действия, поскольку эффект от интерференции и других факторов, вносящих вклад в потерю данных, возрастает с ростом расстояния. Один из способов получить большую полосу пропускания в сети – это разбить длинный путь на большое количество коротких участков и передавать данные через них. Именно это и делает ячеистая сеть.

Кроме того, для передачи данных на короткие расстояния требуется меньшая мощность, поэтому ячеистые сети могут работать в более широкой полосе, несмотря на ограничения американской Федеральной комиссии по коммуникациям (ФСС) на максимум передаваемой мощности.

Возможность эффективного использования пространства – еще одно преимущество ячеистой сети над сетью с одним центром. Как уже отмечалось, в сети с одним центром часто случается ситуация, когда много устройств обращаются к одной и той же ТД. Если несколько устройств стремятся получить доступ в сеть в одно и то же время, возникает перегрузка сети и передача трафика замедляется. В ячеистой сети наоборот многие устройства могут работать в сети в одно и то же время через разные узлы, не влияя существенно на работу всей сети.

Более короткий диапазон действия в ячеистой сети ограничивает интерференцию и позволяет одновременно реализовывать пространственно разделенные сеансы передачи данных.

Mesh сети — это вполне осмысленный следующий шаг в развитии беспроводных сетей, в mesh сети вы «сам себе провайдер», вас нельзя отключить от этой сети, с вами нельзя разорвать договор о пользовании интернетом, вас нельзя подслушивать спец. оборудованием.

2. Моделирование и анализ

Целью моей работы являлось моделирование и анализ протоколов AODV и DSDV [2].

Одними из наиболее часто используемых протоколов являются AODV, DSDV. AODV предполагает построение таблиц маршрутизации на каждом узле сети для минимизации времени передачи информации между узлами. Маршруты поддерживаются только в том случае, если они действительно активны. Избежать использования несуществующих маршрутов позволяет наличие в таблицах маршрутизации узлов специального таймера. По истечении указанного времени, при котором маршрут не используется, запись о нем удаляется. При начальном построении маршрутов и для реакции на изменения топологии сети, приводящие к разрывам маршрутов, AODV использует следующие типы пакетов: запрос на построение маршрута (RREQ), ответ на запрос построения маршрута (RREP), ошибка маршрута (RERR).

DSDV (Highly Dynamic Destination-Sequenced Distance-Vector Routing) – проактивный протокол маршрутизации, основанный на алгоритме Беллмана-Форда. Данный протокол реализует периодический обмен таблицами маршрутизации между соседними узлами. В качестве метрики маршрута DSDV использует число ретрансляций на пути от отправителя к получателю. Каждый беспроводной узел хранит таблицу маршрутизации, которая содержит все возможные узлы назначения, число ретрансляций до каждого из них и номер последовательности, привязанный к каждому узлу назначения. Номер последовательности позволяет идентифицировать более новые маршруты до получателя и избегать тем самым формирования маршрутных петель при построении маршрута.

В качестве средства моделирования я использовал сетевой симулятор Network Simulator 2 [3]. Несомненным достоинством данного симулятора для случая беспроводной сети является его возможность при помощи генератора сценариев определить передвижение узлов. Для сравнения протоколов маршрутизации в пакете ns2 была построена модель беспроводной сети, состоящей из 10 узлов, каждый из которых сочетает в себе функции маршрутизатора и оконечного устройства. Сеть располагалась на территории 200x200 м. Все узлы сети в начальный момент времени стационарны, а затем с заданного момента времени начинают движение по заданным маршрутам со случайно выбранной

скоростью в диапазоне от 0 до 20 м/с, использовались антенны с круговой диаграммой направленности в горизонтальной плоскости и высотой 0,5 м. В качестве типа трафика использовался CBR (Constant Bit Rate), Pareto, Expo и Real Audio с размером пакета 512 байт. Мощность сигнала на приеме рассчитывалась с помощью двулучевой модели распространения сигнала с отражением одного из лучей от поверхности. Вариант топологии рассматриваемой сети изображен на рис. 1

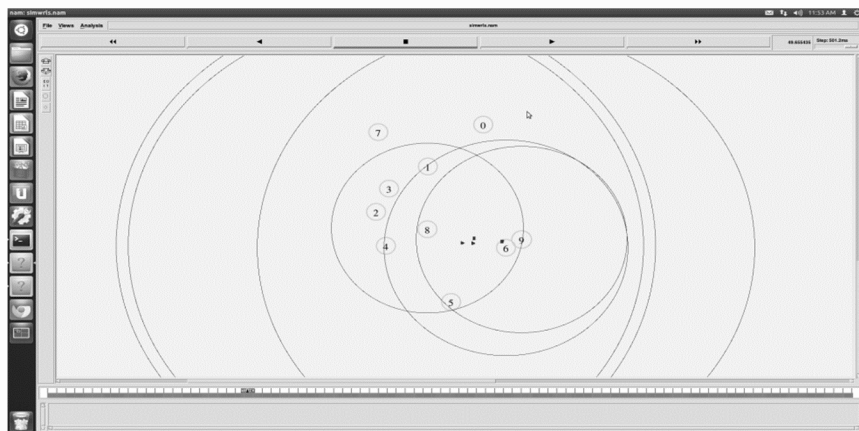


Рис. 1. Модель ячеистой топологии в ps2

В качестве примера на рис. 2. представлены результаты моделирования с использованием трафика CBR.

Параметры	Протокол	AODV	DSDV
Объем ретранслированной пользовательской информации		2484972	1601892
Объем принятой пользовательской информации		5308828	4445984
Объем служебной информации		19400	30120

Рис. 2. Результат моделирования сети с трафиком CBR

Из полученных результатов следует, что в сети с преобладанием трафика CBR лучше использовать протокол AODV, так как он оказывает меньшую нагрузку в сети и обеспечивает наименьшую потерю пользовательской информации, чего не скажешь про протокол DSDV. Однако, на коммутация сети с использованием протокола AODV уходит больше времени, и, как правило, строятся не самые оптимальные маршруты.

Заключение

При анализе протоколов, в конкретном случае, нельзя определить какой протокол является лучшим для построения мобильной беспроводной сети с ячеистой топологией. Необходимо исходить из имеющегося преобладания определенного вида трафика в такой сети.

Что касается самой сети, предлагаемая сверхвысокоскоростная широкополосная беспроводная mesh-сеть обладает рядом существенных преимуществ перед существующими технологиями. К ним можно отнести высокую скорость развертывания, отсутствие необходимости частотного планирования, возможность достичь низкой стоимости оборудования, гибкость и адаптивность архитектуры. Инвестиции в развертывание и обслуживание такой сети существенно ниже, чем для современных сетевых технологий, обеспечивающих аналогичные скорости передачи. Сеть можно развернуть не только быстро, но и на неподготовленных площадках, включая автомобили. Поскольку все узлы являются ретрансляторами, сеть будет успешно функционировать на любой пересеченной местности, включая плотную городскую застройку. Подобная сеть позволит передавать весь современный контент, доступный сегодня только в проводных сетях.

Список литературы

1. Маршрутизация в беспроводных мобильных Ad hoc-сетях / В.М. Винокуров и др. // Доклады ТУСУРа. – 2010. – № 2 (22). – Ч. 1. – С. 288–292.
2. Вишневский В.М. Mesh-сети стандарта IEEE 802.11s: протоколы маршрутизации / В.М. Вишневский и др. // ISSN 1992-4178. Первая миля. – 2009. – № 1. – С. 16–21.
3. Галкин А.М. Пакет моделирования NS-2 : учебное пособие / А.М. Галкин, Е.А. Кучерявый, Д.А. Молчанов. – СПб. : СПбГУТ, 2007. – 77 с.

ИСПОЛЬЗОВАНИЕ КАНОНИЧЕСКИХ УРАВНЕНИЙ В ЗАДАЧЕ ОБ АФФИННОЙ ОДНОРОДНОСТИ

Е.В. Ремизова
Студент

А.В. Лобода
Профессор

1. Основные понятия и подходы к задаче

Основываясь на работах Лободы А.В., Ходарева А.С., мы рассматриваем вещественно-аналитическую поверхность $M \subset \mathbb{C}^3$, заданную вблизи начала координат (каноническим) уравнением следующего вида

$$v = (|z_1^2| + |z_2^2|) + (\varepsilon_1 z_1^2 + \varepsilon_2 z_2^2) + (\varepsilon_1 \bar{z}_1^{-2} + \varepsilon_2 \bar{z}_2^{-2}) + \sum_{k \geq 3} F_k(z, \bar{z}, u) \quad (1)$$

В изучаемом нами случае $\varepsilon_1 = \varepsilon_2 = \varepsilon$, $0 < \varepsilon \neq 1/2$. Здесь $z = (z_1, z_2)$, $w = u + iv$ – координаты в пространстве \mathbb{C}^3 , F_k – однородный многочлен суммарного веса k по z, \bar{z}, u (вес переменной u равен 2, а веса z_k, \bar{z}_k считаются равными 1).

Аффинная однородность M понимается как существование в группе аффинных преобразований 3-мерного комплексного пространства $Aff(3, \mathbb{C})$ некоторой подгруппы Ли $G(M)$, транзитивно действующей на поверхности M . Мы обсуждаем поверхности, однородные вблизи выделенной точки (для удобства эта точка – начало координат).

От группы Ли $G(M)$ переходим к инфинитезимальным преобразованиям поверхности, то есть к соответствующей алгебре $g(M)$ векторных полей, касательных к многообразию M . Эту алгебру можно считать состоящей из квадратных матриц 4-ого порядка с комплексными коэффициентами и имеющей некоторые особенности в строении базисных матриц.

Из однородности M вытекает наличие как минимум 5-мерной алгебры таких полей на M . Вопрос о возможных размерностях алгебр $g(M)$ можно считать одним из основных в задаче классификации однородных многообразий того или иного класса. Можно показать, что в нашем случае имеется три возможности для размерности $\dim_{\mathbb{R}} g(M)$: 5, 6 и 7.

Каждый элемент $g(M)$ можно записать в виде

$$Z = (A_1 z_1 + A_2 z_2 + A_3 w + p) \frac{\partial}{\partial z_1} + (B_1 z_1 + B_2 z_2 + B_3 w + s) \frac{\partial}{\partial z_2} + (a z_1 + b z_2 + c w + q) \frac{\partial}{\partial w} \quad (2)$$

где $A_k, B_k, a, b, c, p, s, q$ – комплексные константы.

Используя матричное представление, удобно записывать поле (2) в виде (4x4)- матрицы

$$\begin{pmatrix} A_1 & A_2 & A_3 & p \\ B_1 & B_2 & B_3 & s \\ a & b & c & q \\ 0 & 0 & 0 & 0 \end{pmatrix}. \quad (3)$$

При этом алгебра $g(M)$ превращается в матричную алгебру Ли (мы сохраним для нее то же обозначение) с операцией коммутирования матриц $[Z_1, Z_2] = Z_1 Z_2 - Z_2 Z_1$.

Можно построить описание всех однородных поверхностей на основе предварительного описания всех матричных алгебр Ли, возможных в этой задаче. Такой подход реализован в работе [1]. Важным начальным звеном такой схемы является использование уточненных канонических уравнений изучаемых вложенных подмногообразий (см. [2]). Получение оценок размерности изучаемых алгебр и последующий выход на итоговые результаты при этом существенно упрощаются.

2. Алгебра размерности семь

Рассмотрим вопрос о 7-мерных алгебрах Ли, которые могут соответствовать обсуждаемым однородным поверхностям. Несложно показать, что базис такой алгебры должен иметь вид

$$\begin{pmatrix} A1_1 & A2_1 & A3_1 & 1 \\ B1_1 & B2_1 & B3_1 & 0 \\ 2i(1+2\varepsilon) & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, \begin{pmatrix} A1_2 & A2_2 & A3_2 & i \\ B1_2 & B2_2 & B3_2 & 0 \\ 2(1-2\varepsilon) & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}.$$

$$\begin{pmatrix} A1_3 & A2_3 & A3_3 & 0 \\ B1_3 & B2_3 & B3_3 & 1 \\ 0 & 2i(1+2\varepsilon) & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} A1_4 & A2_4 & A3_4 & 0 \\ B1_4 & B2_4 & B3_4 & i \\ 0 & 2(1-2\varepsilon) & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} 0 & 0 & A3_5 & 0 \\ 0 & 0 & B3_5 & 0 \\ 0 & 0 & 2iq\lambda & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} 1 & 0 & A3_6 & 0 \\ 0 & 1 & B3_6 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix},$$

$$\begin{pmatrix} 0 & -1 & A3_7 & 0 \\ 1 & 0 & B3_7 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}.$$

$$c_i = \operatorname{Re}\{c_i\} + 2iq\lambda; \quad \operatorname{Re}\{c_i\} = 0, i = 1..5, 7, \quad \operatorname{Re}\{B1_i\} = 0, i = 1..6$$

Здесь в качестве «образующих» параметров выступают $p, s, q, \operatorname{Re}\{c\}, \operatorname{Re}\{B1\}$. Последние не равны 0 в случае $\operatorname{Re}\{c_6\} = 2, \operatorname{Re}\{B1_7\} = 1$.

Кроме того, обозначим элементы 2×2 блоков матриц как $t_{i+} t_{i+1}$. Согласно определению алгебры Ли, она замкнута относительно матричной скобки, то есть $\forall i, j = 1..7 \quad W_{ij} = [E_i, E_j] = \sum_{m=1}^7 \alpha_m E_m$. Сначала, из рассмотрения скобки W_{67} можно уточнить информацию о матрицах E_6, E_7 . Например, выполняются следующие равенства:

$$A3_7 = B3_6, \quad B3_7 = -A3_6.$$

Коэффициенты разложения α_m можно найти из четвертых столбцов W_{ij} . В итоге, мы имеем матричные тождества

$$[E_i, E_j] - \sum_{m=1}^7 \alpha_m E_m \equiv 0.$$

Выделяя различные подсистемы уравнений и решая их, мы пришли к 1-параметрическому семейству алгебр с базами

$$\begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 2i(1+2\varepsilon) & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} 0 & 0 & 0 & i \\ 0 & 0 & 0 & 0 \\ 2(1-2\varepsilon) & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 2i(1+2\varepsilon) & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix},$$

$$\begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & i \\ 0 & 2(1-2\varepsilon) & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}.$$

Эти алгебры Ли соответствуют, как несложно убедиться, аффинно-однородным квадратикам

$$v = (|z_1^2| + |z_2^2|) + \varepsilon(z_1^2 + z_2^2 + \bar{z}_1^{-2} + \bar{z}_2^{-2}).$$

Основные трудности при изучении однородных поверхностей по такой схеме связаны с пятимерными алгебрами.

3. Алгебры размерности пять

Базис пятимерной алгебры в обсуждаемом случае имеет общий вид:

$$\begin{pmatrix} A1_1 & A2_1 & A3_1 & 1 \\ B1_1 & B2_1 & B3_1 & 0 \\ 2i(1+2\varepsilon) & 0 & c_1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} A1_2 & A2_2 & A3_2 & i \\ B1_2 & B2_2 & B3_2 & 0 \\ 2(1-2\varepsilon) & 0 & c_2 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

$$\begin{pmatrix} A1_3 & A2_3 & A3_3 & 0 \\ B1_3 & B2_3 & B3_3 & 1 \\ 0 & 2i(1+2\varepsilon) & c_3 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} A1_4 & A2_4 & A3_4 & 0 \\ B1_4 & B2_4 & B3_4 & i \\ 0 & 2(1-2\varepsilon) & c_4 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} A1_5 & A2_5 & A3_5 & 0 \\ B1_5 & B2_5 & B3_5 & 0 \\ 0 & 0 & 2iq\lambda & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

Ниже предлагается способ упрощения процесса решения системы

$$W_{ij} - L = [E_i, E_j] - \sum_{m=1}^5 \alpha_m E_m \equiv 0. \text{ Эта система содержит квадратичные}$$

уравнения с большим числом неизвестных в ней и достаточно сложно

решается. Решение существенно упрощается, если удастся уменьшить число неизвестных в системе.

При этом сами неизвестные, т.е. A_k, B_k можно связать с коэффициентами канонических уравнений изучаемых поверхностей.

4. Основное соотношение и его обсуждения

Факт касания векторным полем (2) поверхности M можно записать в виде тождества $Re\{Z(\Phi)\}_{|M} \equiv 0$, где $\Phi = -Imw + F$ – определяющая функция поверхности. Представим это тождество, которое будем называть *основным соотношением*, в виде

$$Re\left\{ [A_1z_1 + A_2z_2 + A_3(u + iF) + p] \frac{\partial F}{\partial z_1} + [B_1z_1 + B_2z_2 + B_3(u + iF) + s] \frac{\partial F}{\partial z_2} + [az_1 + bz_2 + c(u + iF) + q] \frac{1}{2} \left(i + \frac{\partial F}{\partial u} \right) \right\} \equiv 0. \quad (4)$$

Левая часть тождества (4) есть некоторая аналитическая функция $H(z, \bar{z}, u)$. Учитывая канонический вид исходного уравнения поверхности, разложим эту функцию в весовое разложение и рассмотрим компоненты весов 0, 1, 2 основного тождества. В общем виде эти уравнения выписаны в [2] и [3]. В нашем случае простейшие следствия из компонент весов 0, 1 имеют вид:

$$a = 2i(\bar{p} + 2\bar{\varepsilon}p), \quad b = 2i(\bar{s} + 2\bar{\varepsilon}s), \quad q \in R$$

В компоненте веса 2, т.е. в равенстве

$$Re\left\{ p \frac{\partial F_3}{\partial z_1} + (A_1z_1 + A_2z_2) \frac{\partial F_2}{\partial z_1} + s \frac{\partial F_3}{\partial z_2} + (B_1z_1 + B_2z_2) \frac{\partial F_2}{\partial z_1} + \frac{1}{2}(az_1 + bz_2) \frac{\partial F_3}{\partial u} + \frac{1}{2}ic(u + iF_2) + \frac{1}{2}q \frac{\partial F_4}{\partial u} \right\} \equiv 0; \quad (5)$$

участвует многочлен

$$F_3 = (f_{0,3}z_2^3 + f_{1,2}z_1z_2^2 + f_{2,1}z_1^2z_2 + f_{3,0}z_1^3) +$$

$$+ \bar{z}_1(g_{0,2}z_2^2 + g_{1,1}z_1z_2 + g_{2,0}z_1^2) + \bar{z}_2(h_{0,2}z_2^2 + h_{1,1}z_1z_2 + h_{2,0}z_1^2) + (c, c)$$

из канонического уравнения поверхности (здесь через (c, c) обозначено выражение, комплексно сопряженное предыдущим слагаемым).

Его коэффициенты удобно выписывать следующим образом:

$$\begin{pmatrix} f_{0,3} & f_{1,2} & f_{2,1} & f_{3,0} \\ g_{0,2} & g_{1,1} & g_{2,0} & \\ h_{0,2} & h_{1,1} & h_{2,0} & \end{pmatrix}$$

В терминах этих коэффициентов тождество (5) распадается на 4 комплексных и 3 вещественных уравнения (итого 11 вещественных):

$$u: \quad Re(-2q\lambda + ic) = 0, \text{ откуда } Im(c_i) = 2q\lambda, \lambda \in \mathbb{R}$$

$$z_1^2: \quad sf_{2,1} + 3pf_{3,0} + 2\varepsilon A_1 + g_{2,0}\bar{p} + h_{2,0}\bar{s} - \varepsilon Re(c) = 0$$

$$z_2^2: \quad 2\varepsilon B_2 + 3sf_{0,3} + pf_{1,2} + g_{0,2}\bar{p} + h_{0,2}\bar{s} - \varepsilon Re(c) = 0$$

$$|z_1|^2: \quad Re(-2pg_{2,0} - sg_{1,1} + 1/2c + A_1) = 0$$

$$|z_2|^2: \quad Re(-2ph_{1,1} - 4sh_{0,2} + 1/2c + B_2) = 0$$

$$z_1 z_2 + \overline{z_1 z_2}: \quad 2pf_{2,1} + 2sf_{1,2} + 2\varepsilon A_2 + 2\varepsilon B_1 + g_{1,1}\bar{p} + h_{1,1}\bar{s} = 0$$

$$\overline{z_1 z_2} + z_2 z_1: \quad 2p\overline{h_{2,0}} + \overline{sh_{1,1}} + \overline{B_1} + pg_{1,1} + 2sg_{0,2} + A_2 = 0$$

Из 8 уравнений находятся выражения для элементов A_1, A_2, B_1, B_2, c через коэффициенты многочлена $F_3(z, \bar{z}, u)$ уравнения поверхности:

$$\begin{aligned} A_1 = & \frac{1}{2} Re c + \frac{1}{2\varepsilon} ((3Im f_{3,0} + i(-3Re f_{3,0} + Re g_{2,0}) - Im g_{2,0}) Im p + \\ & + (Im f_{2,1} + i(-Re f_{2,1} + Re h_{2,0}) - Im h_{2,0}) Im s + \\ & + (-3Re f_{3,0} - Re g_{2,0} + i(-3Im f_{3,0} - Im g_{2,0})) Re p + \\ & + (i(-Im f_{2,1} - Im h_{2,0}) - Re f_{2,1} - Re h_{2,0}) Re s) \end{aligned}$$

$$\begin{aligned} B_2 = & \frac{1}{2} Re c + \frac{1}{2\varepsilon} (Im f_{1,2} + i(-Re f_{1,2} + Re g_{0,2}) - Im g_{0,2}) Im p + \\ & + (3Im f_{0,3} + i(-3Re f_{0,3} + Re h_{0,2}) - Im h_{0,2}) Im s + \\ & + (i(-Im f_{1,2} - Im g_{0,2}) - Re f_{1,2} - Re g_{0,2}) Re p + \\ & + (-3Re f_{0,3} - Re h_{0,2} + i(-3Im f_{0,3} - Im h_{0,2})) Re s) \end{aligned}$$

$$B_1 = Re B_1 + \frac{i}{4\varepsilon} L$$

$$\begin{aligned}
A_2 &= -Re B_1 - \frac{1}{4\varepsilon} L + i(Im(g_{1,1} + 2h_{2,0})Im p + Im(g_{0,2} + 2h_{1,1})Im s + \\
&\quad Re(g_{1,1} + 2h_{2,0})Re p + Re(g_{0,2} + 2h_{1,1})Re s) \\
L &= Re(-2f_{2,1} - 2\varepsilon(-g_{1,1} + 2h_{2,0}) + g_{1,1})Im p + \\
&\quad Re(-2f_{1,2} - 2\varepsilon(-2g_{0,2} + h_{1,1}) + h_{1,1})Im s + \\
&\quad Im(-2f_{2,1} - 2\varepsilon(-g_{1,1} + 2h_{2,0}) + g_{1,1})Re p + \\
&\quad Im(-2f_{1,2} - 2\varepsilon(-2g_{0,2} + h_{1,1}) + h_{1,1})Re s
\end{aligned} \tag{6}$$

Оставшиеся 3 уравнения позволяют получить дополнительные зависимости между группами коэффициентов многочлена F_3 , а также аналогичные связи на коэффициенты многочлена F_4 . Для нас более важен именно F_3 , соответствующие зависимости выписаны ниже:

$$\begin{cases}
f_{1,2} - 3f_{3,0} - h_{1,1} + g_{2,0} = \overline{-g_{0,2}} - 2iIm g_{2,0} \\
f_{2,1} - 3f_{0,3} - g_{1,1} + h_{0,2} = \overline{-h_{2,0}} - 2iIm h_{0,2} \\
4\varepsilon h_{2,0} - 2f_{2,1} = \overline{g_{1,1}} - 2\varepsilon g_{1,1} \\
4\varepsilon g_{0,2} - 2f_{1,2} = \overline{h_{1,1}} - 2\varepsilon h_{1,1}
\end{cases} \tag{7}$$

Использование полученной информации о коэффициентах канонического уравнения может помочь в решении системы квадратичных уравнений.

5. Улучшение канонических уравнений поверхности

Еще один прием уменьшения числа неизвестных в этой системе (причем сразу на две единицы), связан с возможностью упрощения исходного канонического уравнения поверхности (1) в обсуждаемом случае.

Отметим, что имеется два типа преобразований,

$$\begin{pmatrix} z_1^* \\ z_2^* \\ w^* \end{pmatrix} = \begin{pmatrix} kz_1 \\ kz_2 \\ k^2w \end{pmatrix}, \quad \begin{pmatrix} z_1^* \\ z_2^* \\ w^* \end{pmatrix} = \begin{pmatrix} \cos(\phi)z_1 + \sin(\phi)z_2 \\ -\sin(\phi)z_1 + \cos(\phi)z_2 \\ w \end{pmatrix} \tag{8}$$

сохраняющих младшие слагаемые канонического уравнения, но изменяющие старшие веса, начиная с третьего.

Отметим, однако, что в полном объеме этот способ возможен только для моего случая в уравнении поверхности M , т.к. преобразование поворота при $\varepsilon_1 \neq \varepsilon_2$ не сохраняет квадрику.

При подстановке 3-го и 4-го уравнений в "системе связи" (7) коэффициентов f, g, h в выражение для L (6) мы получим:

$$\begin{aligned} 4\epsilon h_{2,0} - 2f_{2,1} &= \overline{g_{1,1}} - 2\epsilon g_{1,1}, \\ 4\epsilon g_{0,2} - 2f_{1,2} &= \overline{h_{1,1}} - 2\epsilon h_{1,1}, \\ L &= \operatorname{Re}(-2f_{2,1} - 2\epsilon(-g_{1,1} + 2h_{2,0}) + g_{1,1}) \operatorname{Im} p + \\ &+ \operatorname{Im}(-2f_{2,1} - 2\epsilon(-g_{1,1} + 2h_{2,0}) + g_{1,1}) \operatorname{Re} p + \\ &+ 2(1 - 2\epsilon) \operatorname{Re}(h_{1,1}) \operatorname{Im} s - 4\epsilon \operatorname{Im}(h_{1,1}) \operatorname{Re} s \end{aligned}$$

При выборе в преобразовании «частичного поворота» (8) такого угла ϕ , что $h_{1,1} = 0$, $B1$ и $A2$ примут вид:

$$\begin{aligned} B1 &= \operatorname{Re} B1 + \frac{i}{4\epsilon} L(s), \\ A2 &= -\operatorname{Re} B1 - \frac{1}{4\epsilon} L(s) + i(\operatorname{Im}(g_{1,1} + 2h_{2,0}) \operatorname{Im} p + \operatorname{Re}(g_{1,1} + 2h_{2,0}) \operatorname{Re} p) + \\ &+ i(\operatorname{Im}(g_{0,2}) \operatorname{Im} s + \operatorname{Re}(g_{0,2}) \operatorname{Re} s) \end{aligned}$$

Преобразованием "сжатия" можно добиться того, чтобы $A2 = -\operatorname{Re} B1 + iL_1(p)$ не зависело от s . Таким образом, получаем:

$$\begin{aligned} B1_1 &= \operatorname{Re} B1_1; & A2_1 &= -\operatorname{Re} B1_1 + iL_1(1) \\ B1_2 &= \operatorname{Re} B1_2; & A2_2 &= -\operatorname{Re} B1_2 + iL_1(i); \\ B1_3 &= \operatorname{Re} B1_3 + \frac{i}{4\epsilon} L(1); & A2_3 &= -\operatorname{Re} B1_3; \\ B1_4 &= \operatorname{Re} B1_4 + \frac{i}{4\epsilon} L(i); & A2_4 &= -\operatorname{Re} B1_4; \end{aligned}$$

При выборе таких значений ϕ, k мы смогли избавиться от 8 переменных t : четыре переменные t стали 0, а четыре другие переменные $t_k = \operatorname{Re}(A2)$ стали равны $-\operatorname{Re}(B1)$.

Заключение

Показано, что исследование вида канонического уравнения поверхности может существенно облегчить задачу нахождения алгебр в самом сложном – пятимерном случае.

Список литературы

1. Doubrov V.M. Homogeneous surfaces in the 3-dimensional affine geometry / V.M. Doubrov, V.P. Komrakov, M. Rabinovich // Geometry and Topology of Submanifolds. – VIII World Scientific.
2. Лобода А.В. Об одном семействе аффинно-однородных вещественных гиперповерхностей 3-мерного комплексного пространства /

А.В. Лобода, А.С. Ходарев // Известия вузов. Сер. Математика. – 2003. – № 10. – С. 38–50.

3. Лобода А.В. Об аффинной однородности поверхностей трубчатого типа в S^3 / А. В. Лобода, Т. Т. З. Нгуен // Труды МИАН им. В.А. Стеклова. – 2012. – Т. 279. – С. 1–18.

РЕАЛИЗАЦИЯ ПОВТОРИТЕЛЯ НА БАЗЕ MSP430 С ИСПОЛЬЗОВАНИЕМ ПРОТОКОЛА SIMPLICITY

А.А. Решетов
Студент

Д.Н. Борисов
Доцент

Введение

На сегодняшний день, при создании какой-либо беспроводной сети между устройствами, основополагающими факторами являются: экономия энергии при эксплуатации, экономия при покупке необходимого оборудования и программного обеспечения, возможность применения данной сети в разных сферах. Причем, если все эти факторы не будут учтены, то сеть может оказаться попросту не эффективной, приобретенное оборудование может потреблять большое количество энергии и в результате использование такой сети обойдется куда дороже, чем ее создание, а при смене сферы деятельности, сеть окажется непригодной.

1. Повторитель на основе микроконтроллера MSP430

В ходе работы, был реализован повторитель, который является отдельным компонентом сети и не связан с другими устройствами. Такой повторитель автоматически подключается к сети, при его включении. При этом нет необходимости настраивать соединение после каждого обрыва связи.

Повторитель был реализован на базе микроконтроллера MSP430 [1, 2] с использованием протокола Simplicity [3]. Различная стоимость такого микроконтроллера примерно 5\$. Протокол Simplicity является свободно распространяемым продуктом компании Texas Instruments.

Кроме того, при разработке использовался отладочный комплект eZ430, который позволяет запрограммировать неограниченное количество устройств. На рис. 1 представлен отладочный набор с микроконтроллером.

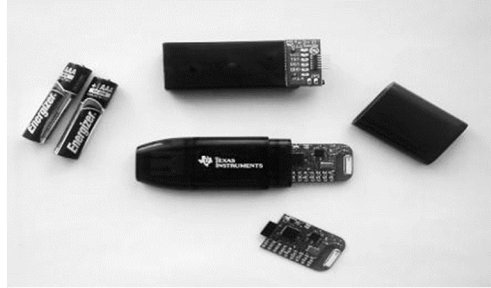


Рис. 1. Отладочный комплект eZ430 и микроконтроллер MSP430

В этом случае при программировании микроконтроллера используется персональный компьютер с установленной интегрированной средой разработки приложений IAR Embedded Workbench.

Основное назначение повторителя – способность увеличивать расстояние приема-передачи между узлами сети. Опытным путем было установлено, что максимальное расстояние между двумя узлами, реализованное на основе микроконтроллера MSP430, не должно превышать 40 метров. Естественно при использовании повторителей расстояние увеличивается кратно их количеству. В реальных условиях при передаче данных узлы не всегда располагаются на открытой местности в пределах прямой видимости. Зачастую в помещениях даже при небольших расстояниях при передаче данных из-за наличия преград может понадобиться повторитель. В этом случае повторитель становится новым узлом сети через который реализуется передача данных. Так, например, в офисных помещениях в качестве преград могут выступать бетонные стены и металлические конструкции.

Сфера применения устройств с повторителями может быть очень разнообразна. На основе таких устройств можно создавать самоорганизующиеся сенсорные сети, узлы которых устанавливаются произвольно на некоторой протяженной местности. Используя различные датчики, например, датчики по контролю влаги, температуры, дыма, движения, проникновения, удара и т.д., стоимость которых незначительна, можно проводить контроль или исследование окружающей среды в режиме реального времени на достаточно протяженных расстояниях.

На рис. 2 представлена топология беспроводной сети с использованием протокола SimpliciTI.

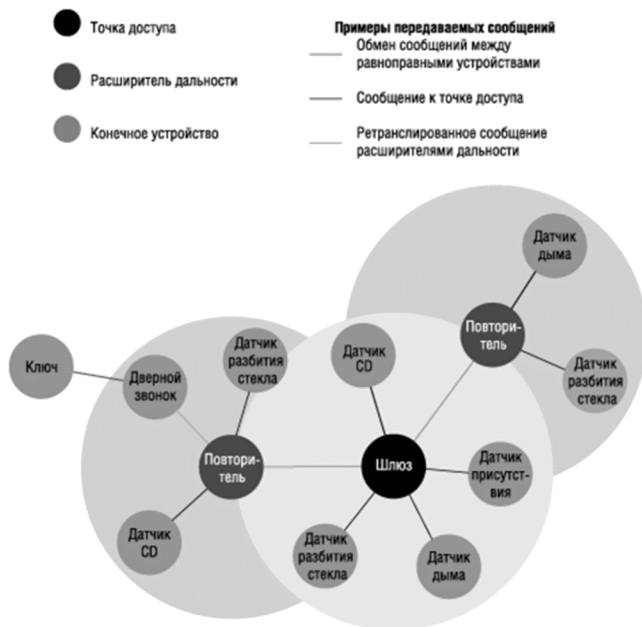


Рис. 2. Топология сети

При включении и отключении одного узла все остальные узлы продолжают свою работу без учета работы других узлов. В результате, если у нас нет необходимости постоянно контролировать какой-то показатель постоянно, можно в определенное время считывать данные в конкретный промежуток времени, а в остальное время эти узлы отключать. Данный подход позволит значительно экономить энергию потребления одного микроконтроллера.

Алгоритм работы повторителя следующий: полученные данные узел отправляет на повторитель, а тот в свою очередь дублирует сигнал, после чего данные получает конечное устройство. На рис. 3 представлена блок-схема алгоритма реализации конечного устройства.

Для контроля данного процесса можно использовать программу Sensor Monitor фирмы Texas Instruments [4]. На рис. 4 показан, случай передачи информации о температуре датчика и потребляемой энергии между двумя узлами через повторитель.

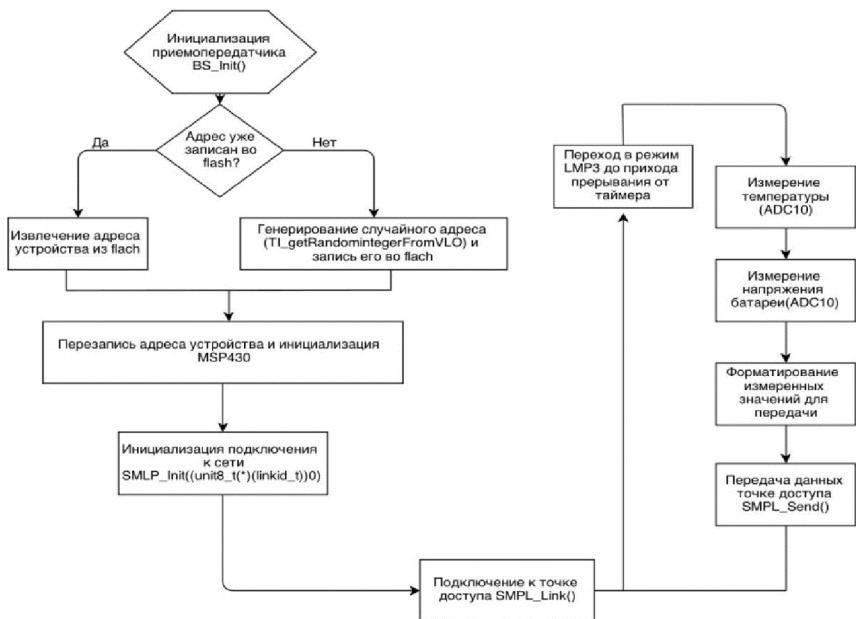


Рис. 3. Блок-схема алгоритма реализации конечного устройства

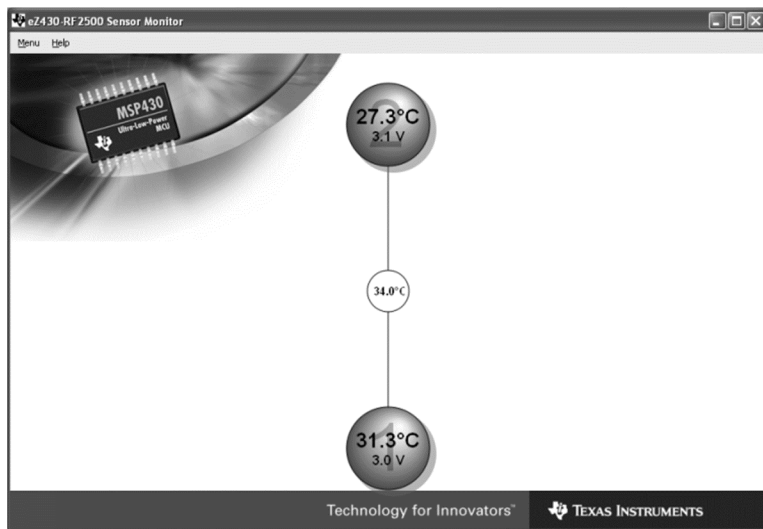


Рис. 4. Контроль передачи данных с помощью Sensor Monitor

2 Моделирование в Proteus

Для исследования работы сети с повторителями используем среду Proteus [4, 5]. Среда Proteus была выбрана благодаря наличию микроконтроллера MSP430 в стандартной библиотеке компонентов. Среда позволяет использовать микроконтроллер и изменять необходимые параметры (рис. 5).

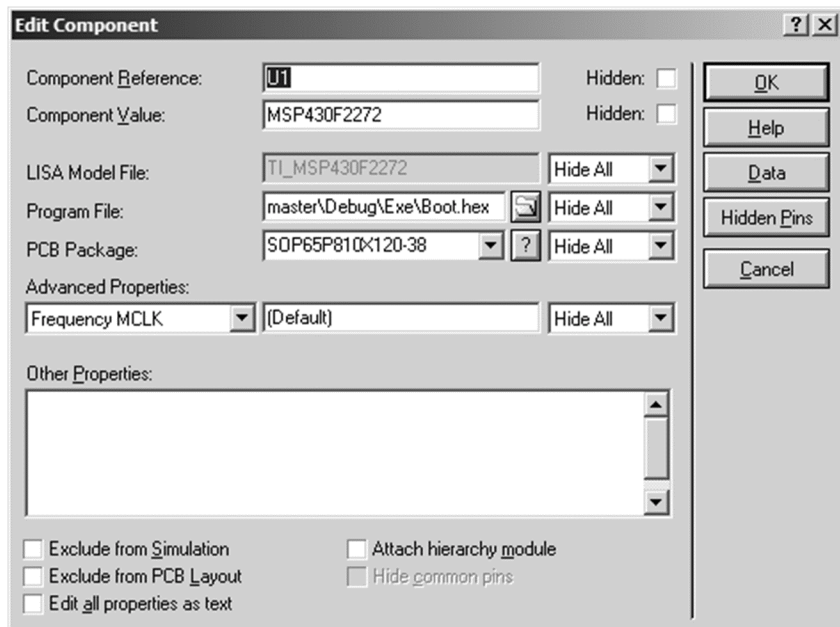


Рис. 5. Компонент MSP430 в среде Proteus

Данная среда позволяет моделировать сеть с учетом протокола передачи данных, но в ней отсутствует возможность описывать беспроводные сети. В связи с этим использовалась некоторая идеализация реального случая – беспроводной канал был заменен на проводной. Однако данная идеализация не приводит к принципиальным неточностям и может использоваться для анализа.

С учетом вышеизложенного, для анализа работоспособности сети была смоделирована сеть, состоящая из 20 узлов, работающая по протоколу SimpliCITI (рис. 6).

Программа Proteus, используя компонент SPI Debug, позволяет проводить анализ конфликтов, возникающих при передаче данных (рис. 7).

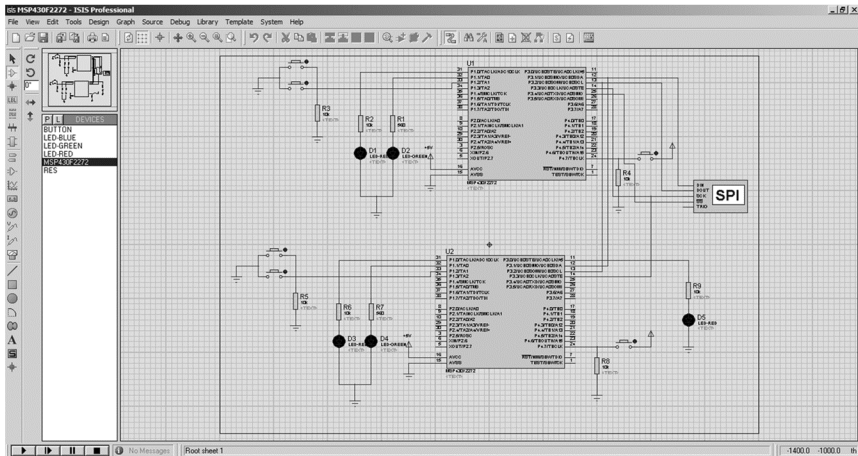


Рис. 6. Моделирование сети в среде Proteus

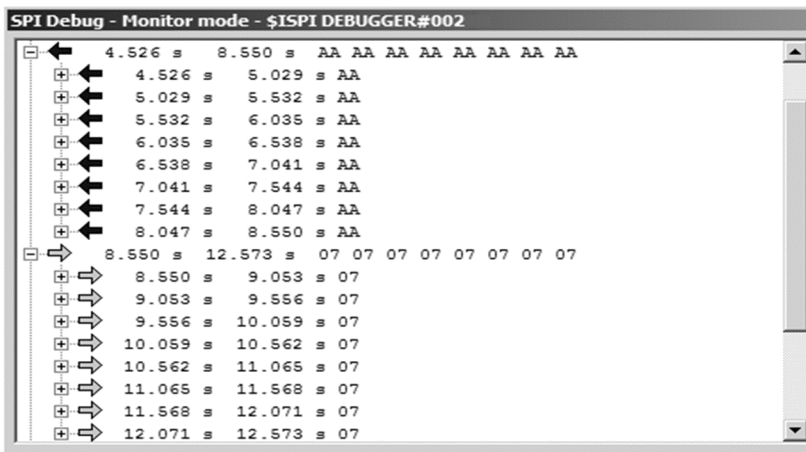


Рис. 7. Анализ работы сети в SPI Debug

Исследования показали, что микроконтроллер MSP430 пригоден для использования в качестве повторителя сети, построенной на основе протокола SimpliciTI.

Заключение

С использованием среды Proteus была создана модель беспроводной сети, построенной на основе протокола SimpliciTI и состоящей из 20 узлов каждый из которых включал микроконтроллер MSP430. Анализ данной

модели показал пригодность использования микроконтроллера MSP430 в качестве повторителей.

В качестве лабораторного эксперимента реализована беспроводная сеть на основе протокола SimpliCI TI состоящая из 5 узлов на базе микроконтроллера MSP430. Данная сеть содержала 2 конечных устройства и 3 повторителя. Для примера работоспособности сети была осуществлена передача температуры устройства на расстояния, превышающие предел прямой видимости узлов.

Таким образом, проведенные исследования показали, целесообразность и возможность создания сети на основе протокола SimpliCI TI и микроконтроллера MSP430.

Список литературы

1. Микроконтроллер MSP 430. – Режим доступа: [http://msp430-ru.blogspot.ru](http://msp430.ru.blogspot.ru)
2. Микроконтроллеры MSP430 – средства отладки. – Режим доступа: <http://www.phyton.ru/pages/page30.html>
3. Протокол SimpliCI TI. – Режим доступа: <http://www.ti.com/tool/simpliCI TI>
4. Среда Proteus. – Режим доступа: http://www.labcenter.com/download/prodemo_download.cfm
5. Использование среды Proteus. – Режим доступа: <http://nov-gen.ru/programmy/o-programme-proteus-chast-1>

СИМУЛЯЦИЯ ДВИЖЕНИЯ МЯГКИХ ТЕЛ (ОДЕЖДЫ) В КОМПЬЮТЕРНОЙ ГРАФИКЕ

А.С. Суровцев
Студент

Н.А. Тюкачёв
Доцент

И.С. Губарев
Ассистент

Введение

В компьютерной индустрии для отражения физических процессов, происходящих в макромире, существует две группы моделей. Первая группа описывает поведение жестких тел (Rigid Bodies). Вторая – поведение мягких тел (Soft Bodies). Симуляция динамики и взаимодействия первой группы моделей на данный момент времени уже не

является проблемой, как со стороны требуемой точности, так и со стороны скорости вычислений. Построение моделей второй группы сложнее как для исследований, так и при визуализации. Примерами моделей второй группы являются поведение ткани и динамика жидкости или газа. Визуализация характера поведения второй группы моделей является задачей, которая предполагает большое количество вычислений, что затрудняет интерактивное проведение самой симуляции. Внедрение различных математических подходов позволяет улучшить желаемые показатели.

Цель данной работы – разработать программный продукт, визуализирующий поведение одежды.

1. Описание модели

Для описания ткани основное развитие получила модель, включающая в себя частицы, являющиеся материальными точками и невесомые пружины. Концы всех пружин обязательно прикреплены к каким-либо точкам и концы одной пружины не могут находиться на одной и той же точке. Параметры модели для точек: положение в пространстве, скорость, масса, приложенная сила. Параметры модели для пружин: соединения с точками, длина в нерастянутом состоянии и коэффициент упругости. В стандартной ситуации выделяют три типа пружин:

1. Структурные соединяют частицы $[m, n]$ и $[m + 1, n]$, $[m, n]$ и $[m, n + 1]$.
2. Сдвига – $[m, n]$ и $[m + 1, n + 1]$, $[m + 1, n]$ и $[m, n + 1]$.
3. Сгиба – $[m, n]$ и $[m + 2, n]$, $[m, n]$ и $[m, n + 2]$.

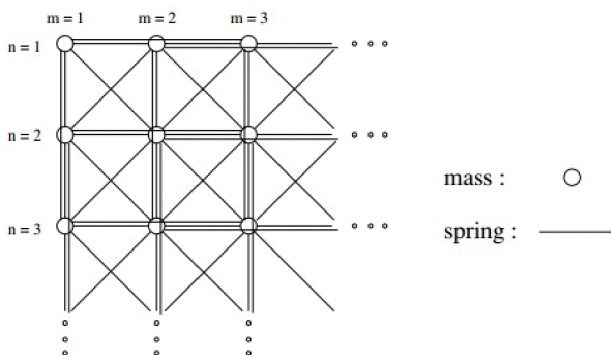


Рис. 1. Соединение пружинами частиц в модели

Эволюция модели во времени задается обновлением положения частиц в пространстве на основе второго закона Ньютона. На каждую точку действует сила тяжести и силы упругости прикрепленных к ней

пружин. Вычисление нового положения достигается путем применения численных методов интегрирования [1].

Неотъемлемой частью моделирования является обнаружение и обработка столкновений частей ткани.

Для регистрации столкновений используют два описания:

1. Столкновение частицы и треугольника. Для нахождения требуется решить квадратное уравнение. Регистрируется основная часть столкновений.

2. Столкновения двух ребер. Для нахождения требуется решение уравнения третьей степени. Регистрирует все возможные столкновения.

Обработка столкновений производится в два этапа:

1. Обработка единичных столкновений. При разрешении используется закон сохранения импульса.

2. Обработка ригидных зон столкновений. Регистрируются остаточные после обработки единичных столкновений. Получается множество зон, для которых обработка выполняется с применением закона сохранения момента импульса [2].

Столкновения рассматриваются между двумя положениями точек. Между этими положениями движение каждой точки считается равномерным и прямолинейным. Вычисляется средняя скорость движения каждой точки. Зная начальное положение и среднюю скорость, находят возникшие столкновения. Начинается первый этап обработки столкновений. После применения закон сохранения для обработки столкновения, вычисляются новые средние скорости. Находится разность средних скоростей до и после обработки столкновений. Эти разности умножаются на промежуток времени, который соответствует перемещению точек из начальных в конечные положения. Получается новое положение точки с уже разрешенным столкновением. Остается обновить скорости точек в конечных положениях. Скорость вычисляется как разность двух положений, поделенная на величину промежутка времени. Второй этап обработки столкновений происходит по тому же сценарию, что и первый. По исходу двух этапов обработки столкновений модель избавлена от столкновений [3].

Процесс эволюции модели во времени принято называть внутренней динамикой. Реализации внутренней динамики и обработки столкновений являются независимыми. Обработку столкновений можно производить не после одного пересчета внутренней динамики, а после нескольких [3].

2. Реализация модели

Для реализации внутренней динамики был выбран метод интегрирования Верле [4].

$$\overrightarrow{x(t + \Delta t)} = 2 * \overrightarrow{x(t)} - \overrightarrow{x(t - \Delta t)} + 2 * \frac{d^2 \overrightarrow{x(t)}}{dt^2} * \frac{\Delta t}{2} + O(\Delta t^4). \quad (1)$$

Таким образом, для расчета положения точки в пространстве для следующего момента времени требуется знать:

- текущую координату $\overrightarrow{x(t)}$,
- координату в предыдущий момент времени $\overrightarrow{x(t - \Delta t)}$
- и ускорение точки в данный момент времени $\frac{d^2 \overrightarrow{x(t)}}{d t^2}$.

Ускорение же в свою очередь получаем из второго закона Ньютона:

$$m * \frac{d^2 \overrightarrow{x(t)}}{d t^2} = \vec{F}_{\text{равн}} \quad (2)$$

где $\vec{F}_{\text{равн}}$ – сила, равнодействующая приложенным к точке силам.

Для увеличения скорости регистрации (или обнаружения) было принято решение искать столкновения одного элемента (будь то частица или ребро) не со всеми элементами модели (соответственно треугольником или ребром), а с теми, которые делят с фиксированным элементом некую область пространства. Для этого было принято решение использовать пространственный метод, предложенный в курсе Real Time Physics Class от nVidia, раздел Collision Detection. Здесь предлагается использовать хэш-таблицу. Каждая ячейка хэш-таблицы является списком, который будет содержать множество элементов, содержащихся в определенной области, между которыми и будут искаться столкновения. Положения объекта в хэш-таблице вычисляется:

$$pos = ((x * 92837111) xor (y * 689287499) xor (z * 283923481)) \quad (3)$$

mod tableSize,

где pos – искомое положение в хэш-таблице; $tableSize$ – размер хэш-таблицы; x, y и z – положение точки, разделенное на размер области [5].

3. Нестабильность модели

В ходе экспериментов, выполненных на первых стадиях разработки модели, было получено то, что возникают ситуации, когда частицы начинают колебаться. Нередко это приводит к резонансу и выведению модели из стабильного состояния (рис. 2). Возникновение таких нестабильностей связано с методом интегрирования. В результате равнодействующая сила, приложенная к частице за весь промежуток времени Исползование такого грубого приближения, приводят к тому, что некоторые пружины оказываются либо сильно сжатыми, либо сильно растянутыми. Избегать неточности за счет уменьшения времени дискретизации модели зачастую не удается.

Решение данной проблемы можно найти в применении другого метода интегрирования. Для этого был реализован метод Рунге-Кутты 4 порядка. Однако это не избавило от нестабильностей, а только сдвинуло порог параметров модели для проблемных участков.

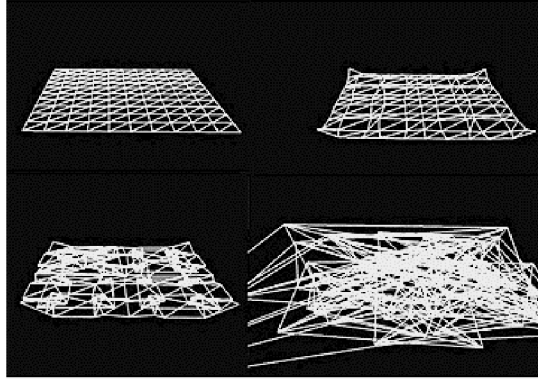


Рис. 2. Пример нестабильности модели

Еще одним путем решения является введение коэффициентов затухания:

1. При расчете сил, действующих со стороны пружин [3].
2. При расчете положения точек (для уменьшения скоростей движения точек) [6].

Это также не позволило избавиться от нестабильностей модели.

4. Решение проблемы с нестабильностью

Решение данной проблемы было найдено в работе [7]. Здесь предлагается для определения положения использовать неявный метод Эйлера:

$$M(\vec{x} - \vec{y}) = h^2 * \vec{f}(x), \quad (4)$$

где

$$\vec{x} = \vec{q}_{n+1},$$

$$\vec{y} = 2 * \vec{q}_n - \vec{q}_{n-1},$$

M – матрица масс частиц,

\vec{q}_{n-1} – предыдущее положение точек,

\vec{q}_n – текущее положение точек,

\vec{q}_{n+1} – вычисляемое положение точек,

h – интервал дискретизации,

$\vec{f}(x)$ – вектор столбец равнодействующих сил действующих на частицы.

Метод интегрирования является неявным потому, что и в левой части и в правой части как параметр функции встречаются вычисляемые значения. Решение уравнения (4) производится итеративно путем минимизации растяжения пружин (столбец равнодействующих сил разложен на два $-J * \vec{d}$ и \vec{f}_{ext}):

$$(M + h^2 * L) * \vec{x} = M * \vec{y} + h^2 * J * \vec{d} + h^2 * \vec{f}_{ext}, \quad (5)$$

где \vec{f}_{ext} – вектор-столбец внешних сил, действующих на частицы.

Данный метод не заменяет используемых ранее методов интегрирования, а корректирует положения точек, после основного метода интегрирования. Поэтому первоначальное значение \vec{y} задается как новое положение, вычисленное явным методом интегрирования. При этом решается линейное уравнение (5) относительно \vec{x} . Если не достигается допустимое значение по растяжению пружин, заново выполняется составление системы, в этом случае за \vec{y} принимается \vec{x} и снова решается уравнение.

После проведения нескольких итераций решения системы удастся избавиться модель от колебания частиц и выведению модели из стабильного состояния [7]. Но эти вычисления требуют больших вычислительных возможностей. В связи с этим алгоритмы внутренней динамики и разрешения столкновений переводят на выполнение графическими процессорами. Ведь даже бюджетные современные графические процессоры состоят из сотни ядер, которые в сумме дают вычислительные возможности (количество операций в единицу времени), превосходящие в десятки раз возможности центрального процессора.

Заключение

Разработан программный продукт, реализующий симуляцию поведения ткани с использованием различных методов интегрирования. Внедрено пространственное хеширование для уменьшения числа сравнений при обнаружении столкновений. С помощью неявного интегрирования удалось избежать возникновения колебания частиц и нестабильных ситуаций без экспоненциального увеличения числа итераций в единицу времени.

Список литературы

1. Provot X. Deformable constraints in a mass-spring model to describe rigid cloth behavior. – Режим доступа: <http://graphics.stanford.edu/courses/cs468-02-winter/Papers/Rigidcloth.pdf>
2. Provot X. Collision and self-collision handling in cloth model dedicated to design garments. – Режим доступа: <http://kucg.korea.ac.kr/seminar/2004/src/PA-04-18.pdf>
3. Bridson R. Robust treatment of collisions, contact and friction for cloth animation – Режим доступа: <http://graphics.cs.cmu.edu/courses/15-869/lec/07/p594-bridson.pdf>
4. Müller M. Position Based Dynamics. – Режим доступа: <http://www.mathiasmueller.info/publications/posbaseddyn.pdf>
5. Müller M. Real Time Physics Class Note / M. Müller, J. Stam, D. James, J. Ratcliff // Collision Detection, 2008. – NVIDIA

6. Verlet integration. – Режим доступа: <http://www.saylor.org/site/wp-content/uploads/2011/06/MA221-6.1.pdf>

7. Fast simulation of mass-springs systems. – Режим доступа: <http://graphics.berkeley.edu/papers/Liu-FSM-2013-11/Liu-FSM-2013-11.pdf>

ПОЛЕ НЕЧЕТКИХ ЧИСЕЛ

А.О. Шевляков
Студент

М.Г. Матвеев
Профессор

Введение

В задачах управления часто возникает неопределенность. Для её разрешения в настоящее время используются случайные числа (описываемые в теории вероятностей) или нечеткие числа (теория нечетких множеств и нечеткой логики) [1]. Для использования случайных чисел нам нужно знать распределение, которое во многих случаях нет возможности получить. В то же время нечеткие числа хорошо представляют данные, получаемые от экспертов. Отсюда большой интерес к развитию данного направления.

Большинство методов решения управленческих задач разработаны для обычных, четких чисел. Потому для применения этих методов необходим аппарат, позволяющий совершать над нечеткими числами те же операции, что и над четкими числами. Известны два основных направления реализации вычислительных операций с нечеткими числами: введение арифметики нечетких чисел (в частности L-R чисел) [2], использование α -уровневого принципа обобщения и интервальной алгебры [1]. Нечеткие арифметики и интервальные алгебры имеют ряд существенных недостатков, к которым относятся: неоправданное расширение неопределенности в результате выполнения операций; невыполнение естественных математических соотношений, например, вычитание равных операндов не дает в результате нуль, даже нечеткий. Это связано с отсутствием на множестве нечетких чисел адекватной алгебраической структуры. Цель статьи – попытка введения такого представления нечеткого числа, которое обеспечит требуемую алгебраическую структуру.

Поле нечетких чисел

Было решено пойти по пути введения арифметики для LR-чисел (в частности, треугольных чисел) и развить её до поля. Традиционным способом представления треугольного числа является задание тройкой чисел (α, a, β) , где α и β – соответственно левый и правый коэффициенты нечеткости, a – мода треугольного числа [2].

Был предложен совершенно иной подход к представлению треугольного числа. Был введен некий аналог массы для таких чисел, рассчитываемый как масса плоской пластины с переменной плотностью $\gamma(x, y) = x$. Масса такой пластины рассчитывается по формуле

$$m = \iint_D \gamma(x, y) dx dy. \text{ В случае треугольников интеграл разбивается на два –}$$

по левой и правой частям треугольника. Решив интегралы можно массу левой и правой частей вычислить с помощью моды a и соответственно левых q и правых p коэффициентов нечеткости. Масса левой части

$$m_l = \frac{qa}{2} - \frac{q^2}{6}, \text{ масса правой – } m_r = \frac{pa}{2} + \frac{p^2}{6}. \text{ Имея массу какой-либо}$$

части треугольника, можно по имеющейся моде, решив квадратное уравнение и отбросив неподходящее отрицательное значение, однозначно восстановить коэффициент нечеткости. В таком случае треугольное число можно задавать тройкой чисел – массой левой части m_l , модой a и массой правой части m_r .

Определим основные операции (сложение, умножение, вычитание, деление) на представленных таким образом нечетких числах. Полная масса в результате операции определяется путем применения данной операции к полным массам операндов.

Операция сложения вводилась из следующего соображения: разность масс правой и левой частей результата операции равна сумме разностей масс обоих слагаемых:

$$M_r - M_l = (m_{r1} - m_{l1}) + (m_{r2} - m_{l2}).$$

В итоге операция сложение примет вид:

$$(m_{l1}, a_1, m_{r1}) + (m_{l2}, a_2, m_{r2}) = (m_{l1} + m_{l2}, a_1 + a_2, m_{r1} + m_{r2}).$$

Вводится нулевой и единичный элемент.

Нулевой элемент – четкий ноль:

$$(0, 0, 0).$$

Единичное число:

$$\left(\frac{1}{2}, 1, \frac{1}{2}\right).$$

Противоположное число определяется из условия равенства нулю суммы противоположного и самого числа и имеет вид:

$$-(m_l, a, m_r) = (-m_l, -a, -m_r).$$

Вычитание вводится как операция, обратная сложению:

$$(m_{l1}, a_1, m_{r1}) - (m_{l2}, a_2, m_{r2}) = (m_{l1} - m_{l2}, a_1 - a_2, m_{r1} - m_{r2}).$$

Умножение вводилось таким образом: разность масс правой и левой части результата операции равна сумме разности масс первого числа умноженного на полную массу второго и разности масс второго числа умноженного на полную массу первого:

$$M_r - M_l = (m_{r2} + m_{l2}) \cdot (m_{r1} - m_{l1}) + (m_{r1} + m_{l1}) \cdot (m_{r2} - m_{l2}).$$

Умножение примет вид:

$$(m_{l1}, a_1, m_{r1}) (m_{l2}, a_2, m_{r2}) = \left(\frac{-m_{r1}m_{r2} + m_{r1}m_{l2} + m_{l1}m_{r2} + 3m_{l1}m_{l2}}{2}, \right. \\ \left. a_1 a_2, \frac{3m_{r1}m_{r2} + m_{r1}m_{l2} + m_{l1}m_{r2} - m_{l1}m_{l2}}{2} \right).$$

Обратное число определяется из условия равенства единице произведения обратного и самого числа и имеет вид:

$$(m_l, a, m_r)^{-1} = \left(\frac{m_r}{(m_l + m_r)^2}, \frac{1}{a}, \frac{m_l}{(m_l + m_r)^2} \right).$$

Деление вводится как операция, обратная умножению:

$$(m_{l1}, a_1, m_{r1}) \div (m_{l2}, a_2, m_{r2}) = \left(\frac{m_{r1}m_{r2} - m_{r1}m_{l2} + 3m_{l1}m_{r2} + m_{l1}m_{l2}}{2(m_{r2} + m_{l2})^2}, \right. \\ \left. \frac{a_1}{a_2}, \frac{m_{r1}m_{r2} + 3m_{r1}m_{l2} - m_{l1}m_{r2} + m_{l1}m_{l2}}{2(m_{r2} + m_{l2})^2} \right).$$

В построенной таким образом системе выполняются свойства, предъявляемые к полю.

Коммутативность сложения:

$$(m_{l1}, a_1, m_{r1}) + (m_{l2}, a_2, m_{r2}) = (m_{l2}, a_2, m_{r2}) + (m_{l1}, a_1, m_{r1}) \\ = (m_{l1} + m_{l2}, a_1 + a_2, m_{r1} + m_{r2}).$$

Коммутативность умножения:

$$(m_{l1}, a_1, m_{r1}) (m_{l2}, a_2, m_{r2}) = (m_{l2}, a_2, m_{r2}) (m_{l1}, a_1, m_{r1}) \\ = \left(\frac{-m_{r1}m_{r2} + m_{r1}m_{l2} + m_{l1}m_{r2} + 3m_{l1}m_{l2}}{2}, a_1 a_2, \right. \\ \left. \frac{3m_{r1}m_{r2} + m_{r1}m_{l2} + m_{l1}m_{r2} - m_{l1}m_{l2}}{2} \right).$$

Ассоциативность сложения:

$$\begin{aligned}
& ((m_{l1}, a_1, m_{r1}) + (m_{l2}, a_2, m_{r2})) + (m_{l3}, a_3, m_{r3}) = \\
& (m_{l1}, a_1, m_{r1}) + ((m_{l2}, a_2, m_{r2}) + (m_{l3}, a_3, m_{r3})) = \\
& (m_{l1} + m_{l2} + m_{l3}, a_1 + a_2 + a_3, m_{r1} + m_{r2} + m_{r3}).
\end{aligned}$$

Ассоциативность умножения:

$$\begin{aligned}
& ((m_{l1}, a_1, m_{r1})(m_{l2}, a_2, m_{r2}))(m_{l3}, a_3, m_{r3}) = \\
& (m_{l1}, a_1, m_{r1})((m_{l2}, a_2, m_{r2})(m_{l3}, a_3, m_{r3})) = \\
& (-m_{r1}m_{r2}m_{r3} + m_{r1}m_{l2}m_{l3} + m_{r2}m_{l1}m_{l3} + m_{r3}m_{l1}m_{l2} + 2m_{l1}m_{l2}m_{l3}, \\
& a_1a_2a_3, 2m_{r1}m_{r2}m_{r3} + m_{r1}m_{r2}m_{l3} + m_{r1}m_{r3}m_{l2} + m_{r2}m_{r3}m_{l1} - m_{l1}m_{l2}m_{l3}).
\end{aligned}$$

Разность одинаковых нечетких чисел дает четкий ноль:

$$(m_l, a, m_r) - (m_l, a, m_r) = (m_l - m_l, a - a, m_r - m_r) = (0, 0, 0).$$

Деление одинаковых чисел всегда дает одну и ту же нечеткую единицу:

$$\begin{aligned}
(m_l, a, m_r) \div (m_l, a, m_r) &= \left(\frac{m_r m_r - m_r m_l + 3m_l m_r + m_l m_l}{2(m_r + m_l)^2}, \frac{a}{a}, \right. \\
&\left. \frac{m_r m_r + 3m_r m_l - m_l m_r - m_l m_l}{2(m_r + m_l)^2} \right) = \left(\frac{1}{2}, 1, \frac{1}{2} \right).
\end{aligned}$$

При сложении (как и при вычитании) с четким нулем получается то же самое число:

$$(m_l, a, m_r) + (0, 0, 0) = (m_l, a, m_r).$$

При умножении (как и при делении) на нечеткую единицу будет получаться то же число:

$$\begin{aligned}
(m_l, a, m_r) \left(\frac{1}{2}, 1, \frac{1}{2} \right) &= \left(\frac{-\frac{1}{2}m_r + \frac{1}{2}m_r + \frac{1}{2}m_l + \frac{3}{2}m_l}{2}, \right. \\
&\left. a, \frac{\frac{3}{2}m_r + \frac{1}{2}m_r + \frac{1}{2}m_l - \frac{1}{2}m_l}{2} \right) = (m_l, a, m_r).
\end{aligned}$$

Умножение на нечеткую двойку (1,2,1) равнозначно сложению одинаковых чисел:

$$\begin{aligned}
(m_l, a, m_r)(1, 2, 1) &= \left(\frac{-m_r + m_r + m_l + 3m_l}{2}, 2a, \right. \\
&\left. \frac{3m_r + m_r + m_l - m_l}{2} \right) = (2m_l, 2a, 2m_r) = (m_l, a, m_r) + (m_l, a, m_r).
\end{aligned}$$

Дистрибутивность умножения относительно сложения:

$$\begin{aligned}
& ((m_{I1}, a_1, m_{r1}) + (m_{I2}, a_2, m_{r2}))(m_{I3}, a_3, m_{r3}) = \\
& \left(\frac{-(m_{r1} + m_{r2})m_{r3} + (m_{r1} + m_{r2})m_{I3} + (m_{I1} + m_{I2})m_{r3} + 3(m_{I1} + m_{I2})m_{I3}}{2}, \right. \\
& \left. a_1 a_2, \frac{3(m_{r1} + m_{r2})m_{r3} + (m_{r1} + m_{r2})m_{I3} + (m_{I1} + m_{I2})m_{r3} - (m_{I1} + m_{I2})m_{I3}}{2} \right) \\
& = (m_{I1}, a_1, m_{r1})(m_{I3}, a_3, m_{r3}) + (m_{I2}, a_2, m_{r2})(m_{I3}, a_3, m_{r3}).
\end{aligned}$$

Рассмотрим пример использования данных операций на примере простейшего линейного уравнения:

$$Ax + B = 0,$$

где $A = (1, 2, 3)$, $B = (2, 4, 2)$.

$$\text{Найдем } x = -\frac{B}{A} = -\frac{(2, 4, 2)}{(1, 2, 3)} = \left(-\frac{3}{4}, -2, -\frac{1}{4}\right).$$

Подставив в исходное уравнение получим тождество.

Таким образом определенная система представляет собой поле.

В некоторых случаях (когда разность масс правой и левой части в результате операции превышает массу всего числа) при вычитании получаются числа, которые для восприятия человеком требуют дополнительной обработки. Придумано несколько способов приведения к обычному виду – так что это не должно сказаться на работоспособности поля.

Заключение

Было сформировано поле нечетких чисел, выполняются все необходимые свойства. Следует в дальнейшем исследовать результаты, даваемые применением данного поля для решения различных управленческих задач. При положительных результатах возможно дальнейшее развитие метода, применение его для задач различного рода.

Список литературы

1. Левин В.И. Интервальная математика и изучение неопределенных систем / В.И. Левин // Наука и образование : электронное научно-техническое издание. – 2005. – №5.

2. Яхьяева Г. Основы теории нечетких множеств // Национальный открытый университет "Интуит". – Режим доступа: http://inter-vuz.tuit.uz/Elib_baza/INTUIT.ru/html/department/ds/fuzzysets/index.html

АНАЛИЗ ДАННЫХ И ПРОГНОЗИРОВАНИЕ В 1С: ПРЕДПРИЯТИЕ 8

А.С. Шилов
Студент

А.И. Другалев
Ассистент

Введение

Начиная с версии 1С:Предприятия 8 фирма 1С начала встраивать в платформу механизм анализа данных. До версии платформы 8.3.4.456, механизм анализа данных развивается и расширяется, но пока в типовых конфигурациях этот механизм не используют, т.к. недостаточно продумано использование различных реализованных подуровней механизма анализа данных. К тому же, большая часть пользователей 1С:Предприятие ведет в программе бухгалтерский или регламентированный учет, не используя до конца даже типовых возможностей 1С.

С развитием возможностей 1С, с усовершенствованием платформы и конфигураций добавляются разные механизмы, с помощью которых можно проводить не только эмпирический анализ действий фирмы, но и выяснять закономерности, получать статистику, выявлять последовательности действий, приводящих к тому или иному результату [1].

1. Исследование механизма анализа данных

Рассмотрим механизмы анализа данных и прогнозирования подробнее. Общая схема механизма анализа данных представлена на рисунке.

Механизм анализа данных и прогнозирования предоставляет пользователям (экономистам, аналитикам и т.д.) возможность осуществлять поиск неочевидных закономерностей в данных, накопленных в информационной базе. Этот механизм позволяет [2]:

- осуществлять поиск закономерностей в исходных данных информационной базы;
- интерактивно управлять параметрами выполняемого анализа;
- осуществлять программный доступ к результату анализа;
- автоматически выводить результат анализа в табличный документ;
- создавать модели прогноза, позволяющие автоматически прогнозировать последующие события или значения неких характеристик новых объектов.

Механизм анализа данных представляет собой набор взаимодействующих друг с другом объектов встроенного языка, что позволяет разработчику использовать его составные части в произвольной комбинации в любом прикладном решении. Встроенные объекты позволяют легко организовать интерактивную настройку параметров анализа пользователем, а также позволяют выводить результат анализа в удобной для отображения форме в табличный документ.



Рисунок. Общая схема механизма анализа данных и прогнозирования

Механизм позволяет работать как с данными, полученными из информационной базы, так и с данными, полученными из внешнего источника, предварительно загруженными в таблицу значений или табличный документ:

Применяя к исходным данным один из видов анализа, можно получить результат анализа. Результат анализа представляет собой некую модель поведения данных. Результат анализа может быть отображен в итоговом документе, или сохранен для дальнейшего использования.

Дальнейшее использование результата анализа заключается в том, что на его основе может быть создана модель прогноза, позволяющая прогнозировать поведение новых данных в соответствии с имеющейся моделью.

В настоящее время механизм анализа данных и прогнозирования содержит пять типов анализа данных: общая статистика, поиск последовательностей, поиск ассоциаций, кластеризация и дерево решений.

2. Общая статистика

Общая статистика представляет собой механизм для сбора общих сведений об исследуемой выборке. Этот тип анализа предназначен для предварительного исследования какого-то источника информации, позволяет получать статистическую информацию об имеющихся данных. Для непрерывных (тип «Число» или «Дата») и дискретных (все другие типы) полей рассчитываются следующие показатели: количество, максимум, медиана, минимум, размах, среднее и стандартное отклонение. Кроме того, для полей с дискретными типами данных вычисляются количество уникальных значений и мода, а также таблицы частот и значений.

3. Поиск последовательностей

Анализ вида «Поиск последовательностей» позволяет выявлять в источнике данных последовательные, часто возникающие цепочки событий. Например, проанализировав заказы клиентов, можно выявить их последовательную взаимосвязь и на основании этого прогнозировать складские запасы, проводить рекламные акции и даже предлагать товары и услуги с опережением, если результаты проведенного анализа покажут высокую вероятность спроса на них в будущем.

4. Поиск ассоциаций

Метод поиска ассоциаций предназначен для выявления устойчивых комбинаций элементов в определенных событиях или объектах. Результаты анализа представляются подсистемой в виде групп ассоциированных элементов.

Первоначально метод был разработан для поиска типичных сочетаний товаров в покупках, поэтому иногда его еще называют анализом покупательской корзины. Применительно к этому сценарию в качестве ассоциируемых элементов, как правило, выступают товарные группы или отдельные товары. А группирующим объектом, объединяющим элементы выборки, может быть любой объект информационной системы, идентифицирующий сделку, например заказ покупателя, акт об оказании услуг или кассовый чек.

Реализованный в платформе алгоритм поиска ассоциаций имеет достаточно гибкие средства управления адекватностью моделей анализа или прогноза. Параметр «Минимальный процент случаев» определяет «порог срабатывания» алгоритма на ту или иную комбинацию элементов в событии или объекте, что позволяет не учитывать слабо распространенные ассоциации. Параметр «Минимальная достоверность» определяет требуемую устойчивость искомых ассоциаций, а параметр «Минимальная значимость» позволяет выявить наиболее приоритетные из них.

Существенно облегчает восприятие результатов анализа и прогноза параметр «Тип отсечения правил», который может принимать значения «Отсекать избыточные» и «Отсекать покрытые другими правилами».

5. Кластеризация

Цель кластеризации — выделение из множества объектов одной природы некоторого количества относительно однородных групп — сегментов или кластеров. Объекты распределяются по группам так, чтобы внутригрупповые отличия были минимальными, а межгрупповые отличия максимальными. Методы кластеризации позволяют перейти от объектного к групповому представлению совокупности произвольных объектов, что существенно упрощает возможность ими оперировать.

Способы формирования кластеров на основе информации о расстоянии между кластеризуемыми объектами определяются методом кластеризации. Любой из реализованных в платформе методов кластеризации предполагает явное указание количества искомым кластеров. Для атрибутов объектов можно вводить веса, что позволяет расставлять приоритеты между ними.

Результатами анализа с помощью кластеризации являются:

- центры кластеров, представляющие собой совокупность усредненных значений входных колонок в каждом кластере;
- таблица межкластерных расстояний (между центрами кластеров), определяющих степень различия между ними;
- значения прогнозных колонок для каждого кластера;
- рейтинг факторов и дерево условий, определивших распределение объектов на кластеры.

Алгоритмы кластеризации позволяют не только провести кластерный анализ объектов на множестве заданных атрибутов, но и спрогнозировать значение одного или нескольких из них для актуальной выборки на основании отнесения объектов этой выборки к тому или иному кластеру.

6. Дерево решений

В результате применения метода Дерева решений к исходным данным создается иерархическая (древовидная) структура правил вида «Если... то...», а алгоритм анализа обеспечивает процесс вычленения на каждом этапе наиболее значимых условий и переходов между ними. Данный алгоритм получил наибольшее распространение при выявлении причинно-следственных связей в данных и описании поведенческих моделей. Типичная зона применимости деревьев решений — оценка различных рисков, например закрытия заказа клиентом или его перехода к конкуренту, несвоевременной поставки товара поставщиком или

просрочки оплаты товарного кредита. В качестве типичных входных факторов модели выступают сумма и состав заказа, текущее сальдо взаиморасчетов, кредитный лимит, процент предоплаты, условия поставки и иные параметры, характеризующие объект прогноза. Адекватная оценка рисков обеспечивает принятие информированных решений по оптимизации отношения доходность/риск в деятельности компании, а также полезна для увеличения реалистичности различных бюджетов.

На основе почти любого из имеющихся механизмов анализа данных в системе 1С:Предприятие возможно построить некоторую модель прогноза. Естественно, что точность прогнозирования, как и характер и функциональность, полностью зависят от степени полноты и целостности заложенных в модель прогноза данных.

7. Модель прогноза

Модели прогноза, создаваемые механизмом, представляют собой специальные объекты, которые создаются из результата анализа данных и позволяют в дальнейшем автоматически выполнять прогноз для новых данных.

Модель прогноза обладает определенными свойствами, с помощью которых можно управлять проводимым прогнозом.

Создание модели прогноза доступно для всех типов анализа за исключением общей статистики.

Заключение

В ходе работы над статьей был рассмотрен механизм анализа данных и прогнозирования. В настоящий момент этот механизм почти не используется в типовых конфигурациях фирмы 1С, хотя использование этого механизма может существенно сократить время на анализ некоторых ситуаций на предприятиях, а также даст некоторые новые возможности.

Примеры использования механизма Анализа данных, которые можно было бы включить в некоторые типовые конфигурации 1С:Предприятия:

1. Тип анализа «Поиск последовательностей» или «Поиск ассоциаций» можно использовать для предложения клиенту нескольких разновидностей наборов продуктов, которые приобретаются с выбранным пользователем товаром

2. Тип анализа «Кластерный анализ» можно применять для анализа платежеспособности клиентов, или для разбиения контрагентов по группам с разными ценовыми категориями.

3. Тип анализа «Дерево решений» можно использовать в качестве анализа действий менеджеров при работе с клиентами, после чего либо заключалась сделка, либо разрывался контракт. Также можно

анализировать причины и последствия похожих наборов данных

4. Модель прогноза можно использовать для получения достоверного прогноза на будущее по уже имеющимся данным анализа

Список литературы

1. Радченко М.Г. 1С:Предприятие 8.3 : практическое пособие разработчика : примеры и типовые приемы / М.Г. Радченко. – М. : Издательство ООО "1С-Публишинг", 2013. – 965 с.

2. Габец А.П. Профессиональная разработка в системе 1С : Предприятие 8 / А.П. Габец, Д.И. Гончаров, Д.В. Козырев. – М. : Издательство "1С-Публишинг", 2013. – 398 с.

АНАЛИЗ И МОДЕЛИРОВАНИЕ ПРОТОКОЛОВ КВАНТОВОЙ КРИПТОГРАФИИ

Э.И. Шипулина
Студент

С.А. Запрягаев
Профессор

Введение

Одним из направлений практического применения квантовой теории информации является квантовая криптография.

Задача классической криптографии заключается в передаче информации между двумя абонентами с обеспечением достаточной криптостойкости зашифрованной информации. Современные методы классической криптографии не позволяют полностью решить проблему защиты передаваемой информации. Криптостойкость наиболее распространенного алгоритма RSA основывается на вычислительной сложности задачи факторизации больших целых чисел. Однако, теоретически, эта задача решается с помощью квантового алгоритма Шора за приемлемое время, что демонстрирует уязвимость такой системы шифрования. На практике применение классических алгоритмов сопряжено еще с одной трудностью: реализации защищенной технологии распределения ключа шифрования.

Квантовая теория информации открывает новые перспективы в создании «абсолютно» защищенных каналов связи, обеспечивающих полную защиту передаваемой информации. За последние годы квантовая криптография прошла путь от теоретической идеи до реальной системы квантового распределения ключей, получившей достаточное применение.

© Шипулина Э.И., Запрягаев С.А., 2014

Среди пользователей квантовых криптографических систем можно выделить службы безопасности, отдельные министерства и крупные банковские объединения. В настоящее время прототип квантовой криптографической системы реализован даже в виде приложения на мобильном устройстве. Массовое применение квантовых криптографических систем ограничивает относительно высокая стоимость их реализации для крупных корпоративных систем. Успешно внедренные квантовые криптографические системы позволяют распределять ключи по квантовому каналу связи на расстояние более 100 км с приемлемыми для решения поставленных задач скоростями. Для реализации квантового канала связи, способного осуществлять передачу криптограмм на такие расстояния, наиболее подходящей средой является оптическое волокно.

1. Протоколы квантовой криптографии

В настоящее время в квантовой криптографии выделилось направление, связанное с развитием систем распределения ключей, основанных на эффекте квантового запутывания состояний. Запутанными состояниями являются состояния составной системы, которые невозможно представить в виде тензорного произведения одночастичных состояний. Проявление квантового запутывания состояний состоит в том, что измерение лишь части составного квантового состояния фиксирует все состояние в целом. Основным протоколом данного направления является протокол E91.

В настоящей работе выполнено моделирование протоколов квантовой криптографии BB84 и B92 [1-3], представляющих решение задачи по реализации системы «абсолютно» защищенной технологии распределения классических ключей с применением квантового и классического каналов связи. Задача протоколов состоит в создании одноразового случайного классического ключа для осуществления кодирования передаваемой информации по классическому каналу связи (аналог одноразового шифроблокнота). Теоретически квантовый канал связи передает последовательность кубит, каждый из которых представляет собой суперпозицию двух квантовых состояний.

Носителями квантовой информации в протоколе BB84 являются фотоны, поляризованные под углами 0° , 45° , 90° и 135° . Поляризованные под углами 0° и 90° фотоны кодируют 0 и 1 в прямолинейном базисе, который можно обозначить как «+». А поляризованные под углами 45° и 135° фотоны кодируют 0 и 1 в диагональном базисе «×». Базисы расположены друг относительно друга под углом 45° . Внутри обоих базисов состояния ортогональны, но состояния из разных базисов являются попарно неортогональными, что обеспечивает возможность детектирования попыток вмешательства в сеанс связи.

В описании криптографических протоколов легитимных абонентов принято называть Алисой и Бобом, а злоумышленника – Евой. Алиса посылает Бобу по квантовому каналу связи последовательность поляризованных фотонов, случайным образом задавая их поляризацию и фиксируя примененную последовательность состояний поляризатора. Боб случайным образом выбирает ориентацию анализатора и измеряет поляризацию получаемых фотонов (для распознавания поляризации под углами 0° и 90° необходима прямолинейная ориентация поляризатора «+», а для 45° и 135° – диагональная «×»). Затем по открытому каналу связи Боб сообщает Алисе, какой тип анализатора он использовал для каждого из принятых фотонов, а Алиса указывает, какие из них были выбраны верно. Отобрав только те значения, для которых Боб верно выбрал тип анализатора, стороны получают последовательность, из которой половина бит используется для проверки канала связи, а вторая половина формирует уникальный случайный ключ.

Если во время сеанса генерации ключа в квантовый канал связи внедрится злоумышленник Ева и будет считывать случайными анализаторами последовательность фотонов, передаваемых Алисой, то производимые над этими фотонами измерения разрушат их исходные поляризационные состояния, что впоследствии обнаружится сторонами при сравнении состояний фотонов, соответствующих верно выбранным Бобом типам анализатора. В реализованной в настоящей работе программной оболочке для проверки на защищенность квантового канала связи сравниваются все верно выбранные значения, а ключ формируется из каждого второго.

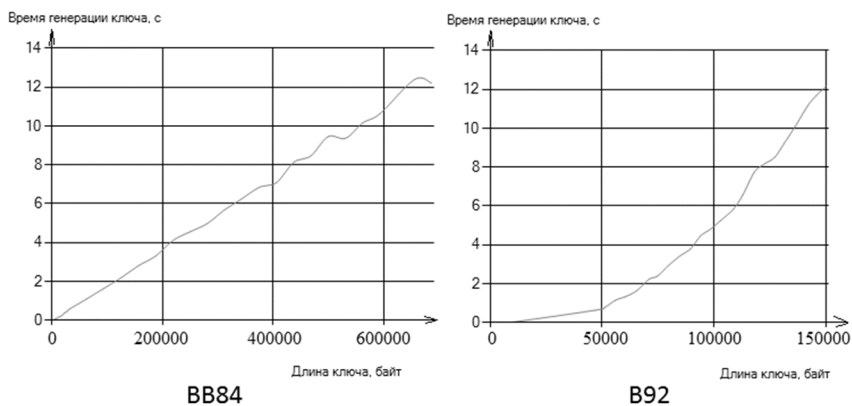
Помимо протокола BB84 был смоделирован и протокол B92. В этом случае у каждого из абонентов есть специальное квантовое устройство: у Алисы – для преобразования исходной случайной бинарной последовательности в совокупность кубит по определенному правилу, а у Боба – для преобразования полученных им кубит в соответствии с операторами проектирования. Квантовая машина Алисы производит неортогональные состояния $|0\rangle$ и $|q_+\rangle = (|0\rangle + |1\rangle)/\sqrt{2}$, а квантовое устройство Боба проектирует эти кубиты на пару неортогональных состояний $|1\rangle$ и $|q_-\rangle = (|0\rangle - |1\rangle)/\sqrt{2}$. В остальном технология формирования случайного ключа аналогична протоколу BB84.

2. Программная реализация

Программный комплекс, моделирующий работу выбранных квантовых протоколов распределения ключей, написан на языке программирования C# под платформу .NET. Разработанную в ходе

настоящей работы программную оболочку можно использовать для изучения, тестирования, анализа и сравнения двух наиболее распространенных протоколов квантовой криптографии BB84 и B92 [4].

В процессе моделирования системы формирования уникального случайного ключа в соответствии с протоколами BB84 и B92 был разработан эффективный алгоритм, который позволяет за тысячные доли секунды сгенерировать уникальный случайный ключ длиной около 5 тысяч бит. Наличие злоумышленника в квантовом канале связи определяется в процессе формирования ключа, что приводит к прекращению сеанса связи. Для анализа затрачиваемого времени была произведена оценка зависимости среднего времени генерации ключа от его длины (рис. 1а, 1б).



а) протокол BB84; б) протокол B92

Рис. 1. Зависимость времени генерации ключа от его длины

На графике по результатам протокола BB84 на выделенном начальном диапазоне длин ключа четко прослеживается линейная зависимость, а на графике по протоколу B92 – экспоненциальная. За 4 секунды по протоколу BB84 удастся сгенерировать ключ длиной 200 тыс. байт (или 195,3 килобайт), а по протоколу B92 около 100 тыс. байт. Но это не означает, что реализованный алгоритм протокола B92 в два раза менее эффективен. Причиной такого расхождения является то, что в протоколе BB84 число случаев, в которых выбранные Бобом базисы совпали, составляет в среднем половину исходной последовательности, а в протоколе B92 – в два раза меньше: от длины исходной последовательности.

Помимо этого в работе проведен анализ влияния шума на квантовый

канал связи. Суть одного эксперимента заключалась в том, что в исходную последовательность фотонов, которую Алиса передает Бобу, случайным образом вносилось одно искажение. На основании результатов проведенных экспериментов выявлена средняя вероятность влияния одного искажения на результирующий ключ. Ее величина около 0,25.

Также оценено влияние различного количества искажений, изменяющегося с шагом в 1, при фиксированной длине исходной последовательности на итоговый ключ. Результирующий график зависимости отношения количества экспериментов, в результате которых ключ изменился, к общему числу экспериментов от количества искажений представлен на рис. 2.

Отношение кол-ва экспериментов, в рез-те кот. ключ изменился, к общему числу экспериментов

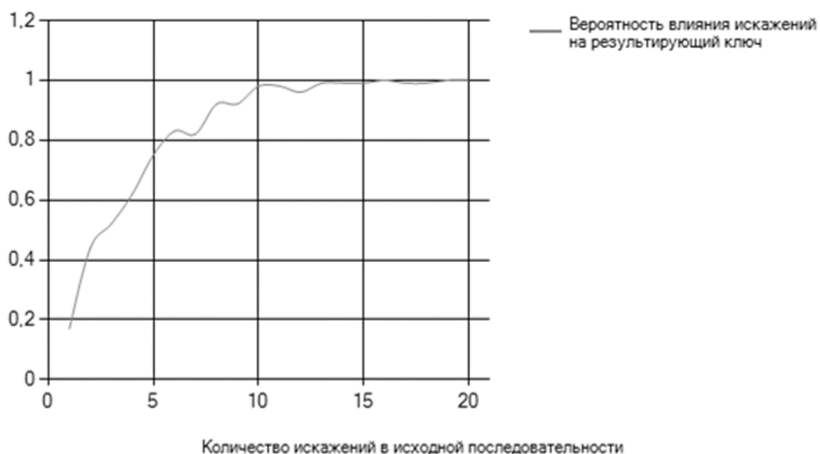


Рис. 2. Вероятность влияния искажения на результирующий ключ

Заключение

В результате в настоящей работе был решен ряд научно-технических задач, совокупность которых определяла цель работы: моделирование защищенной системы распределения квантовых криптографических ключей с применением моделей квантового и классического каналов связи. Смоделирована работа двух известных протоколов квантовой криптографии, основанных на кодировании квантового состояния одиночной частицы: BB84 и B92. Для выявления попытки внедрения злоумышленника был реализован алгоритм проверки на защищенность квантового канала связи. Для сравнения скорости работы двух выбранных протоколов была произведена оценка времени, затрачиваемого на генерацию ключа. Помимо этого было проведено

моделирование процесса зашумления квантового канала связи. На основании проведенных экспериментов с целью выявления характера изменения ключа при зашумлении канала была определена вероятность влияния искажений на результат.

Квантовая криптография заняла достойное место среди систем, обеспечивающих конфиденциальную передачу информации, и интерес к технологиям ее реализации продолжает возрастать. Научный мир, находясь в поиске наиболее удачных решений, обеспечивающих приемлемую для поставленных задач дальность связи, скорость и защищенность передачи данных, обратился к квантовой криптографии, как к одному из оптимальных и эффективных решений и продолжает исследования в данной области, развивая теорию квантовой криптографии и открывая новые интересные решения в этой области.

Список литературы

1. Bennett C.H. Quantum Cryptography: Public Key Distribution and Coin Tossing / C.H. Bennett, G. Brassard // Proc. of IEEE Int. Conf. on Comput. Sys. and Sign. Proces. – 1984. – P. 175–179.
2. Bennett C.H. Quantum Cryptography using any two Nonorthogonal States/ C.H. Bennett // Phys. Rev. Lett. –1992. – Vol. 68. – P. 3121.
3. Килин С.Я. Квантовая информация / С.Я. Килин // Успехи физических наук. – 1999. – Т. 169. – № 5. – С. 514–516.
4. Шипулина Э.И. Анализ и моделирование протоколов квантовой криптографии / Э.И. Шипулина, С.А. Запругаев // Информатика: проблемы, методология, технологии. Материалы XIV Международной научно-методической конференции. Воронеж : Воронежский государственный университет, 6-8 февраля 2014. – Воронеж : ВГУ, 2014 – Т. 3. – С. 307–311.

АНАЛИЗ БИЗНЕС-ПРОЦЕССОВ ПРЕДПРИЯТИЯ И НАСТРОЙКА СИСТЕМЫ АВТОМАТИЗИРОВАННОГО ВЕДЕНИЯ СКЛАДСКОГО УЧЕТА СРЕДСТВАМИ SAP ERP

Д.В. Шульгин
Студент

И.В. Илларионов
Доцент

Введение

Развитие промышленного производства продолжается по пути роста разнообразия производимой продукции, усложнения технологических

© Шульгин Д.В., Илларионов И.В., 2014

процессов и процедур управления. Сегодня ни одно предприятие не может обойтись без использования информационных технологий для поддержки выполняемых управленческих процедур. Эти процедуры выполняются в автоматизированных системах управления – ERP. Системы ERP имеют модульную структуру [1], в которой выделяют подсистемы планирования, логистики, управления производством, финансами, персоналом и др.

Автоматизация каждого из этих модулей помогает добиться ускорения процессов планирования, управления, анализа. От скорости принятия тех или иных решений зависит успех и эффективность бизнеса в целом, как и от времени реагирования на те или иные обстоятельства.

1. SAP WMS (Warehouse Management System)

Отдельное место в ведении бизнеса играет складской учёт [2]. Торговля и склад любого предприятия подразумевает склад товаров и продукции любого плана: одежды, материалов и так далее, которые требуют контроля. Учет на складе без специального программного обеспечения становится весьма затруднительным и неточным из-за сильно присутствующего человеческого фактора. Поэтому организация эффективной работы на крупном складе просто невозможна без использования автоматизированных систем управления класса WMS (Warehouse Management System, система управления складами).

Использование решений SAP для управления складами (SAP WMS) позволяет оптимизировать складскую деятельность, включая обработку входящей и исходящей документации, управление объектами и хранение, инвентаризацию, а также планируемое и условное сквозное складирование. Решение SAP WMS позволяет описать физическую топологию склада и в системе реального времени управлять складскими процессами. Гибкость системы дает возможность оперативно изменять структуру складского комплекса, удовлетворяя новые требования рынка: менять конфигурацию склада, менять стратегию поиска и раскладки запасов по складским местам.

2. Цель эксперимента

Цель эксперимента в данной статье – анализ бизнес-процессов предприятия и настройка системы Управления складами SAP WMS с учётом всех особенностей.

В качестве исходной была взята компания ООО «СтудСтрой». Данная компания занимается строительством и обустройством хозяйских участков. Компания оказывает свои услуги в рамках определённого региона, в данном случае – Воронежской области, по всей территории которой находятся отделения (Воронеж, Рамонь, Усмань, Новая Усмань,

Лиски, Россось) – подразделения, готовые заняться своими обязанностями на близлежащих участках. У компании имеется центральный складской комплекс, в котором хранится основная часть сырья, инвентаря, спецодежды и инструментов, а также склады при каждом из подразделений, куда запасы приходят с центрального склада непосредственно перед использованием. Компания ООО "СтудСтрой" основана на реальной компании с реальными бизнес-процессами.

Для ведения складского учёта необходимо решение SAP WMS с реализацией основных бизнес-процессов и операций. Так как процессы МТО и СУС тесно связаны между собой, то была определена необходимость частично провести настройку таких блоков как Закупки и Сбыт.

3. Определение, анализ и оптимизация бизнес-процессов

Для начала было необходимо построить модель EPC для существующих бизнес-процессов [3]. После этого было проведено имитационное моделирование [4], которое позволило увидеть слабые стороны существующей системы складского учета. В рамках затрачиваемого времени и удобства данные бизнес-процессы были оптимизированы и изменены с последующими имитациями в качестве проверки. В итоге были выделены и определены следующие бизнес-процессы, требующие дальнейшей настройки в системе SAP WMS:

- Поступление МТР
- Хранение МТР
- Инвентаризация
- Отпуск МТР

4. Настройка орг. структуры предприятия

После определения выполняемых функций, было решено приступить к настройке в соответствии с ними решения SAP. Первоначально была выполнена общая настройка предприятия: определены его организационные уровни и единицы, закупочные организации и рынки сбыта, определены связи между ними [3]. Однако подробнее о настройке самой системы управления складами (СУС).

5. Настройка системы управления складами

После настройки основной структуры была проведена настройка Системы управления складами. Для начала была определена орг. структура (рис. 1).

Центральный склад представляет собой складской комплекс – номер склада 940, который включает в себя типы складов – Крытый склад 101 и открытую площадку 201. Каждый из типов складов в свою очередь делится на складские участки по признаку материала: Спецодежда, Инструменты,

Хрупкие материалы, Крупногабаритные запасы, Вспомогательные материалы, Тара и тарные материалы.

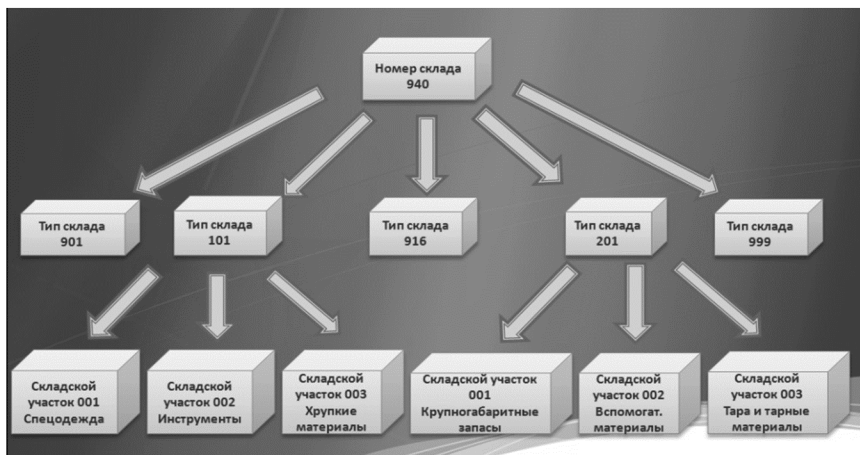


Рис 1. Орг. структура ООО "СтудСтрой".

Настройка орг. структуры СУС в системе SAP включает в себя следующие пункты:

- Определение номера склада;
- Определение типов склада;
- Определение складских участков;
- Определение типов складских мест;
- Определение типов единиц складирования;
- Определение структуры складских мест;
- Определение промежуточных складских мест.

После настройки структуры проводилась настройка основных процессов, определяющих функционал, стратегии, методы:

- Определение кодов для типов склада и складских участков;
- Активация поиска типа склада;
- Активация поиска складского участка;
- Активация поиска складского места;
- Определение видов движения;
- Присвоение видов движения СУС ссылочным видам движения;
- Определение значений по умолчанию и видов инвентаризации.

Меню для выполнения настройки орг. структуры и складской логистики в системе SAP показано на рис 2.

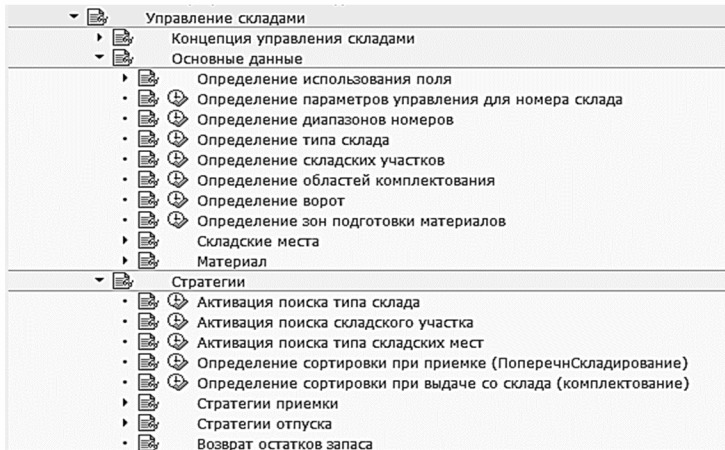


Рис. 2. Меню настройки SAP WMS

Заключение

Данная статья посвящена ERP системам класса WMS, а также их влияние на эффективность бизнеса. Описано создание решения SAP WMS для деятельности склада конкретного предприятия и процесс подготовки к реализации бизнес-процессов. В итоге данного эксперимента получилась полностью рабочая система управления складом с автоматизированными процессами поступления, размещения, хранения, отпуска МТР с возможностью проведения инвентаризации и мониторинга. Использование решения SAP WMS позволяет решать задачи складского учета с минимальными временными и умственными затратами, что при условии постоянного роста конкуренции является несомненным преимуществом.

Список литературы

1. SAP R/3: менеджмент : пер. с нем. / под ред. М. Ребштока и К. Хильдебранда ; науч. ред. Н. П. Беляцкий. – Минск : Новое знание, 2001. – 207 с.
2. Thomas J. Supply Chain Management with APO / J. Thomas. – Berlin : Springer, 2003. – 250 p.
3. Sharma K. Configuring SAP ERP Sales and Distribution 2010 / K. Sharma, A. Mutsaddi. – New York : Wiley, 2010. – 672 p.
4. Схиртладзе А.Г. Проектирование единого информационного пространства виртуальных предприятий / А.Г. Схиртладзе, А.В. Скворцов, Д.А. Чмырь. – М. : Абрис, 2012. – 615 с.

РЕАЛИЗАЦИЯ ОХРАННОЙ СИСТЕМЫ НА КОНТРОЛЛЕРЕ NETDUINO PLUS С ИСПОЛЬЗОВАНИЕМ GSM И LAN КАНАЛОВ ДЛЯ ПЕРЕДАЧИ ДАННЫХ

П.А. Щеголеватых
Студент

Д.Н. Борисов
Доцент

Введение

Благодаря бурно развивающейся полупроводниковой индустрии все цифровые составляющие полноценного компьютера могут быть объединены в один микрочип – микроконтроллер. Микроконтроллер с 32-битным процессором на сегодняшний день стоит менее чем \$10, может иметь в два раза больше памяти, чем оригинальный 8-битный компьютер Apple II с его 48КВ памяти, и может работать в 100 раз быстрее.

Такие факторы, как небольшой размер микросхем при достаточной мощности процессора, увеличение объемов памяти, встроенная поддержка стандартных протоколов взаимодействия (таких как TCP/IP, GSM и др.), низкое энергопотребление, сравнительно низкая цена и возрастающая популярность делают актуальной проблему разработки различного рода систем, рассчитанных на частное использование.

Целью данной работы является разработка охранной системы на контроллере Netduino Plus, которая оснащена инфракрасным датчиком движения и позволяет отправлять сообщения через GSM и LAN каналы передачи данных. Также необходимо обеспечить «адекватную» реакцию системы на различного рода неполадки. Например, при механических повреждениях или отказах некоторых составляющих модулей система должна продолжать свою работу с некоторыми ограничениями функциональности и восстанавливать исходное состояние при устранении неполадки.

Практическая новизна данной работы заключается в том, что на момент начала разработки в продаже не существовало охранных устройств, способных использовать LAN и GSM каналы передачи данных с промежуточным хранением очереди сообщений на встроенной флэш карте в случае различных неполадок. Большинство коммерческих GSM устройств сигнализации не устойчивы к подавлению GSM сигнала. Это справедливо и для коммерческих WiFi устройств. Данная система может быть использована для охраны различных частных помещений, например для дома и дачи.

1. Платформа .NET Micro Framework

Microsoft .NET Micro Framework [1] это платформа для запуска управляемого кода на маленьких встраиваемых устройствах, которые ограничены по ресурсам настолько, что не могут позволить себе запуск Windows CE и .NET Compact Framework, не говоря уже о полноценной версии Windows с конкретной версией .NET Framework.

.NET Micro Framework позволяет разрабатывать приложения для встраиваемых устройств, используя для этого язык высокого уровня C# и мощные среды разработки, такие как Visual Studio для Windows или Mono для Linux и Mac OS X. Это позволяет без больших затрат портировать приложения, используя .NET Compact Framework или даже .NET Framework под эти устройства. Также эта платформа предоставляет расширяемый эмулятор нужного железа для быстрого прототипирования и отладки.

.NET Micro Framework не требует предустановленной операционной системы. Уменьшенная версия Common Language Runtime (TinyCLR) работает напрямую с железом. .NET Micro Framework абстрагирует программиста от низкоуровневых операций с железом используя встроенную библиотеку классов (FCL) и позволяет работать с компонентами железа как с объектами.

.NET Micro Framework это самая маленькая из доступных на сегодняшний день .NET платформ по своим возможностям и предоставляет только подмножество версий .NET Framework и .NET Compact Framework. Также платформа требует 32-битный процессор. Хотя часто при написании программ для встраиваемых систем на языке ассемблера или C все еще используются 16-битные или даже 8-битные процессоры, улучшенная техника производства смогла уменьшить фабричную стоимость и требования к электроэнергии 32-битных процессоров. Несмотря на это, факт, что .NET Micro Framework не требует наличия MMU значит, что она может работать на менее дорогих процессорах чем, скажем, Windows CE.

Несмотря на все преимущества, .NET Micro Framework это не система, работающая в реальном времени. Например, событие таймера может не сработать точно после заданного времени (может отличаться на несколько миллисекунд) или среда выполнения может взять несколько секунд на «размышление», прежде чем среагировать на прерывание спровоцированное сервисом обработки прерываний.

2. Реализация охранной системы

Проект состоит из трех основных компонентов – самого устройства Netduino Plus [2, 3], инфракрасного датчика движения PIR SEN-08630 [4] и GSM модуля Seeed GPRS Shield V1.0 SLD33149P. Устройство Netduino

Plus подключается к роутеру и получает IP адрес по DHCP протоколу. Если роутер не поддерживает DHCP, то есть альтернативная возможность использования статических IP адресов, в этом случае каждый раз при смене конфигурации сети (например, при смене поставщика услуг) придется перенастраивать устройство вручную. Внешний вид прототипа системы представлен на рис. 1.

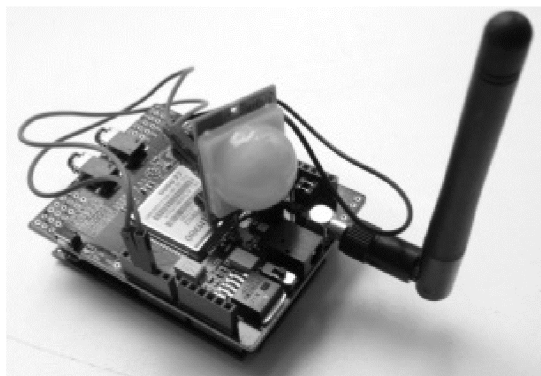


Рис. 1. Прототип системы Secure Room

Контроллер находится в охраняемом помещении и взаимодействует с датчиком движения, получая от него прерывания при обнаружении какого-либо движения в радиусе обзора датчика. Интервал обработки прерываний задается в программном коде устройства. При получении прерывания устройство отправляет заданный текст сообщения с указанием точной даты и времени на заданный в программном коде адрес (список адресов) электронной почты используя LAN канал передачи данных, а также передает SMS сообщение по GSM каналу на заданный номер (список номеров).

В случае недоступности одного или нескольких каналов (например, при использовании так называемых «GSM глушилок», которые заглушают сигнал GSM антенны, либо при отключении LAN сети со стороны провайдера, либо два таких случая одновременно) неотправленные сообщения сохраняются на microSD карте устройства для повторной отправки через заданный в программном коде временной интервал (например 60 секунд).

Сохранение неотправленных сообщений на карте также происходит в случае всякого рода «исключительных ситуаций» во время работы устройства. Например, недоступность почтового сервера, повреждение сетевого кабеля, механические повреждения GSM антенны или SIM карты и т.п. Эти исключительные ситуации обрабатываются автоматически. Все

неотправленные сообщения постепенно будут отправлены после исправления неполадки, даже если устройство будет выключено из сети или перезагружено. Неотправленные сообщения будут храниться на карте в *.CSV формате, что позволит при необходимости прочитать их при извлечении карты в любом текстовом редакторе.

На рис. 2 представлена блок-схема процесса загрузки системы.

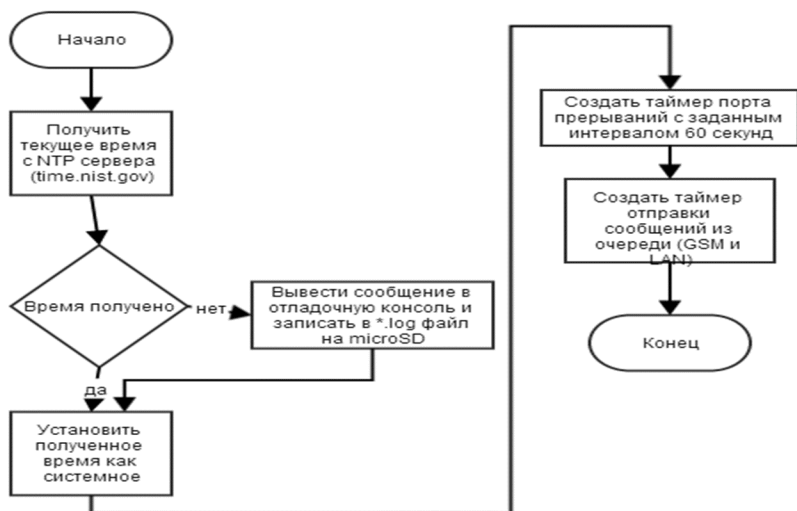


Рис. 2. Блок-схема процесса загрузки системы

При включении устройства сначала происходит установка системного времени. Оно берется с заданного в настройках NTP сервера. Далее создаются два таймера. Первый отвечает за управление портом прерывания PIR датчика движения, второй отвечает за рассылку сообщений в очереди на microSD карте.

Основной процесс отправки сообщений представлен на блок-схеме (рис. 3).

При поступлении прерывания с датчика система отправляет сообщение последовательно по GSM и LAN каналам в виде SMS и Email соответственно. Если отправка не удастся (возникает ошибка при отправке), то сообщение добавляется в очередь.

Наконец процесс отправки сообщений из очереди представлен на блок-схеме (рис. 4).

Из схемы видно, что обработка очередей идет последовательно и в случае успешной отправки эти сообщение удаляется из очереди. Каждое сообщение имеет уникальный идентификатор в системе (GUID).

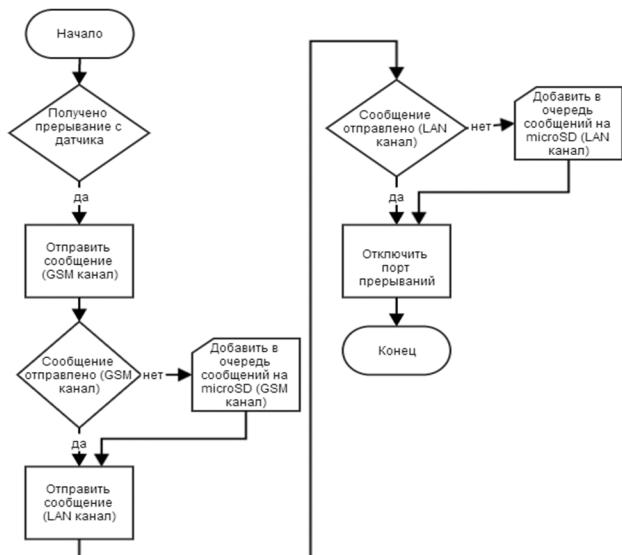


Рис. 3. Блок-схема процесса отправки сообщений

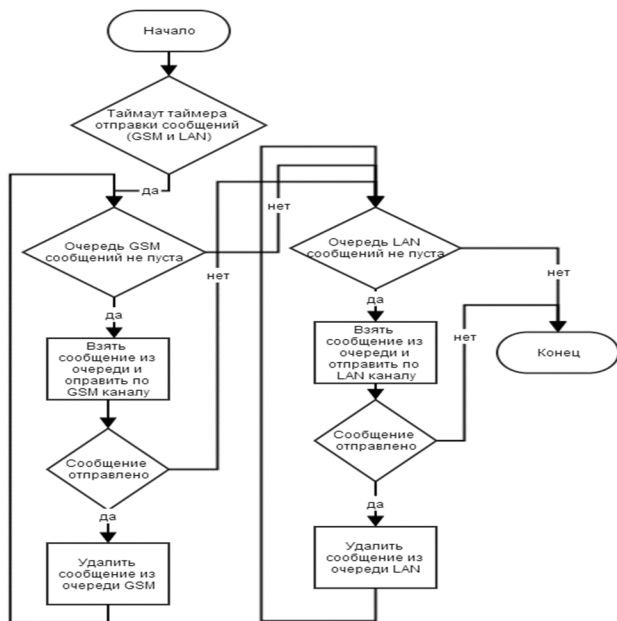


Рис. 4. Блок-схема обработки очереди сообщений

Заключение

Платформа .NET Micro Framework позволяет ускорить процесс разработки за счет наличия встроенной библиотеки классов (FCL) для различных задач, таких как работа с файловой системой, веб, обработка данных в различных структурах (массивы, списки, очереди, стек, деревья). Также одним из главных преимуществ платформы является переносимость программного кода, независимо от типа конкретного устройства (контроллера).

Проанализировав существующие решения стоит отметить, что большинство из них работает только по GSM протоколу и не устойчиво к различного рода «глушилкам» сигналов.

При возникновении различных неполадок в большинстве случаев не требуется перезагрузка контроллера по питанию. Система способна восстанавливаться сама при их устранении.

Возможным улучшением системы в дальнейшем может стать добавление фото или видео камеры для снятия изображений в момент возникновения движений в охраняемом помещении и сохранение этих изображений на microSD карте. Также на текущий момент все настройки такие как текст сообщения, адреса серверов, номера телефонов и т.д. задаются при программировании контроллера перед установкой системы. Их изменение потребует вызова обслуживающего персонала для перепрограммирования устройства. В дальнейшем это можно улучшить, обеспечив возможность хранения настроек на microSD карте с последующей загрузкой их в память при запуске системы.

Список литературы

1. Thompson D. Embedded Programming with the Microsoft .NET Micro Framework / D. Thompson, Rob S. Miles. – Washington : Microsoft Press, 2007. – 257 p.
2. Walker Ch. Getting Started with Netduino / Ch. Walker. – Sebastopol : O'Reilly Media, 2012. – 84 p.
3. Pfister C. Getting Started with the Internet of Things / C. Pfister. – Sebastopol : O'Reilly Media, 2011. – 175 p.
4. PIR Motion Sensor. – Режим доступа: <http://learn.adafruit.com/downloads/pdf/pir-passive-infrared-proximity-motion-sensor.pdf>

СОДЕРЖАНИЕ

<i>Бельий Г.Ю., Родин В.А.</i> ОБ ОДНОЙ МОДИФИКАЦИИ ЗАДАЧИ КОММИВОЯЖЕРА	3
<i>Богоявленская Ю.А., Гаршина В.В.</i> РАЗРАБОТКА СИНТАКСИЧЕСКОГО И СЕМАНТИЧЕСКОГО КОМПОНЕНТОВ ЛИНГВИСТИЧЕСКОГО ПАРСЕРА РУССКОГО ЯЗЫКА.....	6
<i>Божок А.О., Кургалин С.Д., Крыловецкий А.А.</i> МЕТОДЫ И АЛГОРИТМЫ ПОСТРОЕНИЯ ТРЕХМЕРНЫХ МОДЕЛЕЙ В СИСТЕМАХ КОМПЬЮТЕРНОГО ЗРЕНИЯ	9
<i>Бубельник Д.А., Лобода А.В.</i> НЕКОТОРЫЕ ВОПРОСЫ КЛАССИФИКАЦИИ ПЯТИМЕРНЫХ АЛГЕБР ЛИ	14
<i>Воробьева Е.С., Жижелев А.А.</i> ПОСТРОЕНИЕ МОДЕЛИ ПЕРВИЧНОЙ АУТЕНТИФИКАЦИИ ПРИ ПОМОЩИ СИСТЕМЫ «ЭЛЕКТРОННЫЙ СЕКРЕТАРЬ».....	20
<i>Киреевская Е.В., Абрамов И.В.</i> СОЗДАНИЕ ЭФФЕКТИВНОЙ СИСТЕМЫ БЮДЖЕТНОГО УПРАВЛЕНИЯ С ИСПОЛЬЗОВАНИЕМ ПРОЦЕССНОГО ПОДХОДА.....	25
<i>Киселёв Э.Э., Запрягаев С.А.</i> ПРИМЕНЕНИЕ СКРЫТЫХ МОДЕЛЕЙ МАРКОВА ДЛЯ РАСПОЗНАВАНИЯ РУКОПИСНЫХ СИМВОЛОВ.....	29
<i>Климов К.В., Тюкачев Н.А.</i> РЕАЛИЗАЦИЯ АЛГОРИТМА STRUCTURED FOREST ДЛЯ ОБНАРУЖЕНИЯ ГРАНИЦ НА ИЗОБРАЖЕНИЯХ	35
<i>Колесникова Т.В., Илларионов И.В.</i> ПРОЕКТИРОВАНИЕ АРХИТЕКТУРЫ АВТОМАТИЗИРОВАННЫХ ТЕСТОВ ПОЛЬЗОВАТЕЛЬСКОГО ИНТЕРФЕЙСА WEB-ПРИЛОЖЕНИЯ	39
<i>Конотоп М.И., Вялых С.А.</i> ОЦЕНКА НАДЕЖНОСТИ УДОСТОВЕРЯЮЩЕГО ЦЕНТРА	42
<i>Константинова А.П., Гаршина В.В.</i> АВТОМАТИЧЕСКОЕ ФОРМИРОВАНИЕ И ПОСТРОЕНИЕ ОНТОЛОГИЙ ПРЕДМЕТНОЙ ОБЛАСТИ ТЕКСТОВ НА ЕСТЕСТВЕННОМ ЯЗЫКЕ.....	49
<i>Коротких А.А., Сапегин С.В.</i> РАЗРАБОТКА ИНФОРМАЦИОННОЙ СИСТЕМЫ ДЛЯ КООРДИНАЦИИ УЧЕБНЫХ И ИССЛЕДОВАТЕЛЬСКИХ ГРУПП.....	54

<i>Косачев С.Г., Руденко А.В., Атанов А.В.</i> АФФИНО-ОДНОРОДНЫЕ ВЕЩЕСТВЕННЫЕ ГИПЕРПОВЕРХНОСТИ (0,0)-ТИПА В СЗ.....	58
<i>Кочурова Е.Д., Абрамов И.В.</i> BUSINESS INTELLIGENCE И OLAP СЕГОДНЯ.....	61
<i>Лукин Е.О., Нечаев Ю.Б.</i> ИССЛЕДОВАНИЕ МАРШРУТИЗАЦИИ ПРИ РАЗЛИЧНОЙ ПЛОТНОСТИ РАСПОЛОЖЕНИЯ УЗЛОВ.....	67
<i>Мизгаль А.Д., Китаев И.В.</i> РАЗРАБОТКА СИСТЕМЫ ЛОКАЛИЗАЦИИ WI-FI УСТРОЙСТВ В ПОМЕЩЕНИИ.....	75
<i>Мирзай Х., Тюкачев Н.А.</i> МОДИФИКАЦИЯ ДЕТЕКТОРА ОБНАРУЖЕНИЯ КРАЕВ CANNY С ИСПОЛЬЗОВАНИЕМ СТАТИСТКИ И ПРИЗНАК ХААРА	82
<i>Михин П.М., Туровский Я.А.</i> АВТОМАТИЗАЦИЯ ПОИСКА ФЕНОМЕНОВ, СВЯЗАННЫХ С РАЗЛИЧНЫМИ СОСТОЯНИЯМИ ГОЛОВНОГО МОЗГА	92
<i>Михина В.Г., Туровский Я.А.</i> МОДЕЛИРОВАНИЕ И РАСЧЕТ ОПТИМАЛЬНЫХ ПАРАМЕТРОВ ДЛЯ ЭКЗОСКЕЛЕТА ВЕРХНЕЙ КОНЕЧНОСТИ	95
<i>Молодых Д.Н., Сирота А.А.</i> РАЗРАБОТКА МОДЕЛЕЙ И АЛГОРИТМОВ ФОРМИРОВАНИЯ АНАЛИТИЧЕСКОЙ ОТЧЕТНОСТИ В СИСТЕМЕ SAP BW НА ОСНОВЕ СТАТИСТИЧЕСКИХ ДАННЫХ, АГРЕГИРУЕМЫХ В ВИДЕ OLAP-КУБОВ	99
<i>Назарьева В.А., Матвеев М.Г.</i> ПОСТРОЕНИЕ ИМИТАЦИОННОЙ МОДЕЛИ УПРАВЛЕНИЯ ЗАПАСАМИ НА СКЛАДЕ СРЕДСТВАМИ BUSINESS STUDIO	104
<i>Новичихин Д.Ю., Хлебостроев В.Г.</i> ПРИМЕНЕНИЕ МЕТОДА ЧАСТИЦ ДЛЯ МОДЕЛИРОВАНИЯ МЕХАНИЧЕСКОГО ВЗАИМОДЕЙСТВИЯ ТЕЛ.....	110
<i>Нужных А.В., Губарев И.С.</i> ПРОГРАММНО-АППАРАТНЫЙ КОМПЛЕКС АНАЛИТИКИ ПОВЕДЕНИЯ ЛЮДЕЙ НА ОСНОВЕ ТЕХНОЛОГИИ WI-FI	116
<i>Пашенцева А.С., Митрофанова Е.Ю.</i> ONLINE-СЕРВИС ДЛЯ СКРЫТОГО АННОТИРОВАНИЯ ФАЙЛОВ ГРАФИЧЕСКИХ ФОРМАТОВ.....	121
<i>Перелётов И.М., Хлебостроев В.Г.</i> РЕШЕНИЕ ПРОБЛЕМЫ	

РАСПОЗНАВАНИЯ ПЕЧАТНЫХ ТЕКСТОВ С ИСПОЛЬЗОВАНИЕМ СРЕДСТВ БИБЛИОТЕКИ OPENCV	127
<i>Петрушин И.А., Сапегин С.В.</i> СИСТЕМА ИНТЕГРАЦИИ WEB- САЙТОВ РЕКРУТМЕНТА	133
<i>Полухина Н.Ю., Сирота А.А.</i> МАТЕМАТИЧЕСКОЕ И ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ДЛЯ ОБОСНОВАНИЯ СТРУКТУРЫ ИНФОРМАЦИОННОЙ СИСТЕМЫ С ИСПОЛЬЗОВАНИЕМ МЕТОДОВ СИСТЕМНОГО АНАЛИЗА	139
<i>Рагуцкий А.И. Нечаев Ю.Б.</i> МОБИЛЬНЫЕ БЕСПРОВОДНЫЕ СЕТИ ЯЧЕЙСТОЙ ТОПОЛОГИИ.....	148
<i>Ремизова Е.В., Лобода А.В.</i> ИСПОЛЬЗОВАНИЕ КАНОНИЧЕСКИХ УРАВНЕНИЙ В ЗАДАЧЕ ОБ АФФИННОЙ ОДНОРОДНОСТИ.....	155
<i>Решетов А.А., Борисов Д.Н.</i> РЕАЛИЗАЦИЯ ПОВТОРИТЕЛЯ НА БАЗЕ MSP430 С ИСПОЛЬЗОВАНИЕМ ПРОТОКОЛА SIMPLICITY	163
<i>Суровцев А.С., Тюкачев Н.А., Губарев И.С.</i> СИМУЛЯЦИЯ ДВИЖЕНИЯ МЯГКИХ ТЕЛ (ОДЕЖДЫ) В КОМПЬЮТЕРНОЙ ГРАФИКЕ.....	169
<i>Шевляков А.О., Матвеев М.Г.</i> ПОЛЕ НЕЧЕТКИХ ЧИСЕЛ.....	175
<i>Шилов А.С., Другалев А.И.</i> АНАЛИЗ ДАННЫХ И ПРОГНОЗИРОВАНИЕ В 1С: ПРЕДПРИЯТИЕ 8.....	180
<i>Шипулина Э.И., Запрягаев С.А.</i> АНАЛИЗ И МОДЕЛИРОВАНИЕ ПРОТОКОЛОВ КВАНТОВОЙ КРИПТОГРАФИИ	185
<i>Шульгин Д.В., Илларионов И.В.</i> АНАЛИЗ БИЗНЕС-ПРОЦЕССОВ ПРЕДПРИЯТИЯ И НАСТРОЙКА СИСТЕМЫ АВТОМАТИЗИРОВАННОГО ВЕДЕНИЯ СКЛАДСКОГО УЧЕТА СРЕДСТВАМИ SAP ERP.....	190
<i>Щеголеватых П.А., Борисов Д.Н.</i> РЕАЛИЗАЦИЯ ОХРАННОЙ СИСТЕМЫ НА КОНТРОЛЛЕРЕ NETDUINO PLUS С ИСПОЛЬЗОВАНИЕМ GSM И LAN КАНАЛОВ ДЛЯ ПЕРЕДАЧИ ДАННЫХ	195

Научное издание

**СБОРНИК СТУДЕНЧЕСКИХ НАУЧНЫХ РАБОТ
ФАКУЛЬТЕТА КОМПЬЮТЕРНЫХ НАУК ВГУ**

Выпуск 8

Подп. в печ. 17.06.2014. Формат 60x84/16. Усл. печ. л. 11,9. Тираж 80 экз. Заказ 578.

Издательский дом ВГУ.

394000, г. Воронеж, пл. Ленина, 10.

Отпечатано с готового оригинал-макета в типографии

Издательского дома ВГУ.

394000, г. Воронеж, ул. Пушкинская, 3.