Министерство образования и науки РФ

Федеральное государственное автономное образовательное учреждение высшего образования
«Севастопольский государственный университет»

Институт «Информационные технологии и управление в технических системах»
Кафедра «Информационные технологии и компьютерные системы»

СБОРНИК СТАТЕЙ

студенческой научно-технической конференции

"В МИРЕ КОМПЬЮТЕРНЫХ ТЕХНОЛОГИЙ"

г. Севастополь,
4 – 8 апреля 2016 г.

УДК 004

Мир компьютерных технологий: Сборник статей студенческой научно-технической конференции, г.Севастополь, 04 – 08 апреля 2016 г/ М-во образования и науки РФ, Севастопольский государственный университет; науч. ред. Е.Н. Мащенко – г. Севастополь: СевГУ, 2016. – 241 с.
Режим доступа: http://lib.sevsu.ru/jspui/handle/123456789/7894 — Загл. с титул. экрана.

В сборник входят статьи по материалам докладов студентов и магистрантов–участников студенческой научно-технической конференции (4-8 апреля 2016г.), по направлениям: «Современные направления моделирования в компьютерной инженерии, образовании и науке», «Робототехника. Технологии искусственного интеллекта», «Интеллектуальный анализ данных. Технологии Big Data. Системы поддержки принятия решений», «Облачные и Грид-технологии», «Проектирование аппаратных средств вычислительной техники», «Компьютерные сети», «Web – разработка», «Информационная безопасность», «Системное и прикладное программное обеспечение», «Геоинформационные системы и технологии».
Издание рассчитано на ученых, аспирантов и студентов.

Сборник подготовлен к изданию
кафедрой «Информационные технологии и компьютерные системы»
ФГАОУ ВО «Севастопольский государственный университет».

Научное электронное издание

ОГЛАВЛЕНИЕ

СЕКЦИЯ 1

Современные направления моделирования в компьютерной инженерии, образовании и науке

УДК 004.415
К.В. Белая, магистрант
Научный руководитель: В.И. Шевченко, к.т.н., доцент
Севастопольский государственный университет

МОДЕЛИРОВАНИЕ РАБОЧЕГО ПОВЕДЕНИЯ УДАЛЁННЫХ СОТРУДНИКОВ С ИСПОЛЬЗОВАНИЕМ ИНТЕЛЛЕКТУАЛЬНЫХ АГЕНТОВ

Аннотация: предложен подход для решения задачи моделирования процессов взаимодействия удаленных сотрудников. Разработана мультиагентная модель управления поведением сотрудников.
Ключевые слова: агент, мультиагентная система, удаленное сотрудничество.
Annotation: the approach for the solution of the problem of modeling the interaction of remote employees was offered. A multi-agent model for the management of employee behavior was developed.
Key words: agent, multi-agent system, remote employment.

Введение

В настоящее время в области корпоративного менеджмента большое внимание уделяется такому сравнительно новому явлению, как распределённые офисы [1]. Российские компании осознали, что телекоммуникационная инфраструктура может стать мощным оружием в конкурентной борьбе и стремятся опередить своих соперников по уровню внедрения новых решений. На эти цели предприятия готовы тратить весьма значительные средства − создание собственных сетей становится непременным условием продвижения бизнеса. Сейчас тяжело себе представить современно развивающуюся компанию с одним головным офисом. Для успешного развития современным компаниям необходимы информационные системы, которые будут работать в режиме реального времени и связывать все филиалы (ими могут быть распределенные офисы, розничные точки продаж, сети магазинов) в единый управляющий центр. На данный момент уже существуют технологии, направленные на организацию работы удалённых сотрудников во многих областях деятельности, однако, поскольку, работа с удалёнными сотрудниками рядом специфических особенностей по сравнению с рабочим процессом традиционного офиса, решение задач разработки и исследования новых технологий в данном направлении является актуальным.

В данной статье предложена базовая концептуальная модель рабочего поведения удалённых сотрудников с использованием программных интеллектуальных агентов.

1. Программные интеллектуальные агенты

Интеллектуальные программные агенты – программные компоненты, которые могут выполнять специфические задачи, поставленные пользователем, и обеспечивать такую степень интеллекта, которая позволяет выполнять эти задачи автономно, адаптируясь к среде и взаимодействуя со средой и другими агентами рациональным способом (Рис.1).

Рис. 1. Взаимодействие агента с внутренней и внешней средой

В искусственном интеллекте, под термином интеллектуальный агент понимаются разумные сущности, наблюдающие за окружающей средой и действующие в ней, при этом их поведение рационально в том смысле, что они способны к пониманию и их действия всегда направлены на достижение какой-либо цели. Такой агент может быть как роботом, так и встроенной программной системой. Об интеллектуальности агента можно говорить, если он взаимодействует с окружающей средой примерно так же, как действовал бы человек [2,3].

Агенты могут применяться при решении следующих задач [3]:

1. Мобильные вычисления (миграция агентов может поддерживаться не только между постоянно подсоединенными к сети узлами, но и между мобильными платформами, подключаемыми к постоянной сети на дискретные промежутки времени и возможно по низкоскоростным каналам). Клиент подсоединяется к постоянной сети на короткий промежуток времени с мобильной платформы, отправляет агента для выполнения задачи и отсоединяется; затем клиент подсоединяется к другой точке сети и забирает результаты работы агента. Второй вариант – сервер, на который должен переместиться агент, подсоединяется к сети, а затем отсоединяется. В этом случае агент должен уметь переместиться на такой временно подсоединяемый сервер и вернуться в постоянную сеть.

2. Задачи управления информацией:

– поиск информации (большой объем информации – один человек не в состоянии найти необходимую информацию и проанализировать ее за ограниченное время. Использование интеллектуальных программных агентов позволяет решить эту задачу); поисковые агенты содержат сведения о различных информационных источниках (включая тип информации, способ доступа к ней, а также такие характеристики информационного источника, как надежность и точность данных);

– отбор (обработка) информации. Из всех данных, приходящих к клиенту, выбирают только те данные, которые могут быть интересны клиенту. Используются в комбинации с поисковыми агентами (сначала поиск, затем – отбор);

– мониторинг данных. Извещение пользователя об изменениях в различных источниках данных в режиме реального времени (например, мобильный агент перемещается на вычислительный узел, на котором расположен источник данных; это эффективнее, чем использовать статического агента, посылающего запросы источнику данных);

– универсальный доступ к данным. Агенты – посредники для работы с различными источниками данных, имеющие механизмы для взаимодействия друг с другом (например, агент создает несколько агентов, каждый из которых работает со своим источником данных).

Интеллектуальный агент владеет определенными знаниями о себе и об окружающей среде, и на основе этих знаний он способен определять свое поведение. Способность планировать свои действия делит агентов на *регулирующие* и *планирующие*. Если умение планировать не предусмотрено (регулирующий тип), то агент будет постоянно переоценивать ситуацию и возобновлять свое воздействие на окружающую среду. Планирующий агент может запланировать несколько действий на разные промежутки времени. При этом агент может моделировать развитие ситуации, что дает возможность более адекватно реагировать на текущие ситуации. При этом агент должен принимать во внимание не только свои действия и реакцию на них, но и сохранять модели объектов и агентов окружающей среды для прогнозирования их возможных действий и реакций.

В рамках предлагаемой концептуальной модели предполагается создание системы прогнозирования рабочего поведения удалённых сотрудников за счёт использования интеллектуальных программных агентов. Главная задача агента заключается в том, чтобы накапливать и структурировать данные о процессе рабочего взаимодействия между удалёнными сотрудниками и их менеджером, а также строить на основе получаемых данных некоторую структуру данных и знаний. Эта структура, в совокупности с инструментами её обработки, и будет являться основой модели рабочего поведения сотрудника (МРПС). Обобщенная структура интеллектуального агента приведена на рис.2. Система предназначена

для сопровождения процессов рабочего взаимодействия в условиях удалённого сотрудничества между исполнителями (собственно удалёнными сотрудниками) и их менеджером (в ситуации, когда один менеджер работает с несколькими исполнителями). Такое сопровождение актуально на двух ключевых этапах взаимодействия: этап согласования работы (поиск и выбор исполнителя, а также передача ему некоторой работы) и этап контроля результата (пере дача выполненной работы менеджеру со всеми сопутствующими процессами) [4].

Предполагается, что все индивидуальные модели процессов агентного взаимодействия будут объединены в некоторую общую интерфейсную оболочку, снабжённую модулем маршрутизации запросов и ответов на них (задача реализации такой оболочки в данной работе не рассматривается).

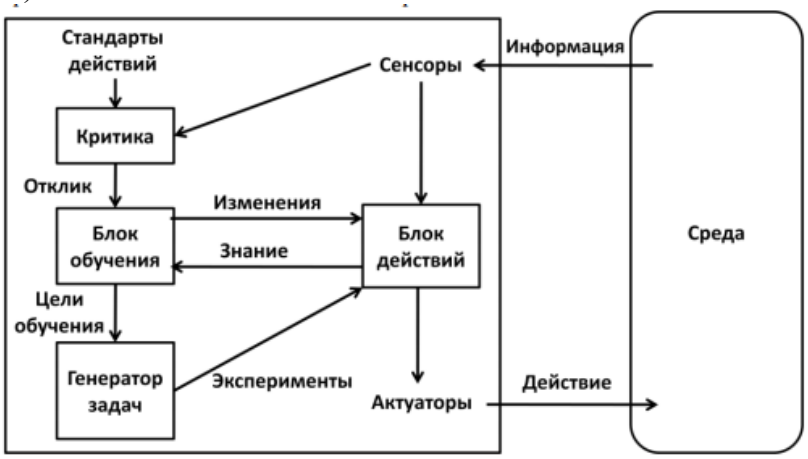


Рис. 2. Структура интеллектуального агента

2. Базовая модель рабочего поведения сотрудников в среде AnyLogic

В ходе исследований в среде AnyLogic 7.2 была построена базовая модель удаленного офиса. Модель эмулирует взаимодействие интеллектуальных сущностей: менеджер, служащий, база знаний и распределенная программная система (РПС). Предполагается, что система обладает ограниченной емкостью и абсолютной надежностью.

Менеджер посылает запросы в базу знаний с некоторым интервалом времени, распределенном по показательному закону. Время обработки удаленным работником запроса менеджера также распределено по экспоненциальному закону. Удаленный офис имеет входной буфер, так называемую базу знаний, которая подбирает работника для выполнения задания по требуемым параметрам.

В качестве основных агентов модели исследуются агенты типа *Служащие* и *РПС*. Тип *служащие*, отвечает за доступность (свободен/занят) удаленного сотрудника к выполнению задания. Тип *РПС* определяет сценарий работы агентов запросов. Агентов типа *РПС* в системе может быть много (различные запросы от менеджера), агент типа *Служащие* представлен в виде коллекции из трех работников (предполагается что в системе есть три удаленных сотрудника, обрабатывающих запросы). Сценарий поведения агентов может быть описан логическими схемами (*Statechart*), включающими функциональные блоки и связи между ними. Элементы разработанной модели представлены на рис. 3 и рис. 4. На рис. 3 видно, что агент типа *Служащие* имеет 2 состояния, т.е. работник может быть или свободен или быть занят обработкой запроса. Также предусмотрен параметр «*svoboden*», для передачи информации о состоянии сотрудника другим агентам.

На рис. 4 представлена диаграмма действий для агентов типа *Запросы*. Сценарий действий заключается в следующем: при создании агента-запроса он сначала ожидает ответа от агента-сотрудника (занят/свободен). После подбора соответствующего работника происходит непосредственно обработка запроса. После того как задание выполнено происходит освобождение агента-сотрудника.

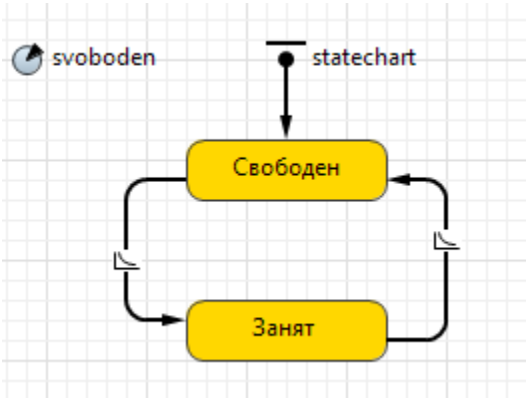


Рис. 3. Агент «Служащие»

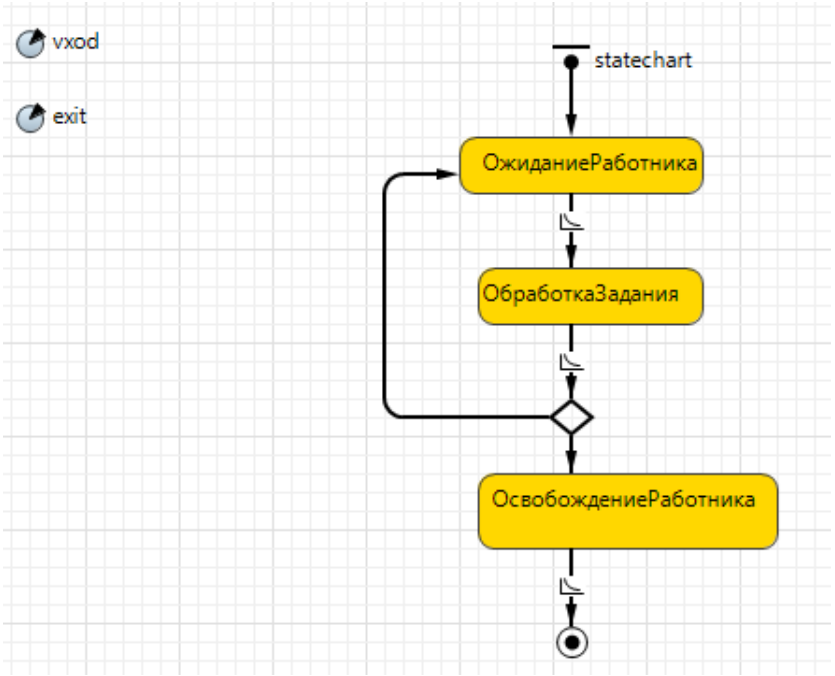


Рис. 4. Агент «Запросы»

Рассматриваемая модель использует концепцию многоподходового моделирования, агентный подход используется для моделирования логики обработки запросов, а дискретно-событийный подход для моделирования самого процесса обработки. Схема комбинированной модели представлена на рис. 5. В схеме модели присутствуют следующие обозначения:

- *запросы[..]* – популяция агентов типа «Запросы»;
- *служащие* – агент типа «Служащие»;
- *time* – суммарное время, затрачиваемое на обработку запросов в системе удаленного офиса;
- *kol_vo* – параметр, накапливающий количество запросов к отдельному агенту-сотруднику;
- *kolzap* – параметр, накапливающий количество успешно отработанных работником запросов;
- *time_obrabotki* – суммарное время нахождения запроса в системе удаленного офиса;
- *ver_obrabotki* – процент обработанных запросов в системе удаленного офиса.

В перспективе дальнейших исследований предполагается дальнейшая модификация модели, а именно введение обучаемости агентов основе принятых ранее решений [5]. Также предполагается введение рейтинговой системы среди агентов-сотрудников, исследование процессов взаимодействия агентов «база знаний» и «менеджер».

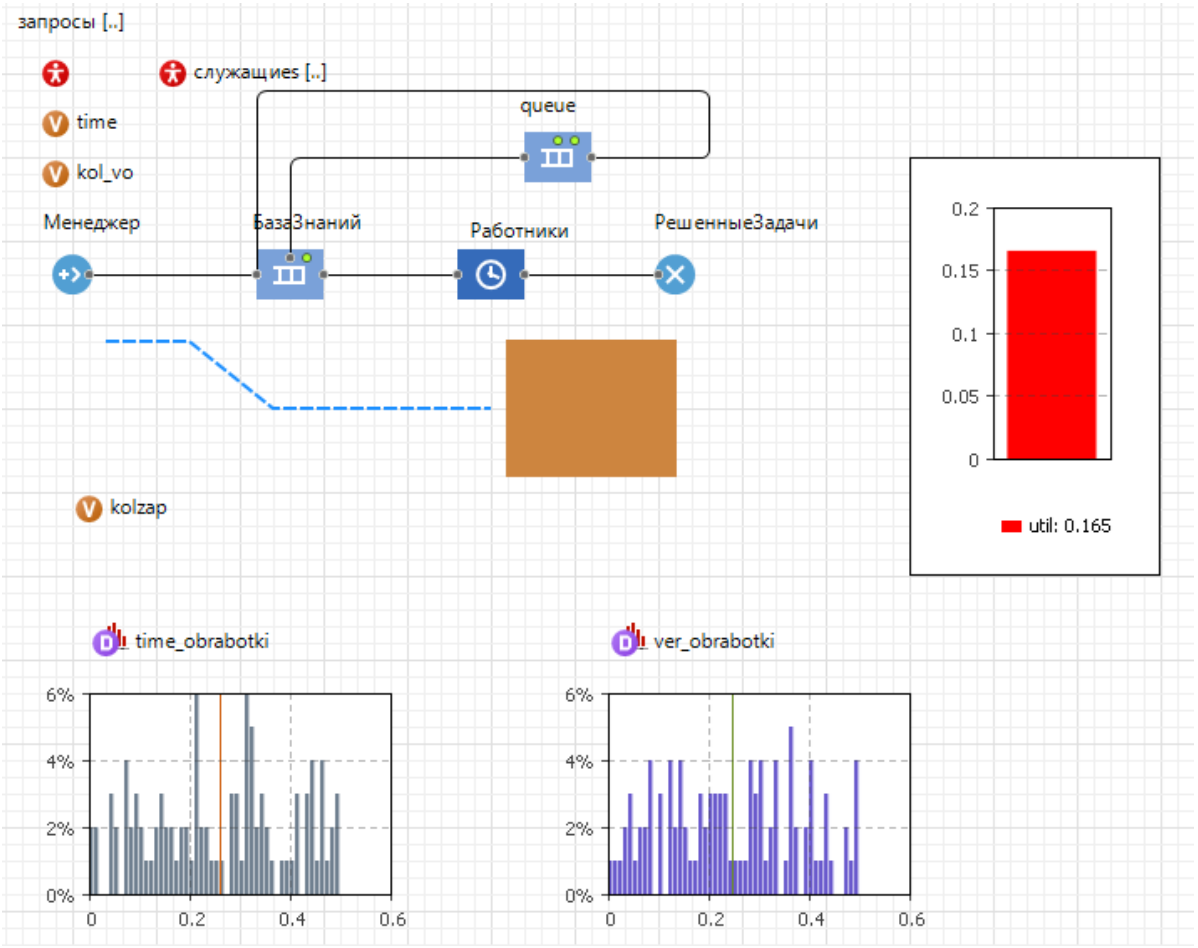


Рис. 5. Схема модели при использовании много подходной концепции

Выводы

Для решения задачи адекватной оценки и прогнозирования рабочего поведения удалённых сотрудников, предложена базовая имитационная модель, основанная на применении программных интеллектуальных агентов. В рамках модели разработана стратегия функционирования системы вцелом и индивидуальные стратегии поведения сотрудников, сформулированы ограничения и допущения моделирования. Применение результатов работы при разработке систем управления распределенным офисом позволяет повысить эффективность принимаемых менеджерами организаций решений в области управления персоналом.

Библиографический список

1. Чеканов А. Распределенный офис: как сократить издержки и не потерять людей [Текст] / А. Чеканов // IT-MANAGER.– 2009.– №1.– С. 6–8.

2. Wooldridge M. An Introduction To Multiagent Systems [Текст] / M. Wooldridge.– Liverpool. : Wiley & Sons, 2002.– 342 p.

3. Рутковский Л. Методы и технологии искусственного интеллекта [Текст] / Л. Рутковсий.– М.: Горячая линия-Телеком, 2010.– 520 с.– ISBN 5-912-0105-6.

4. Шушура А.Н. Моделирование рабочего поведения удалённых сотрудников с использованием интеллектуальных агентов [Текст] / А.Н. Шушура, К.В. Темник // Восточно-Европейский журнал передовых технологий. – 2011. – №6/2(54). – С. 19 – 21.

5. Рассел С. Искусственный интеллект: современный подход [Текст] / С. Рассел, П. Норвиг. – 2-е изд. – М.: Издательский дом «Вильямс», 2006. –1408 с.

УДК 681.3
А.В. Будько, магистрант
Научный руководитель: Н.Е. Сапожников, д-р. техн. наук, профессор
Севастопольский государственный университет

К ВОПРОСУ О ПАРАЛЛЕЛЬНОМ ВЫПОЛНЕНИИ ДИСКРЕТНОГО ПРЕОБРАЗОВАНИЯ ФУРЬЕ

Аннотация: В работе излагается схема параллельного выполнения дискретного преобразования Фурье (ДПФ). Схема включает вычисление базиса ДПФ на основе кусочно-полиномиальной аппроксимации функций с помощью интерполяционного полинома Ньютона.
Ключевые слова: Дискретное преобразование Фурье, Цифровая обработка сигнала, Кусочно-полиномиальная аппроксимация функций, Полином Ньютона.
Annotation: The methods of the parallel execution of the discrete Fourier transform (DFT). The scheme involves the calculation of DFT basis based on piecewise polynomial approximation of functions with the help of Newton's interpolation polynomial.
Key words: The discrete Fourier transform, digital signal processing, piecewise polynomial approximation of functions, polynomial Newton.

Постановка вопроса

Элементарные функции – составляют базис ортогонального разложения. Они вычисляются для реализации алгоритмов ЦОС. Тригонометрические функции, являются базисом дискретного преобразования Фурье.

К их вычислению предъявляются требования:

1. Высокого быстродействия
2. Высокой точности
3. Вычислительной устойчивости

Сложность данных вычислений, а также точность, влияют на быстродействие и точность ЦОС. В дальнейшем алгоритм вычисления базиса ДПФ необходимо распараллелить с помощью кусочно-полиномиальной схемы аппроксимации функций. За основу берутся интерполяционный полином Ньютона, формулы редукции, распространяемые на вычисление ДПФ в целом.

Решение задачи

Фундаментальное уравнение для получения – N-точечного ДПФ записывается в виде

$$X(k) = \frac{1}{N}\sum_{n=0}^{N-1} x(n)[\cos\left(\frac{2\pi nk}{N}\right) - j\sin(\frac{2\pi nk}{N})\,] \qquad (1)$$

Здесь X(k) – частотный выход ДПФ в k-ой точке спектра, где k = 0, 1, …, N-1; N – число отсчетов при вычислении ДПФ; значение x(n) представляет собой n-й отсчет во временной области, n = 0, 1, …, N-1. Выходной спектр ДПФ X(k) - результат свертки между выборкой, состоящей из входных отсчётов во временной области, и набором из N пар гармонических базисных функций (косинус и синус).

Схема [1] вычисления функции основана на интерполяционном полиноме Ньютона. Пусть рассматривается функция одной действительной переменной, для определенности вида

$$f(x) = \sin(\pi x)\,, x \in [0,1] \qquad (2)$$

Вычисление функции требуется для построения базиса преобразования Фурье. Берется система непересекающихся равных по длине подынтервалов, объединение которых совпадает с [0,1]:

$$[0,1] = \bigcup_{i=0}^{P=2}[x_i, x_{i+1}]\bigcup[x_{P-1}, x_P], x_{i+1} - 1 = \frac{1}{P}, i = 0,1,\dots,P-1 \qquad (3)$$

При предварительно данной границе ε полнейшей погрешности аппроксимации функции (2) и для любого раздельно взятого подынтервала из (3) строится интерполяционный полином Ньютона ступени n , где n выбирается наименьшим для достижения данной точности

приближения сразу на всех подынтервалах. При всем этом полином Ньютона преобразуется по дистрибутивности с приведением схожих так, что в следствии преобразования на i-м подынтервале получается канонический вид полинома от одной переменной:

$$P_n(x) = a_{0i} + a_{1i}x^1 + \cdots + a_{ni}x^n, x \in [x_i, x_{i+1}], i = 0,1, \dots, P-1, \quad (4),$$

где согласно (3) для i=P-1 равенство сохраняется при x=x_p.[2,3]

В (4) индекс коэффициента i равен номеру подынтервала. Построение (4) проводится для всех P подынтервалов, что бы заданная граница погрешности не превышалась ни на одном из них:

$$|\sin(\pi x) - P_n(x)| \le \varepsilon, x \in [x_i, x_{i+1}], i = 0,1, \dots, P-1 \quad (5)$$

В случае нахождения минимального значения n для всех подынтервалов, то для функции (2) и для каждого подынтервала из (3) набор коэффициентов в (4) запишем в память машины и сохраним. В дальнейшем, если потребуется вычислить похожую функцию, сначала расшифруется значение индекса i подынтервала, являющееся математическим адресом выборки коэффициентов (4). Если $x \in [x_i, x_{i+1}]$, то $i = int\left(\frac{x}{H}\right)$, где int – целая часть числа, $H = x_{i+1} - x_i$. Очередь времени расшифровки считается как единичный $t = O(1)$. Далее время вычисления функции будет определяться как степень полинома (4). По схеме Горнера это значение вычисляется с временной погрешностью $t(1) = n(t_y + x_c)$, где t_c, t_y – время бинарного сложения и произведения. Если же уменьшить длину подынтервала степень n в (4) можно сделать необходимо низкой(но целой) при пропорциональном возрастании P в (3). Цель осуществления описываемого построения вычисление коэффициентов начинается для значения n=1 при самом малом значении P. Если достижение заданной точности (4) невозможно, то значение P необходимо удвоить; и так до нарушения допустимых границ значений P; при их нарушении возвращаемся к минимальному значению P, но уже при n=2 , и т.д.

В границах интерполяции по Ньютону детализация схемы минимизации степени полинома происходит следующим образом. Если границы i -го подынтервала из (3) обозначить как a_{i0}, b_{i0}, и интервал интерполяции $w_i = \frac{b_{i0} - a_{i0}}{n}$, то параллельные узлы интерполяции на текущем шаге будут представлены как $x_{ij} = a_{i0} + jw_i,\ j = 0,1, \dots, n-1$.

Полином Ньютона степени n на i-м подынтервале для функции (3) и данных узлов интерполяции записывается как $\Psi_{ni}(x) = f(x_{i0}) + \sum_{j=1}^{n} \frac{\Delta^j y_{i0}}{j! w_i^j} \prod_{k=0}^{j=1}(x - x_{ik})$, где $\Delta^j y_{i0}$- конечная разность j-ого порядка в точке x_{i0}, или в обозначении $t = \frac{x - x_{i0}}{w_i}$,

$$\Psi_{ni}(t) = f(x_{i0}) + \sum_{j=1}^{n} \frac{\Delta^j y_{i0}}{j!} \prod_{k=0}^{j=1}(t-k) \quad (6)$$

Процесс приведения (6) к виду (4), влечет значения коэффициентов $a_{i0} = f(x_{i0}), a_{li} = \sum_{j=1}^{n} b_{ij} d_{lj}$, где $b_{ij} = \frac{\Delta^j y_{i0}}{j!}$, d_{lj} – коэффициенты полинома вида $P_{nj}(t) = d_{0j} + d_{1j}t^1 + \cdots + d_{nj}t^n$ с натуральными корнями,

входящими в состав полинома Ньютона. Они не зависят от вида аппроксимируемой функции, по окончанию вычисления их можно сохранить для дальнейших использований интерполяции. Для получения коэффициентов используется аналог формул Виета из [2, 3].

Само построение полинома на каждом подынтервале из (3) начинается с n=1 и P=1. Проверка на точность аппроксимации, заданную соотношением (5), (где $P_{ni}(x) = \Psi_{ni}(t)$) происходит на каждом подынтервале. Если на всех проверочных точках(t), каждого из подынтервалов соотношение (5) не изменилось, то соответственно и сама аппроксимация на всех частях осуществима полиномом $P_{ni}(x) = \Psi_{ni}(t)$ при заданном n. При нарушении (5) в любой из точек, при сохранении того же значения P, необходимо увеличить степень полинома на 1, и весь процесс построения Ψ_n и проверки его на точность аппроксимации повторить. При

достижения нужной погрешности ε в (5) коэффициенты (4) сохраняются в компьютере в адресах (i, l), где i – номер подынтервала, i = 0 ,1, … , P − 1 , l – номер коэффициента полинома в (4), l = 0 ,1,…, n , где n – степень полинома.

Заключение

Подтверждена схема параллельного вычисления базиса ДПФ и вычисления ДПФ. Таблично-алгоритмическая схема построения базиса ДПФ имеет временную сложность O(1) для произвольно заданной заранее границы погрешности ε, время вычисления базиса не зависит от числа отсчётов ДПФ.

Библиографический список

1. Аксайская Л. Н. Разработка и исследование параллельных схем цифровой обработки сигналов на основе минимизации временной сложности вычисления функций / Автореферат диссертации на соискание ученой степени кандидата технических наук. – Таганрог: ТТИ ЮФУ. – 2008. – 18 с.
2. Ромм Я.Е. Бесконфликтные и устойчивые методы детерминированной параллельной обработки / Дис д-ра техн. наук. – Таганрог: ТРТУ, 1998. – 546 с.; ВНТИ Центр.– № 05.990.001006.
3. Ромм Я.Е. Локализация и устойчивое вычисление нулей многочлена на основе сортировки. II // Кибернетика и системный анализ. – 2007. – № 2. – С.161-174.

УДК 681.3
М.С. Гак, магистрант 2-го курса
Научный руководитель: Е.А.Кожаев, канд. техн. наук, доцент
Севастопольский государственный университет

ОПТИМИЗАЦИЯ КОНСТРУКТОРСКО-ТЕХНОЛОГИЧЕСКИХ РЕШЕНИЙ ПРИ ПРОЕКТИРОВАНИИ И ИЗГОТОВЛЕНИИ ИЗДЕЛИЙ МЭА

Аннотация: рассматриваются принципы организации, основные функции и структурные компоненты программного комплекса при оптимизации конструкторско-технологических решений.
Ключевые слова: имитационное моделирование, процедура принятия решения, процедура оптимизации.
Annotation: principles of the organization, main functions and structural components of the software system while optimizing the design and technological solutions was considered.
Keywords: simulation, decision-making procedure, the optimization procedure.

Задача повышения качества микроэлектронной аппаратуры (МЭА) решают на стадии проектирования, когда можно всесторонне проанализировать конструктивные варианты с учетом большого числа требований[1]. Как правило, не существует машины или конструкции, оптимальной по всем критериям одновременно. Поэтому расчеты производят для каждого критерия, строят таблицы результатов расчетов и используют их для обоснования выбора оптимального решения.

Развитие техники сопровождается усложнением всех систем и технологического оборудования. Возрастает трудоемкость их создания при одновременном повышении требований к качеству и эффективности конструкции, что находится в противоречии с необходимостью сокращения сроков ее разработки и промышленного освоения. Ликвидация указанного противоречия наиболее полно реализуется при широком внедрении в проектирование вычислительной техники.

Задачи математического моделирования, с целью принятия решения по выбору оптимального режима работы технологических объектов современного производства МЭА, обычно являются многокритериальными.

Типовой процесс принятия решений на производстве представлен на рисунке 1.

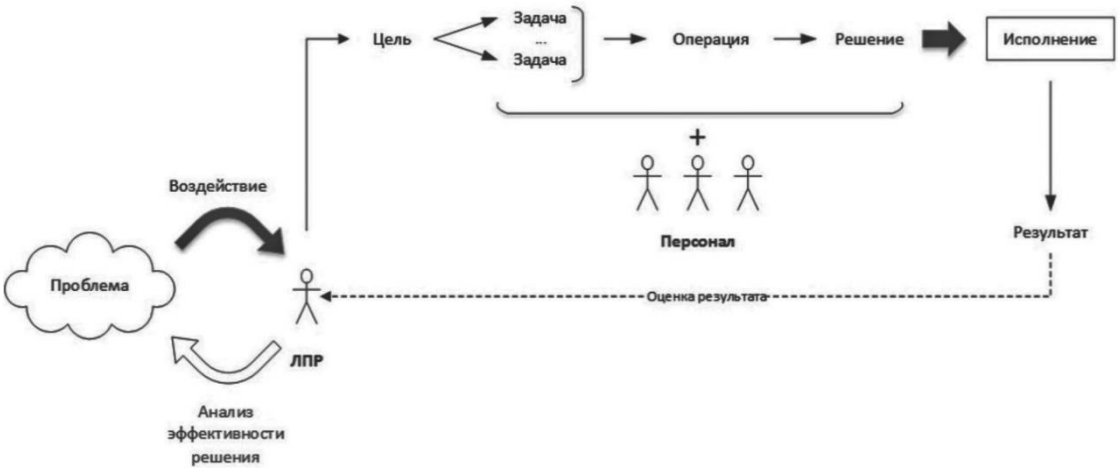


Рис. 1. Процесс управления предприятием

К основным критериям при принятии решения в управлении можно отнести: повышение производительности, обеспечение желаемых качеств вырабатываемых продуктов, снижение их себестоимости, экономия материалов и ресурсов, обеспечение стабильности и улучшения экологического состояния производства, охрана окружающей средыи здоровья персонала, причем часто они бывают противоречивыми.

Таким образом, задачи принятия оптимального решения при управлении технологическими объектами производства МЭА, сводятся к решению задач векторной оптимизации, которые позволяют найти область эффективных решений. Окончательный выбор и принятие решений осуществляет ЛПР (лицо, принимающее решение) на основе своего предпочтения, ситуаций на производстве, рынке и т.д.

Современное предприятие рассматривается как сложная система, в которой в единую логистическую цепь объединяются процессы закупки сырья и материалов, производства продукции, ее хранения и распределения. Особенностью таких систем является невозможность адекватного описания процессов их функционирования с помощью аналитических моделей. Это связано с многообразием связей между структурными элементами систем, неоднозначностью алгоритмов поведения при различных условиях, с корреляционным взаимодействием между большим числом параметров. Необходимость комплексного учета данных факторов затрудняет аналитическую формулировку критериев оптимальности и ограничений в оптимизационных задачах и приводит к использованию аппарата имитационного моделирования [2].

Оптимизационно-имитационный подход основан на совместном применении имитационных и оптимизационных моделей в процессе управления [3]. При этом имитационные модели используются для оценки основных характеристик производственной системы – коэффициентов загрузки оборудования, средней длины очереди на производственных участках, объема выпускаемой продукции, временных параметров и т.д. Выходные характеристики производственной системы являются исходной информацией для формирования критериев оптимальности и ограничений в оптимизационных моделях. Для рационального выбора текущих значений входных управляемых параметров имитационных моделей используются алгоритмы оптимизации поискового типа. Целенаправленным варьированием параметров имитационной модели под управлением процедур оптимизации можно получить оптимальные и близкие к оптимальным решения.

Рассматриваемые оптимизационные задачи обобщенно могут быть сформулированы в виде:

$$f_i(X) \to \min_{X \in D}, i = \overline{1,m};$$

$$D = \{ X \mid x_j^{min} \le x_j \le x_j^{max}, j = \overline{1,n};$$

$$g_s(X) \ge 0, h_k(X) = 0, s = \overline{1,r}, k = \overline{1,p} \}.$$

Здесь $X = (x_1, \dots, x_n)$– вектор варьируемых параметров модели; $f_i(X)$– частные критерии оптимальности; D – допустимая область, представленная ограничениями двух видов: прямыми ($x_j^{min} \le x_j \le x_j^{max}$) и функциональными ($g_s(X) \ge 0, h_k(X) = 0$). При этом предполагается, что критерии и ограничения заданы алгоритмически в виде моделирующих процедур, позволяющих по заданным значениям $X = (x_1, \dots, x_n)$получать значения $f_i(X), g_s(X), h_k(X)$.

На рисунке 2 показана структура программного комплекса поддержки принятия решений при оптимизации производственного процесса на предприятии. Программный комплекс имеет модульную структуру и состоит из набора взаимодействующих блоков, наиболее значимыми из которых являются блоки функционального моделирования, имитационного моделирования, оптимального поиска и управления. Блок функционального моделирования предназначен для структурного и функционального анализа производственных процессов предприятия.

На базе разработанной функциональной модели в дальнейшем строится имитационная модель с целью проведения динамического многовариантного анализа и оптимизации бизнес-процессов производства. В качестве среды моделирования выбрана система AnyLogic, в настоящее время являющаяся наиболее эффективным инструментальным средством имитационного моделирования производственных систем [4]. Достоинством системы является наличие развитого графического интерфейса и средств анимации моделей, способствующих повышению эффективности принимаемых управленческих решений. Выходная информация представляется в виде динамических графиков, диаграмм, структурированных таблиц, обеспечивающих формирование целостной картины производственно-сбытового процесса на предприятии.

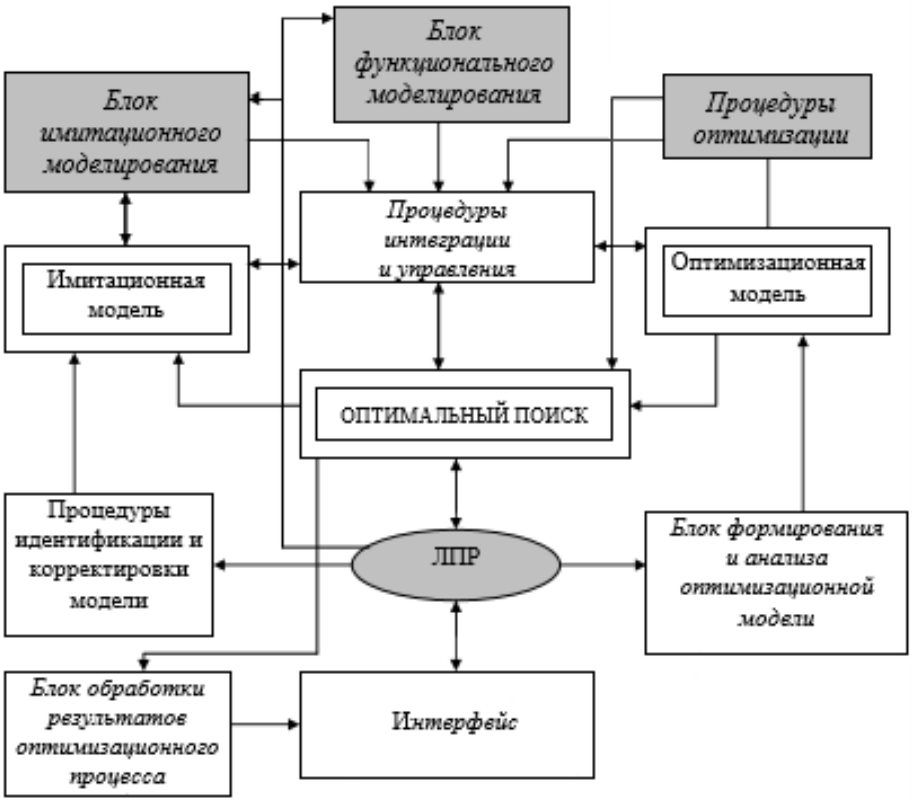


Рис.2. Структура программного комплекса поддержки принятия решений

Решение оптимизационных задач обеспечивается блоком оптимального поиска.Перед началом оптимального поиска осуществляются построение оптимизационной модели (определение размерности задачи, формирование критериев и ограничений), после чего проводится полнофакторный эксперимент данной задачи [5].

Связующим звеном между основными блоками программного комплекса, обеспечивающим передачу информации и формирование управляющих воздействий, является блок управления. Основные его функции – взаимодействие с ЛПР для координации его действий, связь по данным и управлению между основными программными компонентами, передача результатов оптимизации для обработки и дальнейшего анализа.

В заключение следует отметить, что применение программного комплекса поддержки принятия решений при оптимизации производственных систем позволит учитывать различные аспекты, возникающие при выборе оптимальных управляющих воздействий на систему.

Библиографический список

1. Преснухин Л.Н. Конструирование ЭВМ и систем: учебник для вузов / Л. Н. Преснухин, В. А. Шахнов – М. Высшая школа, 1986. – 508 с.
2. Боев В.Д. Компьютерное моделирование: Пособие для практических занятий, курсового и дипломного проектирования в AnyLogic7:. — СПб.:2014. — 432 с.
3. Хоботов Е.Н. Оптимизационно-имитационный подход к моделированию сложных систем /Е.Н.Хоботов – Изв. РАН, 1996. – 224 с.
4. Мезенцев. Н.К. Учебное пособие «Моделирование систем в среде AnyLogic 6.4.1» / Н.К. Мезенцев — М.: 2011. – 103 с.
5. Гайдадин Н.А. Применение полного факторного эксперимента при проведении исследований: метод.указания / сост. А.Н.Гайдадин, С.А.Ефремова; ВолгГТУ. – Волгоград, 2008. – 16 с.

УДК 681.3
М.В. Гришин, магистрант
Научный руководитель: Е.Н. Мащенко, к.т.н., доцент
Севастопольский государственный университет

Исследование алгоритма балансировки нагрузки серверов на основе имитационной модели

Аннотация: В статье рассматриваются особенности процесса балансировки нагрузки серверов, предлагается имитационная модель процесса балансировки по алгоритму RoundRobin .
Ключевые слова: Балансировка нагрузки, распределенные системы, MySql, кэш данных.
Annotation: Article deals with the features of the server load balancing process, it is proposed simulation model of the balancing process using RoundRobin algorithm.
Key words: Balancing of loading, the distributed systems, MySql, cache of data.

Введение

Вопрос о планировании нагрузки на серверное оборудование следует решать на ранней стадии развития любого ИТ-проекта. Первоначально проблемы недостаточной производительности сервера в связи ростом нагрузок можно решать путем наращивания мощности сервера, или же оптимизацией используемых алгоритмов, программных кодов и так далее. Но рано или поздно наступает момент, когда и эти меры оказываются недостаточными.

В таком случае приходится прибегать к кластеризации: несколько серверов объединяются в кластер; нагрузка между ними распределяется при помощи комплекса специальных методов, называемых балансировкой. Помимо решения проблемы высоких нагрузок кластеризация помогает также обеспечить резервирование серверов.

Эффективность кластеризации напрямую зависит от того, как распределяется нагрузка между элементами кластера. Балансировка нагрузки может осуществляться при помощи как аппаратных, так и программных инструментов [1].

В числе целей, для достижения которых используется балансировка, нужно выделить следующие [1]:

- справедливость: нужно гарантировать, чтобы на обработку каждого запроса выделялись системные ресурсы и не допустить возникновения ситуаций, когда один запрос обрабатывается, а все остальные ждут своей очереди;
- эффективность: все серверы, которые обрабатывают запросы, должны быть равномерно загружены; желательно не допускать ситуации, когда один из серверов простаивает в ожидании запросов на обработку (в реальной практике эта цель достигается далеко не всегда);
- сокращение времени выполнения запроса: нужно обеспечить минимальное время между началом обработки запроса (или его постановкой в очередь на обработку) и завершением обработки;
- сокращение времени отклика: нужно минимизировать время ответа на запрос пользователя.

Необходимо также, чтобы алгоритм балансировки обладал следующими свойствами [1]:

- предсказуемость: нужно чётко понимать, в каких ситуациях и при каких нагрузках алгоритм будет эффективным для решения поставленных задач;
- равномерная загрузка ресурсов системы;
- масштабирумость: алгоритм должен сохранять работоспособность при увеличении нагрузки.

Алгоритм RoundRobin

RoundRobin, или алгоритм кругового обслуживания, представляет собой перебор по круговому циклу: первый запрос передаётся одному серверу, затем следующий запрос передаётся другому и так до достижения последнего сервера, а затем всё начинается сначала. Самой распространённой имплементацией этого алгоритма является, конечно же, метод балансировки RoundRobin DNS. Как известно, любой DNS-сервер хранит пару «имя хоста —

IP-адрес» для каждой машины в определённом домене: DNS-сервер проходит по всем записям таблицы и отдаёт на каждый новый запрос следующий IP-адрес. В результате все серверы в кластере получают одинаковое количество запросов.

В числе достоинств этого алгоритма следует назвать, во-первых, независимость от протокола высокого уровня. Для работы по алгоритму RoundRobin используется любой протокол, в котором обращение к серверу идёт по имени.Балансировка на основе алгоритма RoundRobin никак не зависит от нагрузки на сервер: кэширующие DNS-серверы помогут справиться с любым наплывом клиентов. Использование алгоритма RoundRobin не требует связи между серверами, поэтому он может использоваться как для локальной, так и для глобальной балансировки. Наконец, решения на базе алгоритма RoundRobin отличаются низкой стоимостью: чтобы они начали работать, достаточно просто добавить несколько записей в DNS.

Алгоритм RoundRobin имеет и ряд существенных недостатков. Чтобы распределение нагрузки по этому алгоритму отвечало упомянутым выше критериями справедливости и эффективности, нужно, чтобы у каждого сервера был в наличии одинаковый набор ресурсов. При выполнении всех операций также должно быть задействовано одинаковое количество ресурсов. В реальной практике эти условия в большинстве случаев оказываются невыполнимыми. Также при балансировке по алгоритму RoundRobin совершенно не учитывается загруженность того или иного сервера в составе кластера. В силу описанных выше обстоятельств сфера применения алгоритма RoundRobin весьма ограничена.

На рисунке 1 изображена схема распределения нагрузки на сервера базы данных. На рисунке 2 изображена схема работы кэша, служащего для снижения нагрузки на веб-сервера.

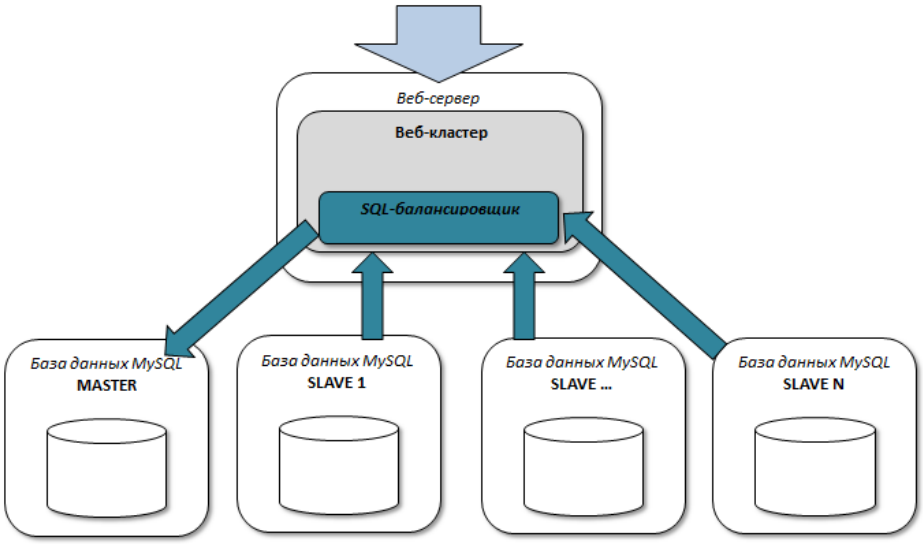


Рис. 1 – Схема распределения нагрузки на сервера базы данных.

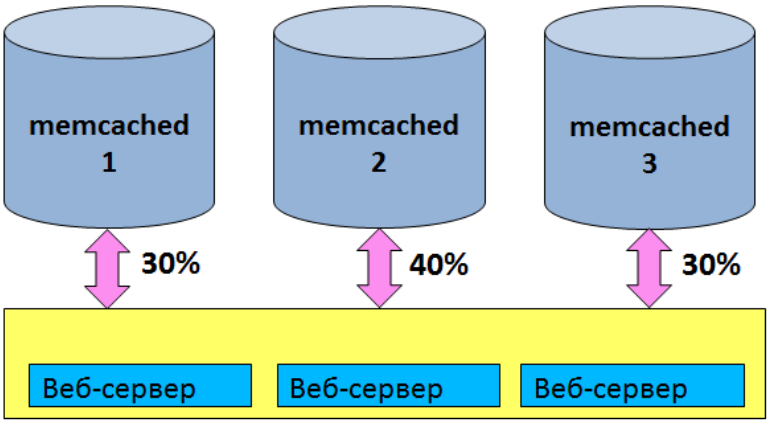


Рис. 2. Схема работы кэша в системе балансировки

За основу имитационной модели взят алгоритм балансировки RoundRobin, так как он прост в использовании и легко расширяем. Схема имитационной модели представлена на рисунке 3.

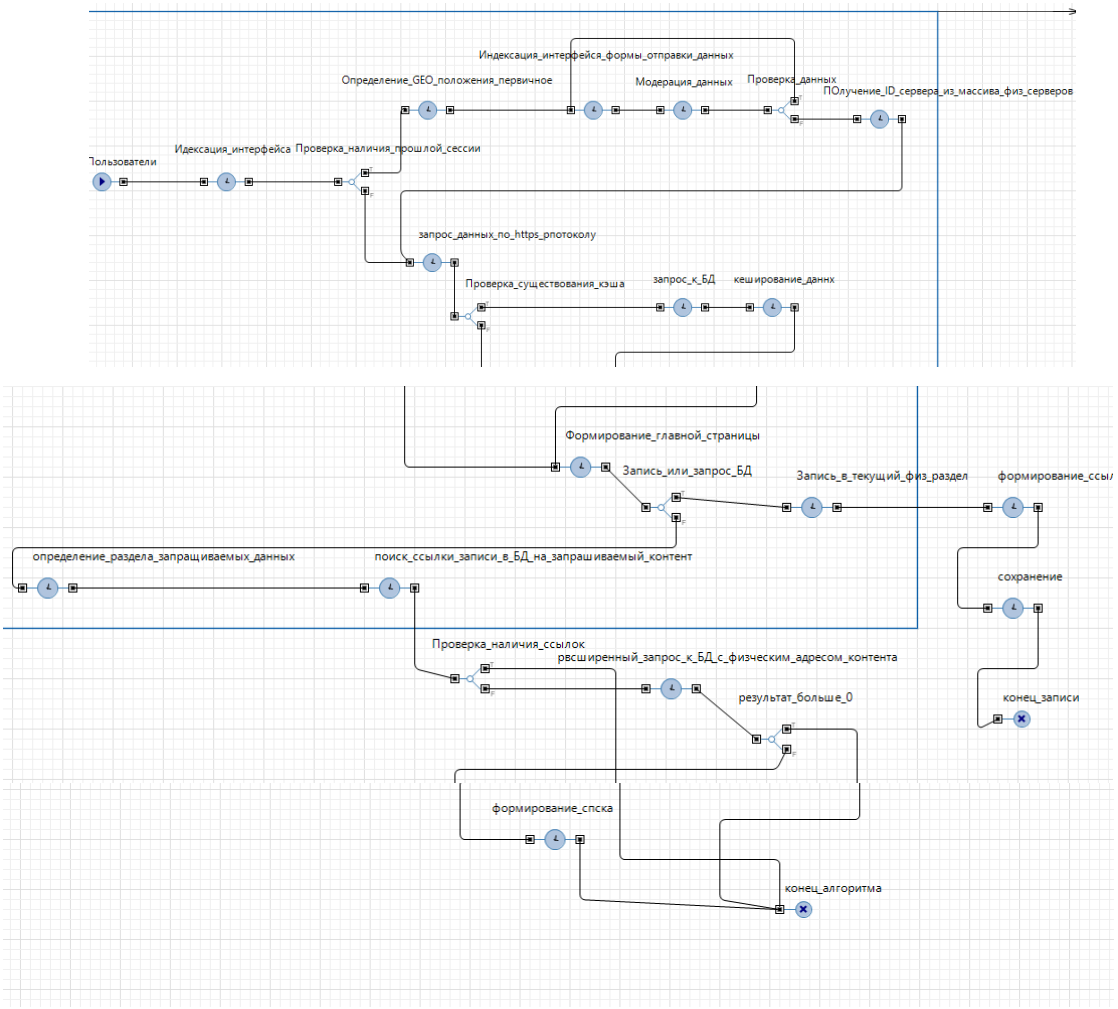


Рис. 3. Имитационная модель процесса балансировки и распределения нагрузки на веб-сервера.

Заключение

В статье рассмотрен расширенный алгоритм балансировки нагрузки на сервер RoundRobin, который работает во взаимодействии с формируемым кэшем, что позволяет эффективно балансировать нагрузку на сервера и базы данных. На основе данного алгоритма построена имитационная модель в среде AnyLogic 7. В модели подробно представлены основные этапы балансировки нагрузки. В качестве потока входящих заявок модели взяты запросы от пользователей системы к серверу в определенные моменты времени. Параметры модели будут определены на основе статистических исследований процесса балансировки серверов в кластере.

Библиографический список

1. А. Емельянов. Балансировка нагрузки: основные алгоритмы и методы [Электронный ресурс] – Режим доступа: https://blog.selectel.ru/balansirovka-nagruzki-osnovnye-algoritmy-i-metody/ (дата обращения: 04.02.2016).
2. Anylogic.ru [Электронный ресурс]: Инструмент многоподходного имитационного моделирования / AnyLogic - единственный многоподходный инструмент имитационного моделирования: агентное, дискретно-событийное моделирование и системная динамика. – 2012. – 21 марта. – Режим доступа: http://www.anylogic.ru/ (дата обращения: 14.02.2016).

УДК 681.3
Е.Б. Доронина, студентка 4-го курса
Научный руководитель: И.В.Дымченко, старший преподаватель
Севастопольский государственный университет

ПРИМЕНЕНИЕ МЕТОДОЛОГИИ ФУНКЦИОНАЛЬНОГО МОДЕЛИРОВАНИЯ ДЛЯ ОПИСАНИЯ ПРОЦЕССОВ ДЕЯТЕЛЬНОСТИ КАФЕДРЫ

Аннотация: предложен подход к функциональному моделированию процессов деятельности выпускающей кафедры CASE-средствами, путем построения модели «как должно быть», с целью оптимизации и информатизации процессов управления подразделением.
Ключевые слова: функциональное моделирование, бизнес-процессы, документооборот кафедры, контекстная диаграмма, диаграмма декомпозиции, CASE-средство.
Annotation: offered approach to functional modeling of processes of activity of the issuing Department of CASE tools by building a model "to-be" for the purpose of optimization and information management processes unit.
Key words: functional modeling, business process, workflow of the Department, context diagram, decomposition diagram, CASE-tool.

Введение. Организационное управление структурным подразделением Университета – кафедрой, является сложноформализуемым процессом, находящимся в постоянной динамике, поэтому его моделирование, с целью определения эффективных механизмов, является важным этапом повышения эффективности функционирования всего ВУЗа.

Цель исследования. Провести изучение и описание структуры процессов деятельности кафедры с использованием методов функционального моделирования в соответствии со стандартом IDEF0 на основе CASE-средства – Ramus Educational (инструментального средства анализа, предназначенного для построения и анализа модели предметной области), с целью выявления процессов, требующих оптимизационных решений.

Основная часть. Для описания процессов деятельности кафедры, предлагается методология, представленная стандартом IDEF0 [1] (с 2000 г. является стандартом функционального моделирования Российской Федерации).

Графическая формализация процессов деятельности кафедры ВУЗа осуществлялась в среде Ramus Educational. Ramus Educational – это бесплатный аналог Ramus [3]. Ramus Education может быть использован для создания диаграмм в нотации IDEF0 и DFD, использует формат файлов полностью совместимый с форматом файла коммерческой версии Ramus. Ramus Educational поддерживает импорт/экспорт файлов в формат IDL, таким образом, реализуя частичную совместимость с подобными программами (например, с CA Erwin Process Modeler). Лицензия Ramus Educational запрещает его использование в коммерческих целях [2].

Ramus Educational поддерживает единый формат файлов с локальной версией Ramus. Таким образом, файл созданный в Ramus Educational можно редактировать в локальной версии Ramus и наоборот (за исключением атрибутов, поддерживаемых только в локальной версии Ramus). В системе имеется возможность импорта/экспорта файлов в формат IDL BPWin. Таким образом, обеспечивается частичная совместимость с CA ERwin Process Modeler (в части графических моделей IDEF0) [4].

Применение указанных технологий позволило сконцентрироваться на самих процессах, которые выбирались на основе образовательной концепции Университета и кафедры, с учетом ФГОС 3+.

Дифференцирование информационных потоков представляет собой довольно сложный этап функционального моделирования, который определяет дальнейшую детализацию процессов.

На рис.1. изображена контекстная диаграмма «Деятельность кафедры». По стандарту IDEF0, основная диаграмма предполагает наличие входных и выходных информационных потоков, механизмов, исполняющих процесс, и ограничений, накладываемых на процесс. Для процесса «Деятельность кафедры» входным потоком является информация об образовательном уровне – «Образовательный уровень». В качестве выходного потока представлена совокупность результатов деятельности кафедры. Верхние входящие стрелки-дуги определяют управляющие воздействия или ограничения, в нашем случае – «Нормативная

документация». Нижние входящие стрелки-дуги являются исполняющими механизмами («Технологии» и «Персонал»).

Результатами деятельности кафедры в современных условиях в модели «как должно быть» определены: «Стратегический план развития», «Результаты научной деятельности», «Образовательно-квалификационный уровень (бакалавр/магистр)» «Образовательный уровень высшей квалификации» и «Образовательно-квалификационный уровень дополнительного образования», «Инновации» [2].

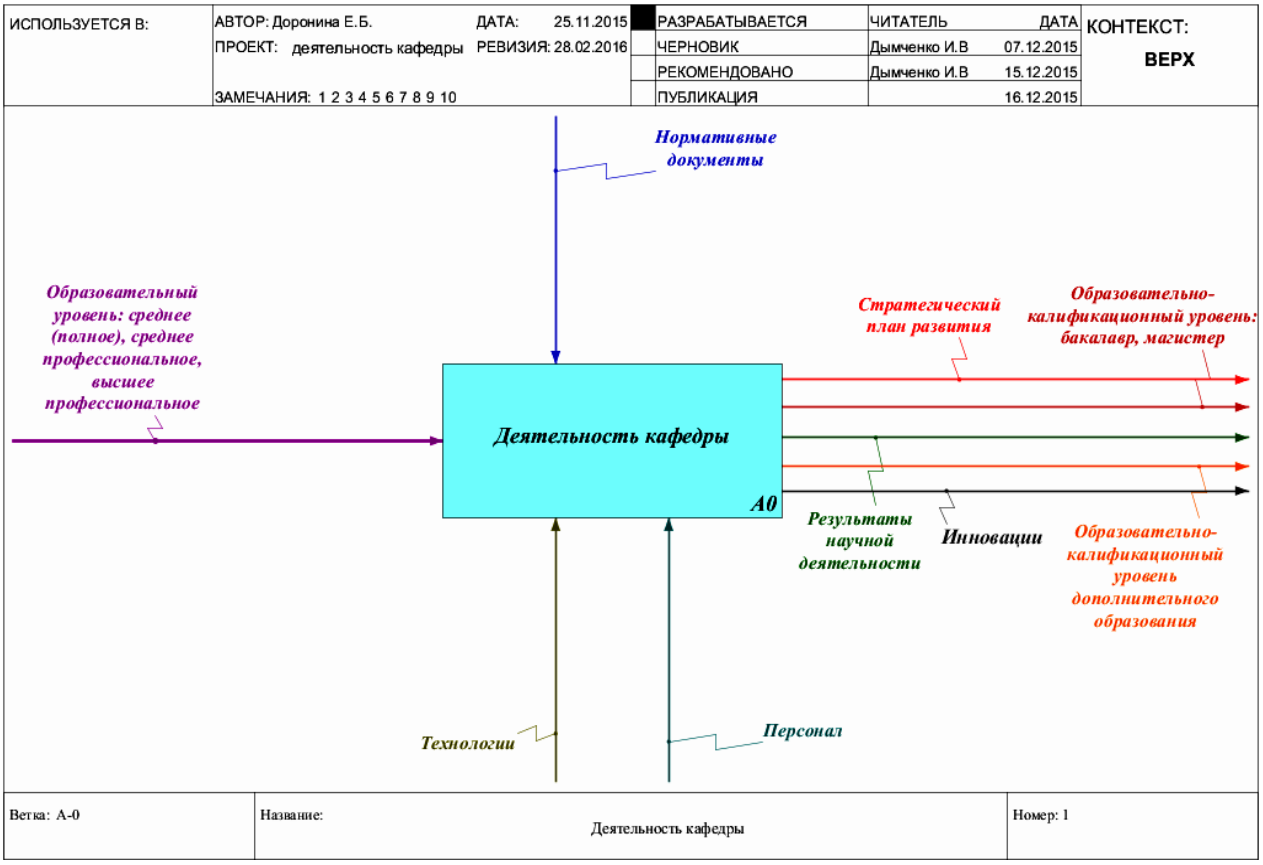


Рис. 1. Контекстная IDEF0 диаграмма «Деятельность кафедры» A-0.

На рис.2. представлена декомпозиция диаграммы «Деятельность кафедры». Так, для класса «Персонал» были введены подклассы: «Профессорско-преподавательский состав», «Инженерно-технический состав», «Учебно-вспомогательный состав». Каждый из указанных подклассов делится, в свою очередь, на экземпляры. Например, подкласс «Профессорско-преподавательский состав» включает в себя такие экземпляры, как: «Ассистент», «Преподаватель», «Старший преподаватель», «Доцент», «Профессор», «Заведующий кафедрой». Если эти экземпляры, в целом, логически ясны и не представляют сложности для выявления, то, например, возникающие между ними связи и их влияние на процессы, не всегда очевидны. Следовательно, это является фактором для реализации принципа итерационного проектирования моделей подобного рода, когда связи могут быть добавлены или исключены в процессе построения диаграммы.

Следующий этап графической формализации процесса представлен на рис.3. Показано формирование связей процесса «Образовательная деятельность», а также приведено уточнение функций основной деятельности кафедры.

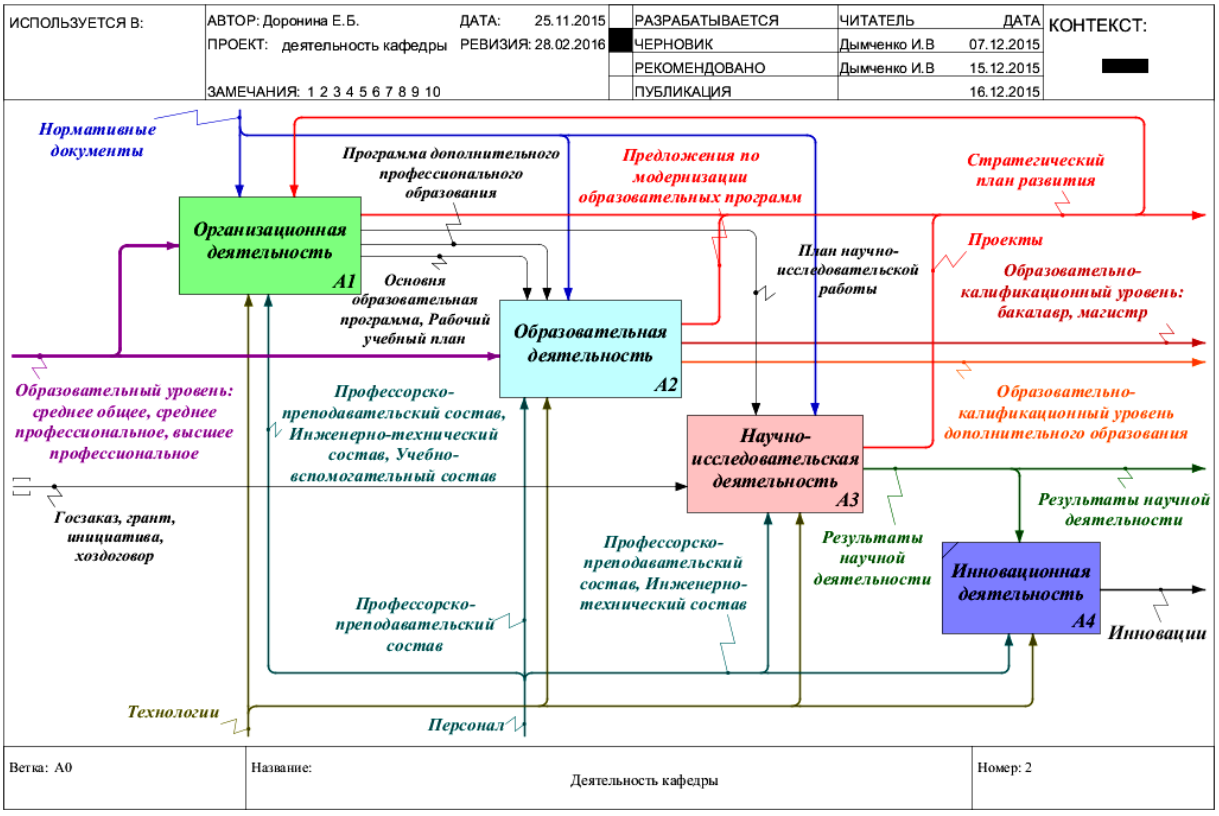


Рис. 2. Диаграмма декомпозиции A0 контекстной диаграммы «Деятельность кафедры».

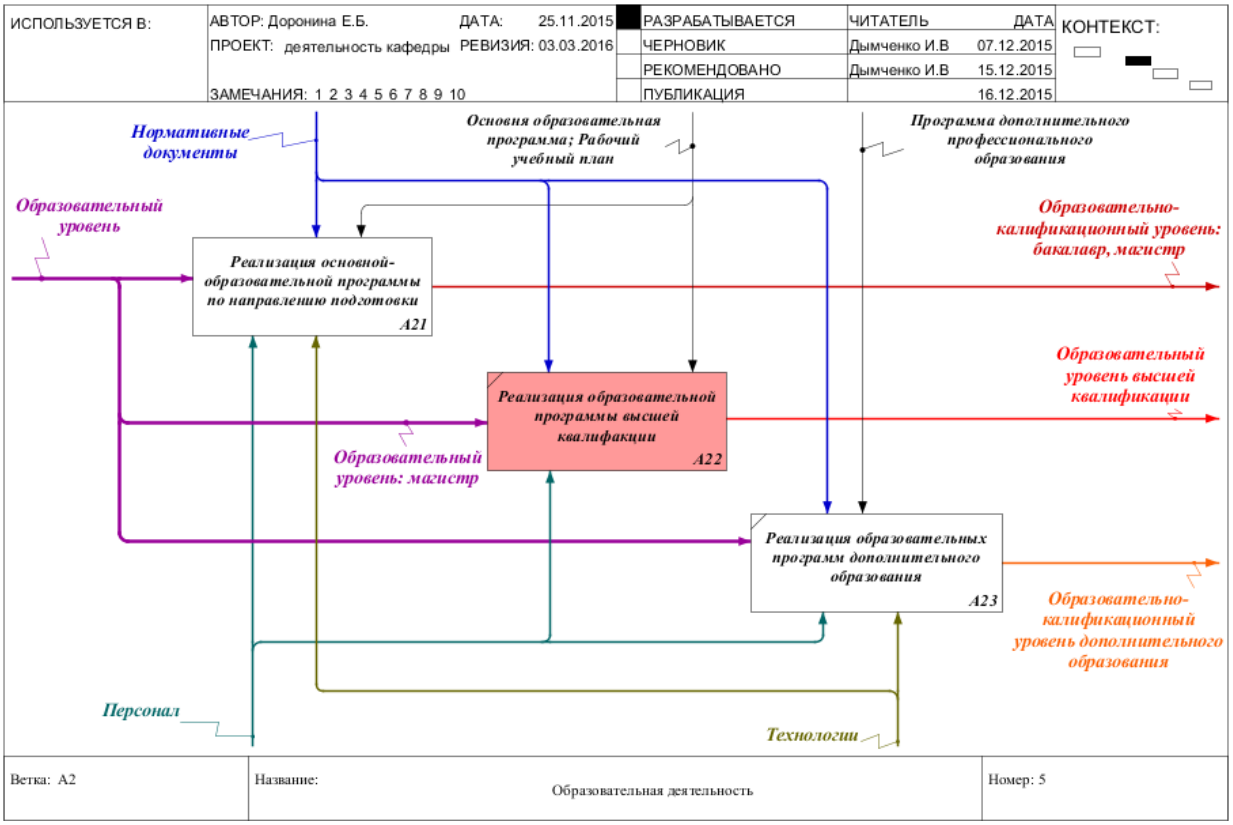


Рис. 3. Диаграмма декомпозиции функционального блока «Образовательная деятельность»

Следует отметить, что представленная функциональная модель «Деятельность кафедры» используется в рамках учебного проекта «Деятельность кафедры, в ходе которого

применяется гибкая методология управления – Канбан (реализация в виде Web-ресурса https://trello.com). Канбан-доска является одним из инструментов отслеживания состояния выполнения задач при ведении проекта, позволяющим реализовать принцип «точно в срок»[5].

На рис.4 представлен фрагмент рабочего интерфейса проекта «Деятельность кафедры» – этап решения задачи создания облака для хранения документации проекта.

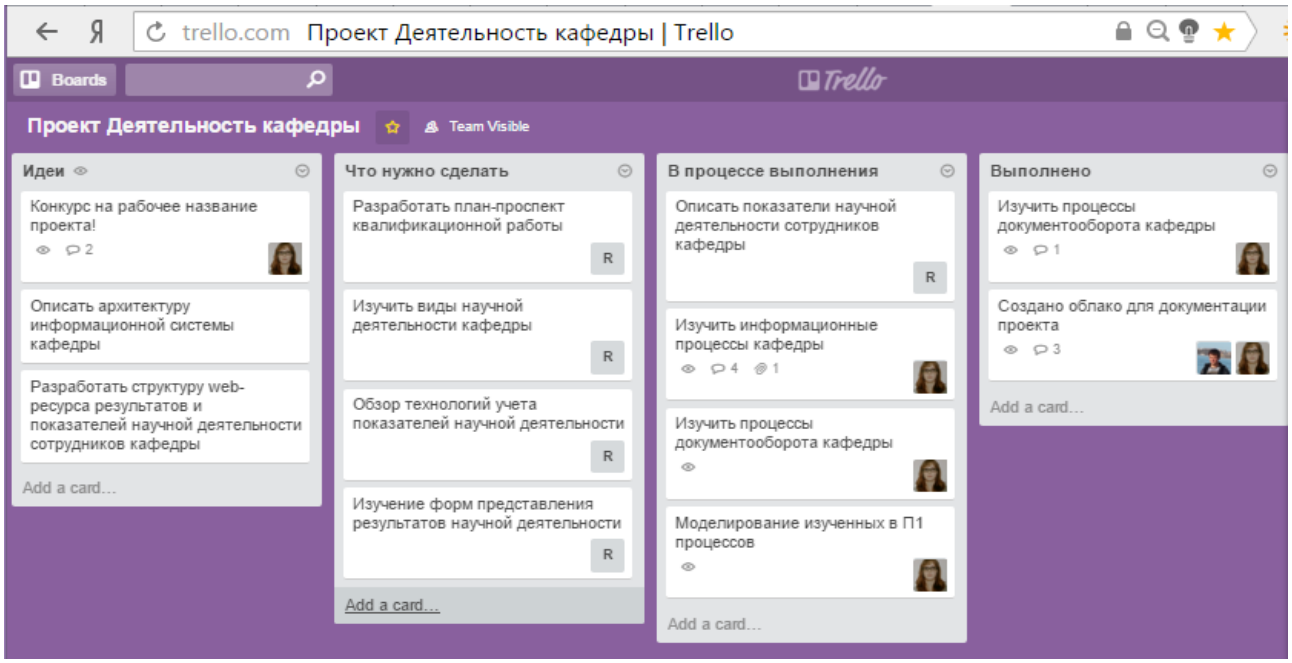


Рис. 4. Канбан-доска для управления проектом «Деятельность кафедры»

Выводы и дальнейшее направление исследований. В процессе исследований в рамках проекта были изучены основные процессы деятельности кафедры. Функциональное моделирование позволило построить модели вида «как должно быть» и формализовать процессы организационной, образовательной, научной и инновационной деятельности.

Далее планируется итерационное уточнение полученной модели, по мере дальнейшей декомпозиции структурно-функциональных блоков, и описание информационных потоков при помощи методологии DFD – диаграмм потоков данных и методологий семейства IDEF – IDEF3, IDEF1X. Это позволит исследовать возможности повышения эффективности деятельности кафедры и выявить процессы, требующие оптимизации, посредством обеспечения их информационными технологиями.

Библиографический список

1. РД IDEF0–2000 Руководящий документ. Методология функционального моделирования IDEF0. ИПК Издательство стандартов, 2000. [Электронный ресурс] Режим доступа – http://www.nsu.ru/smk/files/idef.pdf.
2. Федеральный закон от 29.12.2012 N 273-ФЗ (ред. от 30.12.2015) "Об образовании в Российской Федерации". [Электронный ресурс]. Режим доступа – https://www.consultant.ru/document/cons_doc_LAW_140174/.
3. Официальный русскоязычный сайт проекта Ramus. Режим доступа – http://ramussoftware.com/.
4. Ramus 1.2.5 (Моделирование бизнес-процессов) [Электронный ресурс]. Режим доступа – http://www.twirpx.com/file/132655/.
5. Kanban Guide: Demand Scheduling for Lean Manufacturing, Compiled by Nilesh R Arora. Add ValueConsulting Inc., India 2001, p.11.

УДК 004.415.28
В.С. Коржов, магистрант
Научный руководитель: В.И. Шевченко, к.т.н., доцент
Севастопольский государственный университет

КОМПЬЮТЕРНАЯ СИСТЕМА ДЛЯ ИССЛЕДОВАНИЯ ПРОЦЕССОВ СКЛАДСКОЙ ЛОГИСТИКИ

Аннотация: рассматривается логистическая система с четырьмя потоками данных по типам складирования (малогабаритная упаковка, крупногабаритная упаковка, упаковка требующая использования холодильного оборудования, негабаритная упаковка). Реализована компьютерная система автоматизации складского учета с элементами складской логистики. С целью повышения эффективности работы в систему встроен модуль имитационного моделирования.
Ключевые слова: логистическая система, автоматизация складского учета, имитационная модель, поток данных.
Annotation: logistics system with four streams of data on the storage type (small-size packaging, large-sized packaging, packaging requiring the use of refrigeration equipment, oversized packaging). The computer system of automation of inventory control with elements of logistics warehouse was developed. The system is integrated simulation module to improve the efficiency of its work.
Key words: logistics system, automation of inventory control, simulation model, the data stream.

Значительная роль в современном бизнесе принадлежит логистике, которая позволяет при оптимальных затратах и экономии материальных и финансовых ресурсов достигать поставленных корпоративных целей. В России и за рубежом, особенно в последние годы, выделяются такие функциональные направления логистики, как транспортировка, складирование, управление запасами, упаковка, расширение сервисных заказов и др. Складская логистика является самостоятельной областью знаний, имеющей свой предмет и объект исследования и относящаяся к основным логистическим активностям. Ведущей областью приложения ее усилий является сфера товародвижения, поскольку со склада начинаются и складом завершаются все логистические потоки товародвижения [1,2]. Основной целью складской логистики инновационно-сервисного типа является стремление к понижению общих, совокупных расходов в логистических потоках товародвижения на базе минимизирования риска вероятных утрат при осуществлении инновационного и сервисного потенциалов логистической системы [3,4]. На сегодняшний день разработаны и активно развиваются компьютерные системы автоматизации складского учета, в том числе и на платформе 1С [5,6].

В рамках проводимого исследования разработана компьютерная система автоматизации складского учета на платформе 1С:Предприятие 8.3, которая позволяет вести удаленную инвентаризацию объектов (складов). В системе реализованы следующие бизнес-функции: приход товаров на склад; формирование штрих кода товара; списывание товаров; формирование отчетов по остаткам и движению товаров. В качестве дополнительного функционала реализована логистическая опция карты склада, которая позволяет пользователю определить расположение товара на складе и распределить товарно-материальные ценности (ТМЦ) по складским площадям. Фрагмент интерфейса программной системы приведен на рисунке 1. С целью повышения эффективности работы компьютерной системы (минимизации времени распределения ТМЦ) в среде AnyLogic. был реализован модуль имитационного моделирования, позволяющий исследовать работу компьютерной системы при произвольном многомерном потоке данных о складируемых грузах. Распределение грузов осуществлялось с использованием альтернативных алгоритмов: Дейкстры, Флойда и Флойда-Уоршалла.

а) *б)* *с)*

Рис.1. Элементы интерфейса системы автоматизации складского учета (а –мобильное приложение; б – карта склада; с –рабочий стол пользователя)

1. Постановка задачи

Модуль моделирования должен выполнять следующие функции: имитировать процесс ожидания упаковок для размещения на складе; имитировать процесс размещение товаров на складе по заданному алгоритму; собирать статистические данные по размещенным и неразмещенным упаковкам ТМЦ. При проектировании модуля необходимо учитывать следующие особенности логистической системы: упаковки отличаются по типу и времени на обработку; неразмещенные упаковки должны быть убраны из очереди, так как они будут задерживать поток.

2. Анализ алгоритмов размещения ТМЦ и построение модели

В рамках данного раздела проведен краткий обзор алгоритмов связанных с эффективным размещением товарно-материальных ценностей на территории склада, которые были использованы при разработке системы автоматизированного складского учета. Фрагмент реализации алгоритма обхода графа приведен на рисунке 2.

Алгоритм Дейкстры нахождения кратчайшего пути. *Алгоритм* находит кратчайшее *расстояние* от одной из *вершин графа* до всех остальных и работает только для графов без ребер отрицательного веса. Каждой вершине приписывается *вес* – это *вес* пути от начальной вершины до данной. Также каждая *вершина* может быть выделена. Если *вершина* выделена, то *путь* от нее до начальной вершины кратчайший, если нет – то временный. *Обходя граф*, *алгоритм* считает для каждой вершины *маршрут*, и, если он оказывается кратчайшим, выделяет вершину. Весом данной вершины становится *вес* пути. Для всех соседей данной вершины *алгоритм* также рассчитывает *вес*, при этом ни при каких условиях не выделяя их. *Алгоритм* заканчивает свою работу, дойдя до конечной вершины, и весом *кратчайшего пути* становится *вес* конечной вершины. Базовые шаги алгоритма:

Шаг 1. Всем вершинам, за исключением первой, присваивается *вес* равный бесконечности, а первой вершине – 0.

Шаг 2. Все вершины не выделены.

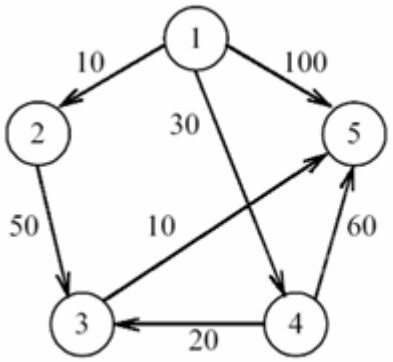
Шаг 3. Первая *вершина* объявляется текущей.

Шаг 4. *Вес* всех невыделенных вершин пересчитывается по формуле: *вес* невыделенной вершины есть минимальное число из старого веса данной вершины, суммы веса текущей вершины и веса *ребра*, соединяющего текущую вершину с невыделенной.

Шаг 5. Среди невыделенных вершин ищется *вершина* с минимальным весом. Если таковая не найдена, то есть *вес* всех вершин равен бесконечности, то *маршрут* не существует. Следовательно, *выход*. Иначе, текущей становится найденная *вершина*. Она же выделяется.

Шаг 6. Если текущей вершиной оказывается конечная, то *путь* найден, и его *вес* есть *вес* конечной вершины.

Шаг 7. Переход на шаг 4.



Итерация	*S*	*w*	D[2]	D[3]	D[4]	D[5]
начало	{1}	–	10	∞	30	100
1	{1, 2}	2	10	60	30	100
2	{1, 2, 4}	4	10	50	30	90
3	{1, 2, 4, 3}	3	10	50	30	60
4	{1, 2, 4, 3, 5}	5	10	50	30	60

Массив **P**: | ⊠ | 1 | 4 | 1 | 3 |

Кратчайший путь из 1 в 5: {1, 4, 3, 5}

Рис.2. Граф и массив вершин

Алгоритм Флойда нахождения кратчайших путей между парами вершин. Рассмотрим помеченный орграф, который содержит времена или трудоемкости размещения ТМЦ, от точки входа склада до места размещения. Необходимо построить таблицу, где приводилось бы минимальное перемещения ТМЦ от одного допустимого места размещения до альтернативного. В этом случае мы сталкиваемся с общей задачей нахождения кратчайших путей, то есть нахождения кратчайших путей между всеми парами вершин орграфа.

Формулировка задачи. Есть ориентированный граф $G(V,E)$, каждой дуге (ребру) (v,w) этого графа сопоставлен неотрицательный вес C_{vw}. Общая задача нахождения кратчайших путей заключается в нахождении для каждой упорядоченной пары вершин (v,w) любого пути от вершины v в вершину w, длина которого минимальна среди всех возможных путей. В алгоритме Флойда-Уоршалла добавлена динамическая составляющая.

Построение имитационной модели. Входными данными является заявка системы (упаковка). Время обработки определяет тип заявки.

Входные переменные модели:

- интенсивности λ поступления запросов (по типам упаковки);
- интенсивность μ обслуживания.

Выходные (контролируемые) переменные:

- $N_{раз}$ – количество размещенных товаров;
- $N_{нераз}$ – количество неразмещенных.

Система может быть представлена моделью СМО с многомерным потоком заявок. Структура системы изображена на рисунке 3.

Описание блоков:

1. Приход разно-габаритных упаковок.

Данный блок имитирует поставку товаров в организацию и при получении приходного складского ордера. Время между поступлением товаров регулируется при помощи слайдера. Параметры: *Tobr* – время обработки упаковки. *Tn, Tm, Tk, T*h – время между прибытиями упаковки.

2. Ожидание и разгрузка.

Блок имитации ожидания погрузки товаров для дальнейшего размещения их на складе. Параметры: *t(delay)* - время обработки заявки. *agent.Tobr* – время таймаута ожидания заявки в очереди.

3. Размещение упаковок на складе.

Блок имитации процессов размещения упаковок на складе в соответствии с заданным алгоритмом. Параметры: *Numbermethod* – номер алгоритма, регулируется слайдером. *t(delay)*

- время обработки заявки по алгоритму. *agent.Tobr2* – время таймаута ожидания заявки в очереди.

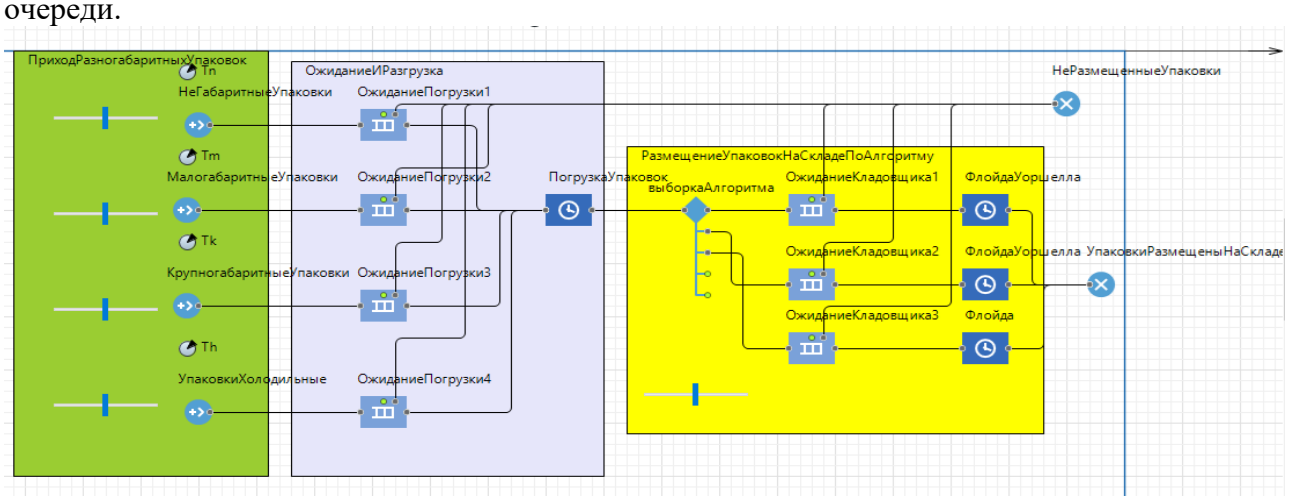


Рис.3. Структура системы

3. Исследования системы моделирования

Было проведено исследование имитационной модели складской логистической системы при произвольном многомерном потоке данных о складируемых грузах. В качестве законов распределения потоков данных о складируемых грузах для каждого потока рассматривались: равномерное распределение, нормальное распределение, гамма распределение. По результатам исследований получены таблицы зависимости количества обработанных и не обработанных заявок от входных параметров и закона распределения поступления заявок в систему. Самым эффективным оказалось равномерное распределение с входными параметрами: μ-23, tn- 7, tm – 8, tk – 7, th – 9, при этом количество размещенных упаковок - 796, количество неразмещенных – нет.

В рамках дальнейших исследований системы будут проведены эксперименты с изменением входных параметров и составлением полно-факторного эксперимента с целью оптимизации складского учета по времени размещения ТМЦ. По результатам экспериментов планируется построение матриц эффективных решений и сведение этих матриц в многомерных OLAP-куб для поддержки принятия решений в задачах складской логистики.

Библиографический список

1. Воронов В. И. Методологические основы формирования и развития региональной логистики: Монография. – Владивосток: Изд-во Дальневосточного университета, 2003. – 316 с.
2. Черняк И. Логистика склад и сервис. Известия Иркутской государственной экономической академии, №3 2004, С.23-27.
3. Гордон М.П. Рынок и логистика. Учебное пособие/ под ред. М. П. Гордона. – М.: Экономика, 2013. – 143 с.
4. Тутова М. Эффективность складской логистики. Современные инновации. Выпуск № 1 (1), 2015. [Электронный ресурс] URL: http://cyberleninka.ru/article/n/effektivnost-skladskoy-logistiki (дата обращения: 17.11.2015).
5. Кудинов А. CRM: Российская практика эффективного бизнеса / А. Кудинов, М. Сорокин, Е. Голышева. – М.: ООО «1С-Паблишинг», 2008. – 430 с.: ил
6. Айвазян С.А., Мхитарян В. С. Прикладная статистика. Основы эконометрики. — М.: ЮНИТИ, 2001. с.93-100.
7. Олифер В. Г., Олифер Н.А. Компьютерные сети: Принципы, технологии, протоколы. — СПб.: Питер, 1999. с.33-51.

УДК 681.3.06
А. С. Кузьмин, студент 2-го курса; Н.Г. Вильсон, студентка 2-го курса.
Научный руководитель: Ю.И. Папкова, канд. физ.-мат. наук, доцент.
Севастопольский государственный университет

ИСПОЛЬЗОВАНИЕ МАТЕМАТИЧЕСКИХ ПАКЕТОВ ПРИКЛАДНЫХ ПРОГРАММ ДЛЯ СПЕЦИАЛЬНОСТИ «ЭЛЕКТРОНИКА И НАНОЭЛЕКТРОННИКА»

Аннотация: Внедрение компьютерного моделирования с использованием математических пакетов прикладных программ в учебный процесс для специальности «Электроника и наноэлектронника».
Ключевые слова: MatLab, Mathematica, Maple, MathCad, интерполяция, аппроксимация.
Annotation: The introduction of computer modeling using mathematical applied program packages in the educational process for the specialty "Electronics and nanoelectronics".
Key words: MatLab, Mathematica, Maple, MathCad, interpolation, approximation.

В настоящее время при разработке и апробации электронных устройств требуется множество трудных и длительных вычислений. С этой целью использование математических пакетов прикладных программ MatLab, Mathematica, Maple, MathCad является основными программными средствами, необходимыми для создания хорошего математического обеспечения. Математический пакет MathCad [1] создан разработчиками как инструмент для работы расчетчиков – инженеров, позволяющий справляться с длительными вычислениями, встречающимися в работе. Главным достоинством пакета MathCad является простота в использовании, поэтому при начальном освоении математических пакетов, а также при решении задач с помощью численных методов студентами младших курсов рекомендуется использование пакета MathCad. MatLab [2], Mathematica [3], Maple [4] могут использовать студенты старших курсов, которые имеют базовые знания в программировании.

Использование компьютерного моделирования очень актуально на сегодняшний день, особенно при решении задач профессиональной направленности. Первый шаг компьютерного моделирования заключается в формулировке математической модели или подходящего математического метода. Например, при обработке экспериментальных данных часто возникает задача аппроксимации результатов эксперимента аналитической функцией, которую можно использовать в последующих расчетах. Рассмотрим эксперимент: источник тока исследуется путем подключения к нему различных резисторов, измеряется сила тока и напряжение на клеммах источника, получении следующий ряд значений:

$$\begin{pmatrix} I, A \\ U, B \end{pmatrix} = \begin{pmatrix} 0.06 & 0.11 & 0.125 & 0.16 & 0.22 & 0.23 \\ 4.8 & 4.8 & 4.4 & 4.1 & 4.0 & 3.3 \end{pmatrix}$$

Возникает вопрос. Чему равно напряжение при других (промежуточных) значениях силы тока? Эту задачу можно решить построением интерполяционного полинома, который легко реализуется в любом математическом пакете, где для получения интерполяционного значения в точке используется соответствующая функция.

Компьютерное моделирование применяется для вычисления задач, которые требуют точности и решаются за приемлемое время. К числу таких задач следует отнести задачи, в которых необходимо подобрать для удовлетворения заданной точности значения входящих параметров. Например, скорость электрона ν, движущегося в электрическом поле с разностью потенциалов U равна:

$$\nu = c\sqrt{1 - \frac{1}{k + m \cdot U}},$$

где c – скорость света.
Значения параметров k и m можно найти с помощью аппроксимации, используя метод нелинейного регрессионного анализа. В данном методе процесс получения наилучших параметров аппроксимирующей функции является итерационным, поэтому его удобнее всего реализовать в любом математическом пакете прикладных программ.

Выбор походящего математического метода при компьютерном моделировании начинается с выделения тех факторов, которые следует принять во внимание. Например, рассмотрим следующую задачу: найти напряжение на туннельном диоде при решении следующего уравнения:

$$\frac{E-U}{P} - I(U) = 0,$$

где $I(U)$ - вольтамперная характеристика туннельного диода.

Данное трансцендентное уравнение можно решить с помощью численных методов в два этапа: а) отделение корней; б) уточнение корня. Оба этапа реализуется в математических пакетах.

Во многих задачах факторы, которые следует принять во внимание, связаны с условием равенства сил или выполнением каких-либо законов. В частности, при решении задач по электротехники для расчета цепи постоянного тока применяется первый и второй законы Кирхгофа. Используя данные законы, получаем систему линейных алгебраических уравнений, которую можно решить с помощью метода Гаусса или матричным методом. Метод Гаусса является точным методом, у него нет ошибки метода, но, если число обусловленности матрицы системы намного больше единицы, то матрица системы является плохо обусловленной матрицей и решений системы может быть найдено с большими ошибками. Поэтому в этом случае использование метода Гаусса становится нецелесообразным. Если матрица системы трехдиагональная, то можно применять метод прогонки, в противном случае используют итерационные методы (метод простой итерации, последовательных приближений и т.д.), реализующиеся в математических пакетах.

Дифференциальные уравнения применяются для компьютерного моделирования широкого круга задач. Выбор метода и реализация решения краевой задачи (задачи Коши) для дифференциального уравнения зависит от типа уравнения. Вследствие универсальности и наличия хорошо разработанной теории чаще всего применяют разностные методы, если возможно точные методы решения дифференциального уравнения. Рассмотрим в качестве примера задачу об определении напряженности статического электрического поля в бесконечном металлическом желобе квадратного поперечного сечения. Верхняя стенка желоба имеет потенциал, остальные стенки заземлены и находятся под нулевым потенциалом. Найдем функцию, описывающую распределение потенциала в прямоугольной области. Известно из электростатики, что если в замкнутом объеме отсутствуют объемные электрические заряды, то распределение потенциала удовлетворяет уравнению Лапласа. Найдем решения поставленной задачи с помощью метода разделения переменных в математическом пакете MathCad.

Найти решение $U(x, y)$ задачи Дирихле в квадрате со стороной 1 для уравнения Лапласа

$$\frac{\partial^2 U}{\partial x^2} + \frac{\partial^2 U}{\partial y^2} = 0$$ с краевыми условиями вида:

$$U(0, y) = 0 \quad (0 \le y \le 1), \quad U(1, y) = 0 \quad (0 \le y \le 1),$$
$$U(x,0) = 0 \quad (0 \le x \le 1), \quad U(x,1) = f(x) \quad (0 \le x \le 1).$$

$f(x) := x + 6 \qquad L := 1 \qquad n := 10 \qquad m := 10$

$$hx := \frac{L}{n} \qquad hy := \frac{L}{m}$$

$$U(i,j) := 4 \cdot \frac{f(i \cdot hx)}{\pi} \cdot \sum_{k=1}^{30} \left[\frac{\sin\left[\frac{(2 \cdot k + 1) \cdot \pi \cdot i \cdot hx}{L}\right] \cdot \sinh\left[\frac{(2 \cdot k + 1) \cdot \pi \cdot j \cdot hy}{L}\right]}{2 \cdot (k + 1) \cdot \sinh\left[\frac{(2 \cdot k + 1) \cdot \pi \cdot L}{L}\right]} \right]$$

$i := 1..9 \qquad j := 1..9$

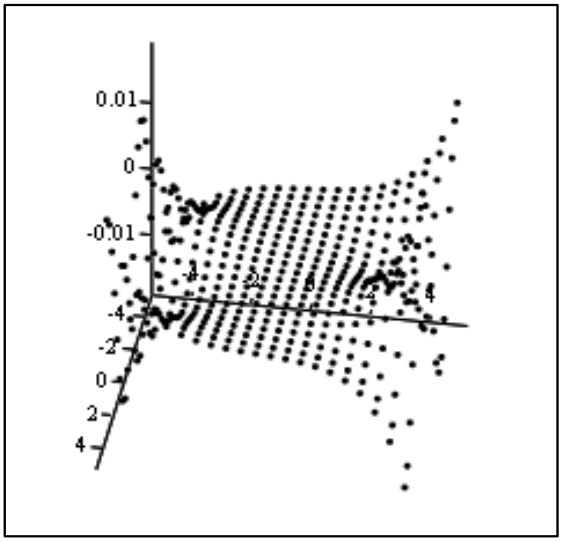


Рис.1

Внедрение компьютерного моделирования с использованием математических пакетов прикладных программ в учебный процесс для специальности «Электроника и наноэлектроника» значительно облегчит решения задач, где необходимы громоздкие и трудоемкие вычисления. В результате решения задач в математическом пакете получаем наглядный, точный и быстрый ответ.

Библиографический список

1. Дьяконов В.П. Mathcad 8-12 для всех. М: Солон пресс, 2015.
2. Дьяконов В.П. MATLAB 6.5 SP1/7.0 + Simulink 5/6 в математике и моделировании. М: Солон пресс, 2014.
3. Дьяконов В.П. Mathematica 5.1/5.2/6 в математических и научно-технических расчетах. М: Солон пресс, 2008. - 744 с.
4. Дьяконов В.П. Maple 10/11/12/13/14 в математических расчетах. М: ДМК пресс, 2015.

УДК 004.942
И.В. Макаров, студент 4-го курса
Научный руководитель: О.В. Ченгарь, к.т.н., доц.
Севастопольский государственный университет

ИССЛЕДОВАНИЕ СТРУКТУРЫ АВТОМАТИЗИРОВАННОГО СКЛАДА ДИСКРЕТНОГО ПРОИЗВОДСТВА ДЛЯ ПОСТРОЕНИЯ КОМПЬЮТЕРНОЙ МОДЕЛИ

Аннотация: в статье рассмотрены вопросы системного анализа организационной структуры автоматизированного склада, исследования его параметров и разработки математической и графовой моделей с целью дальнейшего применения инструментальных возможностей муравьиных алгоритмов для решения задачи оптимизации складского расписания.
Ключевые слова: Автоматизированный склад, дискретное производство, метод муравьиных колоний.
Annotation: the article discusses problems of system analysis of the organizational structure of the automated warehouse, researching its parameters and developing mathematical and graph model for further application of instrumental possibilities of ant algorithms for solving the problem of optimization of warehouse schedule
Key words: Automated warehouse, discrete manufacturing, ant colony algorithm.

Необходимость разработки компьютерной подсистемы моделирования работы автоматизированного склада, как неотъемлемой части системы управления производством, обусловлена интенсификацией использования современных информационных технологий в промышленности [1].

Несмотря на существенное продвижение в области исследования методов и моделей оптимизации производственного расписания [2], и, как следствие, складской организации, задача моделирования структуры автоматизированного склада с целью дальнейшего усовершенствования его производительности является актуальной.

В рамках решения данной задачи проблема оптимизации времени перемещения погрузчика и размещения элементов хранения на складе является одной из важнейших проблем в производстве. В дискретном производстве существенное влияние при планировании оказывают различного рода, как прогнозируемые, так и непредвиденные обстоятельства, принуждающие к полной либо частичной остановке производства, такие как сбои и отказы оборудования, проведение профилактического ремонта, переналадка оборудования и т.д., поэтому всевозможные события подобного рода должны быть обязательно учтены при составлении производственного расписания.

Вследствие этого необходимо иметь возможность перераспределять и пересчитывать складское расписание с учетом минимизации потери времени, без остановки работы склада и производства в целом.

Таким образом, рассматриваемая проблема состоит в отсутствии на текущий момент оптимального решения задачи размещения производственных элементов на складе и оптимизации действий погрузчика по критерию длительности перемещения, минимальному количеству действий и минимизации избыточных перемещений погрузчика.

При всем многообразии методов решения задач комбинаторной оптимизации, приближенные (метаэвристические) методы, хоть и уступают специальным в точности, являются более гибкими и масштабируемыми [3]. Одним из наиболее перспективных методов решения задач с подобной проблематикой является применение метода муравьиных колоний - одного из наиболее эффективных полиномиальных алгоритмов для нахождения приближенных решений задач поиска маршрутов на графах. Основной идеей подхода является анализ и использование модели поведения муравьев, ищущих пути от одной точки к другой и маркирующих наиболее удачные пути большим количеством феромона. Для использования инструментальных возможностей муравьиного алгоритма необходимо представление процесса функционирования автоматизированного склада в виде графа, описывающего последовательность перемещения погрузчика и размещения элементов хранения.

В качестве объекта исследования в данном случае принимается автоматизированный склад в дискретном производстве – территория или помещение, предназначенное для хранения материальных ценностей и оказания складских услуг, оснащенное автоматизированными средствами для приема, выдачи и распределения элементов в зоне хранения (рис.1).

Рис. 1. Структурная схема автоматизированного склада

Основными элементами автоматизированного склада являются кран-штабелер, зоны хранения, приемки и отгрузки материальных элементов [1,4]. Кран-штабелеры подразделяются на мостовые и стеллажные. Количество кран-штабелеров в автоматизированном складе может варьироваться в зависимости от способа организации склада.

Зона хранения обычно состоит из n-го числа стеллажей, каждый из которых разделяется на боле мелкие элементы – ячейки. Каждая ячейка характеризуется следующим набором параметров:

- номер стеллажа;
- номер ячейки;
- тип хранимого элемента (заготовки, инструмент, готовая продукция);
- габариты ячейки: высота, ширина, глубина;
- флаг наполняемости (занято, свободно).

Также осуществляется разделение зоны хранения на отсеки, в зависимости от характеристик хранимых элементов (размер, тип, технические характеристики).

Зоны отгрузки и выгрузки территориально могут находиться как в разных частях склада, так и иметь одно окно приема-передачи хранимых элементов

«Узким местом» управления производством в данном случае являются различного рода отхождения от нормы временных параметров автоматизированного склада, загруженность транспортного модуля (перегруженность/недогруженность), неоптимальное распределение задач для кран-штабелера.

Как любую модель, модель автоматизированного склада можно представить в виде «черного ящика» (рис.2), входными характеристиками которого являются график поставки, время выполнения операций и стоимость выполнения складских операций. Выходными показателями будут показатели эффективности использования ресурсов и временные затраты на размещение и хранение запасов материалов и готовой продукции.

Рис. 2. Модель автоматизированного склада в форме «черного ящика».

Процесс функционирования автоматизированного склада, в данном случае, рассматривается как совокупность операций, производимых над объектами хранения технологическим оборудованием. Все операции, выполняемые кран-штабелером, можно укрупненно разделить на операции погрузки/выгрузки, перемещения грузов, профилактического ремонта, а также запуска и остановки.

Описание элементов производственного процесса на складе и моделирование их функционирования сводится к формальному описанию ячеек склада и моделированию работы погрузчика (рис.3).

Рис. 3 – Алгоритм функционирования складского оборудования

Несмотря на разнообразие типов складов, отличающихся друг от друга функциональным назначением, уровнем автоматизации и другими параметрами, в них выделяют общие признаки:

1. Процесс функционирования автоматизированного склада разделяется на некоторые структурные элементы – складские операции над определенным типом материальных ценностей с учетом установленного порядка их хранения согласно требованиям.
2. Структурно любой склад состоит из конечного множества ячеек для хранения и технологического оборудования, с помощью которого реализуются все складские операции.
3. Все возможные складские машины функционируют, взаимодействуя друг с другом, придерживаясь некоторого алгоритма, при котором их свойства и пути передвижения зависят от взаимно-определяемых условий.
4. Все складские операции являются конечными по времени.
5. Склад имеет зону приемки и зону отгрузки, соответственно для приема и выдачи заготовок и рабочих инструментов (может быть одним структурным элементом).
6. Склад функционирует во времени и не является изолированным элементом, взаимодействуя с внешней средой через прием готовой продукции и выдачу заготовок и инструмента.
7. Автоматизированный склад включает ограниченное множество структурных связей R {r1, r2… rk} между складскими операциями с использованием технологических машин.
8. Маршруты перемещения крана-штабелера есть множество возможных маршрутов M {M1, M2…Mk}
9. Время выполнения различных складских операций определяется множеством T {t1, t2… tk}
10. Перечень ячеек для хранения материальных ценностей есть множество W{w1, w2 … wk}
11. Последовательность приема-хранения-передачи элементов является детерминированной, т.е. учет материальных ценностей является фиксированной последовательностью, задаваемой подсистемой оперативного планирования.

В ходе исследования функционирования автоматизированного склада дискретного производства выделены его структурные элементы и определены их важнейшие характеристики, которые являются основой для построения компьютерной модели. Последовательность выполнения складских операций во многом играет ключевую роль при оптимизации управления технологическими процессами транспортирования и складирования, поэтому важнейшим элементом разрабатываемой модели является алгоритм функционирования складского оборудования.

Библиографический список

1. Гаврилов Д.А. Управление производством на базе стандарта MRP II СПб.: Питер, 2008. - 416 с. - ISBN: 978-5-91180-709-2 (2-е издание)
2. Синица Л.М. Организация производства. Учебник для вузов: ИВЦ Минфина, 2008; ISBN: 978-985-6847-77-9
3. Лореш М.А., автореферат «Разработка и исследование алгоритмов муравьиной колонии: Препринт. Омск, ОмГУ для решения задач оптимального размещения предприятий», 2006г., 113с.
4. Ченгарь О.В. Графоаналитическая модель загрузки гибких производственных модулей автоматизированного технологического участка машиностроительного предприятия / О.В. Ченгарь, Е.О. Савкова // Вісник Східноукраїнського національного університету імені Володимира Даля. Науковий журнал. – Луганськ, 2011 – № 13(167). – С. 239-245.

УДК 517.9

А.А. Начевина, студентка 4-го курса, Е.В. Радевич, аспирантка
Научный руководитель: А.С. Миненко, доктор физ.-мат.наук, профессор
Донецкий национальный технический университет

ЧИСЛЕННОЕ МОДЕЛИРОВАНИЕ ПРОЦЕССА КРИСТАЛЛИЗАЦИИ

Аннотация: Исследуется одна задача Стефана с учетом конвекции в жидкой фазе. Построено приближенное решение этой задачи с использованием малого параметра. Управление процессом осуществляется с применением нечеткой логики.
Ключевые слова: функционал, кристаллизация, тепловой поток, управление
Annotation: The Stefan problem with convection in the liquid phase is investigated. The approximate solution is constructed, using the method of small parameter. The control over this process with the use of fuzzy logic is realized.
Key words: functional, crystallization, thermal stream, controlling

Распространение тепла в различных средах оказывает большое влияние на характер протекания многих важных для практики процессов. Среди задач, связанных с распространением тепла, выделяется класс задач, в которых исследуемое вещество переходит из одной фазы в другую с выделением или поглощением тепла.

Целью данной работы является моделирование процесса кристаллизации металла, изучение процесса завершения получения слитка в кристаллизаторе путем его вытягивания.

Рассматривается задача управления информационными процессами при автоматизации технологий тепловой обработки металла, на основе математического моделирования, анализа статистических данных и теплофизических экспериментальных измерений. В качестве источника информации исследуется математическая модель, основанная на пространственной задаче Стефана, с учетом конвективного движения и примесей в жидкой фазе.

Постановка задачи. Пусть D = (-1< x < 1, y < 0) - полуполоса, заполненная твердым металлом. Обозначим через u (x, y) температуру этого металла. Требуется определить температуру u (x, y) по следующим условиям:

$u_{xx} + u_{yy} + \omega u_y = 0, (x, y) \in D,$ (1)

$u_x \pm \omega_0 u = 0, x = \pm 1, -\omega < y < 0,$ (2)

$u(x, -\infty) = 0,$ (3)

$u_y(x, 0) = v(x), -1 \le x \le 1,$ (4)

Здесь ω и ω_0 – постоянные, соответственно, число Пекле и Нуссельта.

Решение задачи (1)-(4) имеет вид

$$u(x,y) = \sum_{n=0}^{\omega} \frac{\cos \lambda_n x^{\mu_n Y}}{\mu(1+\omega_0 \frac{\cos^2 \lambda_n}{\lambda_n^2})} \int_{-1}^{1} v(\zeta) \cos \lambda_n \zeta d\zeta, \quad (5)$$

где $\mu = -\frac{\omega}{2} + \sqrt{\frac{\omega^2}{4} + \lambda_n^2}$, n = 1,2,3,...$\lambda_n$ - положительные корни уравнения $\lambda = \omega_0 ctg\lambda$.

Отождествим теперь температуру u (x, y) с температурой твердого слитка находящегося в кристаллизаторе при электрошлаковом переплаве. Для вытягивания слитка из кристаллизатора поверхность слитка предварительно обогревается тремя электронными лучами W1, W2 и W3, причем мощность W3 одного из низ равномерно распределена в центральной зоне { -1 ≤ x ≥ 1, y = 0}, а два других сконцентрированы по краям x = ±1 [1]. Независимо от того, в каком отношении находится температура поверхности слитка с критической температурой $T^к$, при которой поверхность слитка отделяется от стенок кристаллизатора, теплообмен слитка с кристаллизатором осуществляется по формуле (2). Для получения температуры слитка достаточно положить в формуле (5) v(x) = (W_1, W_2, W_3).

Далее введем в рассмотрение функционал:

$$I(v) = \int_{n}^{0} (u(1,y) = T^k)^2 dy \quad (6)$$

Рассматривается задача. Требуется определить поток v(x) из допустимого множества U, доставляющий наименьшее значение функционалу I(v). Минимизирующая последовательность v_n строится по формуле $v_{n+1} = v_n + e_n(v_{n-1} - v_n)$, параметр e_n выбирается из условия $\min I(v_n + e_n(v_{n-1} - v_n))$, $0 \le e_n \le 1$ (2). В качестве области определения функции U берется множество кусочно-постоянных ступенчатых функций:

$$v = v_k, x_k \le x \le x_{k+1}, v_k = const, k = 0,1,2,...,m.$$

При этом формула (5) имеет вид:

$$u(x,y) = \sum_{n=0}^{\omega} \frac{\cos\lambda_n x e^{\mu_n y}}{\mu_n(1+\omega_0 \frac{\cos^2\lambda_n}{\lambda_n^2})} \sum_{k=0}^{m} v_k \frac{\sin\lambda_n x_{k+1} - \sin\lambda_n x_k}{\lambda_n},$$

$$aI(v) = I(v_0, v_1, v_2, ..., v_m).$$

При численной реализации задачи необходимо учесть ограничение $2500 \le v(x) \le 5000$, здесь v(x) – мощность потока в единицах МВт/м^2, а также ω = 2,66, ω = 3,05.

Способы решения задачи. Нулевое приближение. Найдем минимум функционала (6), в случае когда $u_0(x,y) = f_0(x,y)v$, где $f_0(x,y) = 2\frac{\cos\lambda_0 x \sin\lambda_0}{\lambda_0\mu_0(1+\omega_0\frac{\cos^2\lambda_0}{\lambda_0^2})} e^{\mu_0 y}$.

Минимум функционала (6) находим из условия

$$\frac{\partial I}{\partial v} = 0.$$

Непосредственные вычисления показывают, что

$$v_0 = 4T^* \frac{\mu_0\lambda_0(1+\omega_0\frac{\cos^2\lambda_0}{\lambda_0^2})}{\sin 2\lambda_0},$$

$$I(v_0) = v_0^2 \frac{\sin 2\lambda_0(1-e^{2\mu_2 H})}{2\lambda_0^2\mu_0^3(1+\omega_0\frac{\cos^2\lambda_0}{\lambda_0^2})} - (T^*)^2 H - 2T^* v_0 \frac{\sin 2\lambda_0(1-e^{\mu_0 H})}{\lambda_0\mu_0^2(1+\omega_0\frac{\cos^2\lambda_0}{\lambda_0^2})}.$$

Первое приближение. Найдем теперь минимум функционала (6), в случае когда $u_1(x,y) = (f_0(x,y))v$, где $f_1(x,y) = 2\frac{\cos\lambda_1 x e^{\mu_1 y}}{\mu_1(1+\omega_0\frac{\cos^2\lambda_1}{\lambda_1^2})}\frac{\sin\lambda_1}{\lambda_1}$.

Поступая, аналогично тому, как это было сделано в случае нулевого приближения, получим:

$$v_1 = 2T^*[\frac{\sin 2\lambda_0(1-e^{\mu_1 H})}{\lambda_0\mu_0^2(1+\omega_0\frac{\cos^2\lambda_0}{\lambda_0^2})} + \frac{\sin 2\lambda_2(1-e^{\mu_2 H})}{\lambda_1\mu_1^2(1+\omega_0\frac{\cos^2\lambda_1}{\lambda_1^2})}]A,$$

$$A = \frac{\sin^2\lambda_0(1-e^{\mu_2 H})}{\lambda_0\mu_1^2(1+\omega_0\frac{\cos^2\lambda_0}{\lambda_0^2})^2} + \frac{\sin^2 2\lambda_1(1-e^{2\mu_1 H})}{2\mu_1^2(1+\omega_1\frac{\cos^2\lambda_1}{\lambda_1^2})^2} +$$

$$+2\frac{\sin 2\lambda_0 \sin 2\lambda_1}{\mu_0\lambda_0(1+\omega_0\frac{\cos^2\lambda_0}{\lambda_0^2})}\frac{(1-e^{\mu_0 H})(1-e^{\mu_1 H})}{\mu_0\lambda_1(1+\omega_0\frac{\cos^2\mu}{\lambda_1^2})}.$$

Далее, имеет место следующая формула:

$$I(v_1)=v_2^1\frac{\sin^2 2\lambda_0(1-e^{2\mu_0 H})}{2\mu_0^3\lambda_0^2(1+\omega_0\frac{\cos^2\lambda_0}{\lambda_1^2})^2}+v_1^2\frac{\sin^2 2\lambda_1(1-e^{2\mu_1 H})}{2\mu_1^3\lambda_1^2(1+\omega_0\frac{\cos^2\lambda_1}{\lambda_1^2})^2}+$$

$$+2v_1^2\frac{\sin 2\lambda_0 \sin 2\lambda_1(1-e^{\mu_1 H})(1-e^{\mu_0 H})}{\mu_1\lambda_1(1+\omega_0\frac{\cos^2\lambda_1}{\lambda_1^2})\mu_0\lambda_0(1+\omega_0\frac{\cos^2\lambda_0}{\lambda_0^2})}-$$

$$-2T^* v_1[\frac{\sin 2\lambda_1(1-e^{\mu_1 H})}{\lambda_1\mu_1^2(1+\omega_0\frac{\cos^2\lambda_1}{\lambda_1^2})}+\frac{\sin 2\lambda_0(1-e^{\mu_0 H})}{\lambda_0\mu_0^2(1+\omega_0\frac{\cos^2\lambda_0}{\lambda_0^2})}]-H(T^*)^2.$$

Приближение любого порядка. Аналогичным образом можно исследовать минимум функционала $I(v_n)$, когда

$$u_n(x,y)=2\frac{v_n\sin\lambda_0\cos\lambda_0 x(1-e^{\lambda_0 H})}{\lambda_0\mu_0[1+\omega_0\frac{\cos^2\lambda_k}{\lambda_k^2}]}e^{\mu_0 y}+2\omega_0 v_n\sum_{k=1}^{n}\frac{\cos\lambda_k x\cos\lambda_k(1-e^{\mu_k H})}{\lambda_k^2\mu_k[1+\omega_0\frac{\cos^2\lambda_k}{\lambda_k^2}]}e^{\mu_k y}.$$

Оценить погрешность предлагаемого метода вычисления минимума функционала (6) можно, используя следующее утверждение.

При достаточно малых значениях ω и при $(x,y)\in\overline{D}$ справедлива оценка:

$$|\sum_{k=n+1}^{\infty}\frac{\cos\lambda_k x\cos\lambda_k(1-e^{\omega_k H})}{\lambda_k^2\mu_k[1+\omega_0\frac{\cos^2\lambda_k}{\lambda_k^2}]}e^{\mu_k y}|\le\sum_{k=n+1}^{\infty}\frac{1}{(k\pi)^2}e^{\mu_k y}.$$

При доказательстве этого утверждения воспользоваться соотношением:

$\lambda_k=n\pi+\varepsilon_n$, где $\varepsilon_n\to 0$ для всех n[2]

Справедливо также утверждение. Пусть выполнены условия $\omega_0\ge\omega\sqrt{2}tg\omega\sqrt{2}$, $0<\omega\le A$, $0<\omega\le\frac{\pi^2}{16}$, $0<v_0\le v(x)<v_1$, при $x\in[-1,1]$, где v_0 и v_1 – некоторые постоянные. Тогда решение краевой задачи (1)-(5) удовлетворяет следующим условиям при $(x,y)\in\overline{D}$:

$u_y(x,y)\le C_1\omega\exp(\mu_0 y)\le C_1\omega\exp(\omega y)$

$C_0=\exp(\mu_0 y)\le u(x,y)\le C_1\exp(\mu_0 y)C_1\exp(\omega y)$,

где $C_1=\frac{6+A(1+\cos\sqrt{A})}{3(1+\cos\sqrt{A})}$, $C_0=\frac{3(1-A)(1+\cos^2\sqrt{A})\cos^2\sqrt{A}}{6+A(1+\cos^2\sqrt{A})}$.

Для утверждения необходимо сравнить с помощью принципа максимума функции $u_y(x,y)$ и $v_y(x,y)$, где $v(x,y)$ – решение задачи (1)-(5) в предположении, что

$v_y(x,0)=v_1$ при x $\in[-1,1]$. Далее, рассматривается функция $f(x,y)=v_y(x,y)-u_y(x,y),(x,y)\in\overline{D}$ и доказывается, что $f(x,y)$ в $\overline{D}$.

Действительно, функция $f(x,y)$ не может принимать наименьшее отрицательное значение внутри D в силу принципа максимума. На вертикальных частях границы x = ±1 функция $f_x(x,y)$ также не может принимать отрицательный минимум. В такой точке имели бы $f_x(x,y)<0$, между тем $f_x(x,y)=-\omega_0 f(x,y)>0$, x=±1, так как $f(x,y)<0$, по предложению. На бесконечности функция $f(x,y)$ исчезает, т.е. $f(x,-\infty)=0$ На границе y = 0, -1≤ x ≤ 1 имеет $f(x,0)=v_y(x,0)-u_y(x,0)=v_1--v(x)\geq 0$ Следовательно, всюду в $\overline{D}$ справедливо неравенство $u_y(x,y)\leq v_y(x,y)$ при (x,y)$\in\overline{D}$. Отсюда с помощью интегрирования по переменной y следует оценка для функции u(x,y) сверху. Аналогичным образом, можно получить оценку на производную u_2(x,y) сверху при (x,y)$\in\overline{D}$. Полученные оценки позволяют оценить температуру u(x,y) и тепловой поток внутри области D не прибегая к решению задачи (1)-(4) [3-4].

Численные результаты решения задачи представлены в таблице 1.

Таблица 1- Численные результаты при различных значениях параметров

ω_0	λ_0	λ_1	T	H	v_0	v_1	$I(v_0)$	$I(v_1)$
1,5	0,9882	3,5422	0,9	-3,0	2,783	4,367	1,011	21,997
1,5	0,9882	3,5422	0,9	-4,0	2,783	5,730	5,428	38,576
1,5	0,9882	3,5422	0,9	-6,0	2,783	6,588	-3,038	62,490
1,5	0,9882	3,5422	0,95	-3,0	2,937	4,610	1,126	24,509
1,5	0,9882	3,5422	0,95	-4,0	2,937	5,730	-0,559	42,982
1,5	0,9882	3,5422	0,95	-6,0	2,937	6,954	-3,385	69,627
1,5	0,9882	3,5422	0,9	-3,0	4,5	4,5	16,728	23,709
1,5	0,9882	3,5422	0,9	-4,0	4,5	4,5	15,256	24,355
1,5	0,9882	3,5422	0,9	-6,0	4,5	4,5	12,397	25,275
1,5	0,9882	3,5422	0,95	-3,0	4,5	4,5	15,537	23,062
1,5	0,9882	3,5422	0,95	-4,0	4,5	4,5	13,866	23,695
1,5	0,9882	3,5422	0,95	-6,0	4,5	4,5	10,718	24,696
1,5	0,9882	3,5422	0,9	-3,0	11,1	11,1	173,683	192,101
1,5	0,9882	3,5422	0,9	-4,0	11,1	11,1	175,717	198,788
1,5	0,9882	3,5422	0,9	-6,0	11,1	11,1	173,351	202,940

Библиографический список

1. Патон Б.Е. Избранные труды. – Киев: Институт электросварки им. Е.О. Патона НАН Украины, 2008.-893 с.

2. Шевченко А.И., Миненко А.С. Методы исследования нелинейных моделей, -Киев: Наук. Думка, 2012.-132 с.

3. Миненко А.С. Вариационные задачи со свободной границей. – Киев: Наук. Думка, 2005.-341 с.

4. Шевченко А.И., Миненко А.С., Сыпко И.А. Моделирование одного класса сложных с нечеткими управлениями // Доп. НАН Украины.- 2013. - №8. – С. 52-54.

УДК 577
А.С. Папкова, магистрант.
Научный руководитель: А.Ф.Хрусталев, д.ф.-м.н., профессор.
Севастопольский государственный университет.

КОМПЬЮТЕРНЫЕ ТЕХНОЛОГИИ В ИССЛЕДОВАТЕЛЬСКИХ ЗАДАЧАХ

Аннотация: На примерах исследовательских задач по химии, показано использование компьютерных технологий, а именно на языке программирования Pascal.
Ключевые слова: задачи с избыточными данными, химические формулы, Pascal, исследовательские задачи, алгоритм решения.
Annotation: In the examples of research problems in chemistry, show the use of computer technology, namely, the Pascal programming language.
Key words: problems containing redundant data, chemical formulas, Pascal, research tasks, solution of algorithm.

В «Концепции развития математического образования в РФ» (далее – Концепция), утвержденной правительством 24 декабря 2013 года, подчеркнуто: «Студенты, изучающие математику, включая информационные технологии, должны уделять значительно больше времени, чем в настоящее время, решению творческих учебных и исследовательских задач». Ниже приведем пример задачи по химии, которая может решаться двумя вариантами.

Задача 1. Определите молекулярную формулу углеводорода, если массовая доля углерода в нем равна 82,8%, а плотность этого вещества составляет 2, 59 г/л [2, с. 6].

При её решении запишем общую формулу углеводорода в виде:

$$C_xH_y \qquad (1)$$

Вариант 1. Исключив из условия информацию о массовой доле углерода, решим задачу в новой формулировке.

Выведите молекулярную формулу углеводорода, плотность которого при (н.у.) составляет 2, 59 г/л. Исходя из этого условия, найдем относительную молекулярную массу углеводорода (ОММ) M_r = 58 (2,59·22,4 – округлено до целого) и составим уравнение для определения натуральных чисел x и y:

$$12x+y = 58 \qquad (2)$$

Поскольку в предельном углеводороде C_xH_{2x+2} с x атомами углерода соединяется максимальное количество атомов водорода, равное 2x+2, для индексов формулы C_xH_y любого углеводорода справедливо неравенство:

$$y \le 2x+2 \qquad (3)$$

Из уравнения (2) следует, что натуральное число x удовлетворяет неравенству $1 \le x \le 4$, а потому для натурального y из (3) имеем: $1 \le y \le 10$.

Теперь можем составить программу, действующую по следующему алгоритму:

```
program E1;
var x, y: integer;
begin
 for x:=1 to 4 do
  for y:=1 to 10 do
   if 12*x+y=58 then
    writeln('Otvet:' , 'x=',x, 'y=',y);
end.
```

otvet: x=4 y=10. Значит, C_4H_{10} – молекулярная формула углеводорода.

Вариант 2. Определите молекулярную формулу углеводорода, если массовая доля углерода в нем равна 82,8%.

На основании (1) находим x:y = 2 : 5; 5x = 2y; из (3) имеем $5x=2y\le2(2x+2)$; $1\le x\le4$; $1 \le y \le 10$.

```
program E2;
var x, y: integer;
begin
 for x:=1 to 4 do
  for y:=1 to 10 do
```

```
  if 5*x=2*y then
   writeln('Otvet:' , 'x=',x, 'y=',y);
end.
```

otvet: x=2 y=5.Исключить.

otvet: x=4 y=10 . Значит, C_4H_{10} – молекулярная формула углеводорода.

Задача 2. Существует ли такое двузначное число, которое при делении на сумму квадратов его цифр даёт в частном 2 и в остатке 6, а при делении на произведение его цифр даёт в частном 4 и остатке 6? [5, с. 230, задача № 1064].

После исключения из условия информации «двузначное число, при делении на сумму квадратов его цифр даёт в частном 2 и в остатке 6», приведём решение этой задачи, основанное на реализации Pascal– программы, включающей операцию получения целого остатка от деления (mod) [1, с. 121].

```
program E2;
var x, y: integer;
begin
 for x:=1 to 9 do
  for y:=1 to 9 do
   if (10*x+y)mod(x*y)=6 then
    writeln('Otvet:' , 10*x+y);
end.
```

Otvet: 33

Из однозначного ответа 33 следует, что частное от деления на произведение его цифр равно 3. Значит, не существует двузначного числа, которое при делении на произведение его цифр и дает в частном 4 и в остатке 6. Следовательно, не существует и двузначного числа, которое удовлетворяет всем условиям задачи, а потому ответ «22» в учебнике [5] и «решебнике» [3] – ошибочный.

Итак, в учебной литературе встречаются задачи, содержащие избыточные данные. Если при их решении проводить исследования по исключению из условия задачи тех или иных сведений, то можно обнаруживать лишние данные, когда они имеются. Такая организация поисковой и творческой деятельности студентов способствует овладению ими обобщённым умением оперировать знаниями из разных учебных дисциплин: особенно из математики, информатики, химии и получению рациональных и даже оптимальных решений.

Заметим, что исходные данные для многих задач, включая химические, находят с помощью измерений. На эксперименты затрачиваются средства и время. Поэтому сокращение эксперимента имеет и экономическое значение. Именно такая мысль нашла отражение в поэме А. С. Пушкина "Борис Годунов": "Учись мой сын: наука сокращает нам опыты быстротекущей жизни... " Такой подход к решению задач согласуется с Концепцией, которая была разработана с учётом требований инновационной экономики.

Можно было бы продолжить творческую работу над задачами с избыточными данными. Но и уже рассмотренные вопросы ярко демонстрируют могущество компьютерных технологий в решении творческих учебных и исследовательских задач.

Библиографический список

1. Босова Л. Л., Босова *А. Ю.* Информатика: учебник для 8 класса. 2-е изд., испр. М.: БИНОМ. Лаборатория знаний, 2014.
2. Гара Н. Н., Габрусева Н.И. Химия. Задачник с «помощником». 10 —11 классы: пособие для учащихся общеобразовательных учреждений. Изд. 2 – е. — М.: Просвещение, 2013.—79с.
3. ГДЗ по алгебре 9 класс Макарычев И.В.[Электронный ресурс]. Режим доступа: http://gdz.name/reshebnik-2015/gdz-po-algebre-9-klass-makarychev (дата обращения: 2.02.2016г).
4. Концепция развития математического образования в Российской Федерации. [Электронный ресурс]. Режим доступа:http://www.rg.ru/2013/12/27/matematika-site-dok.html (дата обращения: 2 февраля 2016 г).
5. Макарычев Ю.Н., Миндюк Н. Г., Нешков К. И., Суворова С. Б.: Алгебра 9 класс. Учебник для общеобразовательных учреждений. М.: Просвещение, 2014.

УДК 681.3
Ровков А.С., магистрант 2-го курса
Научный руководитель: Е.А.Кожаев, канд. техн. наук, доц.
Севастопольский государственный университет

ИССЛЕДОВАНИЕ СИСТЕМ КОНТРОЛЯ ТЕХНОЛОГИЧЕСКИХ ПРОЦЕССОВ ИЗГОТОВЛЕНИИ ИЗДЕЛИЙ МИКРОЭЛЕКТРОННОЙ АППАРАТУРЫ

Аннотация: В работе рассматриваются принципы организации, основные методы и функции процесса контроля качества технологических процессов при изготовлении изделий микроэлектронной аппаратуры.
Ключевые слова: контроль качества, процедура контроля.
Annotation: The paper deals with the principles of the organization, the basic methods and functions of the process of quality control processes in the manufacture of microelectronic devices.
Key words: simulation, quality control, control procedure.

В процессе производства любых изделий невозможно получить всю продукцию тождественного качества, т.е. параметры различных единиц изделий колеблются в определенных пределах. Это колебание вызывается комплексом случайных и систематических причин, которые действуют в процессе производства и определяют погрешности данного технологического процесса. Если колебание параметров находится в допустимых пределах (в пределах допуска), то продукция является годной, если же выходит за эти пределы - брак.

Качество изготавливаемой продукции определяется качеством исходных продуктов, степенью настроенности оборудования, соблюдением технологических режимов, условиями окружающей среды. Для того, чтобы своевременно выявлять брак и вызвавшие его причины, необходимо осуществлять систематический контроль параметров продукции, получать и обрабатывать данные о контролируемых параметрах. При операциях контроля качества приходится иметь дело с большим числом данных, характеризующих параметры изделия, условия процесса и т.д. При этом, как уже отмечалось, всегда наблюдается разброс данных. Анализируя разброс данных, можно найти решение возникающих в процессе производства проблем, например, причину появления брака.

Систематизация, обработка и исследование большого числа данных с помощью различных методов с целью выявления закономерностей, которым они подчиняются, называются статистической обработкой; данные при этом называются статистическими данными, а применяемые методы - статистическими методами. Обычно для обработки и анализа данных используют не один, а несколько статистических методов. Это иногда позволяет получить ценную информацию, которая при анализе разброса данных только одним методом может ускользнуть.

Система технического контроля (СТК) на производстве – это совокупность средств контроля и исполнителей, взаимодействующих с объектом контроля по правилам, установленным соответствующей документацией. Основная цель – создание условий, при которых существенно снижается выпуск бракованной продукции. Для реализации этой цели на СТК возлагаются следующие функции:

- входной контроль материалов, полуфабрикатов и комплектующих изделий;
- операционный контроль деталей и сборочных единиц в процессе изготовления и испытаний;
- приемочный контроль готовых изделий;
- контроль средств технологического оснащения;
- учет и анализ дефектов.

Классификация методов технического контроля приведена в таблице 1.

Таблица 1- Классификация методов технического контроля

Признаки классификации	**Виды технического контроля**	**Характеристика вида контроля**
По местонахождению контроля	Стационарный	Осуществляется на постоянном специальном рабочем месте для проверки объектов
	Летучий, или скользящий	Осуществляется непосредственно на рабочем месте обработки или сборки путем периодических проверок
По различным стадиям технологического процесса	Предварительный	Применяется с целью предупреждения брака. Предварительному контролю подвергаются материалы, заготовки, полуфабрикаты, детали до начала обработки или сборки
	Промежуточный (операционный)	Осуществляется на различных стадиях изготовления заготовок, деталей и сборки изделий. Основная цель – проверка качества выполнения каждой производственной операции по технологическому процессу и исключение брака на последующих операциях
	Окончательный (заключительный)	Контролю подвергаются все детали, узлы и изделия после заключительных операций технологического процесса обработки или сборки. Наиболее ответственная форма предупреждения выпуска недоброкачественной продукции
По охвату объектов контроля	Сплошной	Проверке подвергаются все без исключения представленные заготовки, детали, узлы и изделия
	Выборочный	Проверке подвергается некоторая часть из партии деталей или изделий в зависимости от условий производства
	Статистический	Наиболее активный способ предотвращения потерь от брака. Применяется для анализа и регулирования качества продукции, хода технологического процесса и состояния производственного оборудования. В основу положен выборочный метод, основанный на теории вероятностей и математической статистике
	Инспекционный	Осуществляется для повторного выборочного контроля объектов, ранее сданных производством и принятых ОТК, а также для выборочных наблюдений за работой изделий в эксплуатации в течение установленного гарантийного срока
По охвату операций контроля	Пооперационный	Производится после каждой операции, когда качество последующей операции зависит от предыдущей
	Групповой	Осуществляется после группы неответственных операций или когда характер технологического процесса исключает возможность проверки объектов после каждой операции
По степени автоматизации	Непрерывный	Применяется непосредственно в процессе обработки деталей. Осуществляется автоматизированный контроль размеров и поднастройка оборудования в процессе обработки
	Полуавтоматический	Контроль размеров обработки постоянно отслеживается с применением ЦОУ (цифровое отсчетное устройство). Подналадка оборудования осуществляется вручную
	Автоматизированный приемосдаточный	Применяется для окончательного контроля изделий с использованием автоматизированных измерительных комплексов
По характеру контроля	Визуальный	Осуществляется только внешним осмотром
	Геометрический	Производится проверка размеров и геометрических элементов объекта

Признаки классификации	Виды технического контроля	Характеристика вида контроля
	Метрологический	Осуществляется проверка элементов, от которых зависит качество (структура, твердость и т.п.)
По назначению контроля	Предупредительный	Осуществляется на всех этапах производства с целью предупреждения брака
	Производственный	Применяется для выявления брака при проверке объектов, предъявленных на контроль после завершения определенного производственного этапа или операции. Различают производственный контроль, выполняемый рабочими, наладчиками и мастерами или работниками ОТК

Для обеспечения требуемого качества продукции необходимо вести контроль не только качества материала и покупных комплектующих изделий, но и соблюдения режимов технологических процессов, контролировать геометрические параметры, качество обработки деталей и сборочных единиц.

Методы контроля можно разделить на две группы: контроль качества с разрушением и без разрушения материала (детали). Контроль качества с разрушением проводится методами химического, спектрального, рентгено-структурного и металлографического анализа. Большая трудоемкость, затраты материала и энергетических ресурсов обусловили применение разрушающих методов контроля только в виде выборочного контроля качества. Неразрушающий контроль качества подразделяется на следующие виды: магнитный, электрический, электромагнитный (вихревых токов), радиоволновый, тепловой, оптический, радиационный, акустический, проникающими веществами (капиллярный). Неразрушающий контроль качества позволяет снизить трудоемкость контрольных операций и повысить производительность труда контролеров, а также получить существенную экономию за счет отбраковки некачественного материала перед его обработкой.

При контроле проводят следующие работы: внешний осмотр невооруженным глазом или с помощью оптических приборов; испытание изделий и агрегатов на стендах, установках; контроль качества поверхности визуально или с помощью средств измерений и контрольно-измерительных приборов; измерения геометрических параметров деталей, узлов, изделий (шероховатость, расположение поверхностей, отклонений от формы, размер); определение толщины металлических и неметаллических листов, труб, профилей, проката, тонкостенных деталей, металлических и неметаллических покрытий физическими методами контроля; обнаружение несплошности материала деталей и узлов (трещин, раковин и т.д.); определение структуры металла, его твердости, прочности, электропроводности, правильности выполнения процесса термообработки.

Библиографический список

1. Савельев А. Я. Конструирование ЭВМ и систем: учебник для вузов / А. Я. Савельев, В. А. Овчинников – 2-е изд., перераб. и доп. – М.: Высшая школа, 1989. – 312 с.
2. Медведев А.М. Надежность и контроль качества печатного монтажа – М. Высшая школа, 1986. – 508 с.
3. Готра З. Ю., Николаев И. М. Контроль качества и надежность микросхем**:** учебник для вузов / А. С. Малика — М. Высшая школа, 1980. – 384 с.
4. Хассан Гома. Проектирование систем реального времени, распределенных и параллельных приложений UML / 1998. – 196 с.

УДК 004:007.3:902

А.В. Романенко, магистрант кафедры «Информационные системы»
А.П. Трухина, магистрант кафедры «Информационные системы»
Научный руководитель: И.П. Шумейко, кандидат физико-математических наук, доцент кафедры «Информационные системы»
ФГАОУ «Севастопольский государственный университет»

ПРИМЕНЕНИЕ СТРУКТУРНО-ФУНКЦИОНАЛЬНЫХ МЕТОДОЛОГИЙ ДЛЯ ОПИСАНИЯ ПРОЦЕССОВ АРХЕОЛОГИЧЕСКИХ ИССЛЕДОВАНИЙ

Аннотация: представлено структурно-функциональное описание процессов археологического исследования по методологиям IDEF0, DFD, BPMN с целью проектирования и создания комплексной информационной технологии.
Ключевые слова: комплексная информационная технология, методологии проектирования, IDEF0, DFD, BPMN, диаграмма, нотация, поток данных, бизнес-процесс.
Abstract: represented by structural-functional description of the process of archaeological research on methodologies IDEF0, DFD, BPMN with the aim of designing and developing integrated information technology.
Keywords: complex information technology, design methodology, IDEF0, DFD, BPMN, diagram notation, the data flow, the business process .

Внедрение математических методов и информационных технологий в археологию началось ещё в начале XX века. В настоящее время накоплен значительный опыт применения математических методов и компьютерных технологий в археологии, имеется достаточное количество публикаций по этим вопросам, по которым можно проследить историю их применения [1-3]. Ускоряющийся рост достижений в области информационных технологий несомненно оказывает влияние на различные отрасли научных исследований, в том числе, и на решение стоящих перед археологией задач. Достаточно широк арсенал средств, которые применяются, либо могут применяться в перспективе на каждом из этапов археологических исследований, включая использование современных ГИС-технологий, возможностей спутникового мониторинга и новейших систем хранения и обработки данных.

Однако в силу сложности технологической цепочки, охватывающей весь процесс археологических исследований (предварительный сбор и систематизация материалов, археологические полевые работы, обработка данных, создание научных отчетов), и отсутствия системного подхода к информатизации в этой сфере существуют значительные трудности в формировании единого научного археологического информационного пространства. В связи с этим использование современных подходов к разработке комплексной информационной технологии, определяющей и регулирующей процессы создания, накопления, представления, обработки и передачи археологической информации, является актуальной задачей.

Целью данной работы является анализ структурных методологий проектирования информационных систем и описание на их основе моделей разрабатываемой комплексной информационной технологии археологических исследований согласно функционально-модульному подходу.

Как известно, существуют два основных подхода к разработке информационных систем, принципиальное различие между которыми обусловлено разными способами декомпозиции систем: структурный (функционально-модульный) и объектно-ориентированный подходы [4]. На этапе проектирования для решения поставленной задачи целесообразно использовать именно структурный подход, в рамках которого система разбивается на функциональные подсистемы, которые, в свою очередь, делятся на подфункции, те – на задачи и так далее до конкретных процедур. При этом не теряется целостность рассматриваемой системы и не возникает проблем при описании информационного взаимодействия отдельных компонентов [5].

Среди методологий структурного анализа выделим следующие: методология функционального моделирования процессов IDEF0 (Integrated Definition Function); методология функционального моделирования потоков данных DFD (Data Flow Diagrams); методология моделирования, анализа и реорганизации бизнес-процессов BPMN (Business Process Modeland Notation).

Применительно к процессам археологических исследований методология IDEF0 дает возможность проанализировать существующие информационные потоки, возникающие в процессе археологического исследования и на основе полученных данных построить модель, адекватную существующим информационным потокам.

На начальном этапе модель представляется в виде одного функционального блока, отображающего основную функцию, называемого контекстной диаграммой А-0. К основному блоку подводятся интерфейсные дуги (потоки), отвечающие за различные объекты, определяющие процессы, происходящие в системе. Каждая сторона блока, к которой подходит интерфейсная дуга, имеет свое определенное значение. Левая сторона: «вход» (input) все, что поступает в процесс или потребляется процессом; верхняя сторона – «управление» (control) или ограничения на выполнение операций процесса; правая сторона: «выход» (output) – или результат процесса; нижняя сторона: «механизм» (mechanism), который используется для выполнения процесса.

Относительно основной функции «Археологическое исследование», на вход контекстной диаграммы поступают предварительные научные материалы (гипотезы, предположения). В качестве управления выступают нормативные документы (положения, инструкции, разрешения), а также методики и процедуры ведения работ. На выходе – научная отчетная документация и другие результаты научно-исследовательской деятельности (публикации, монографии и т.д.). Под механизмом, в данном контексте, понимаются археологи, ведущие исследования, а так же инструменты, которые они используют. Уже на этапе создания диаграммы А-0, целесообразно в качестве механизма ввести в описываемую модель информационные технологии.

Следующий этап описания в рамках методологии IDEF0 предполагает декомпозицию контекстной диаграммы, то есть разбиение сложного процесса на составляющие функции, рис.1. В данном случае были выделены три большие группы процессов: проведение полевых археологических работ, обработка и изучение полученного материала, составление научной отчетной документации.

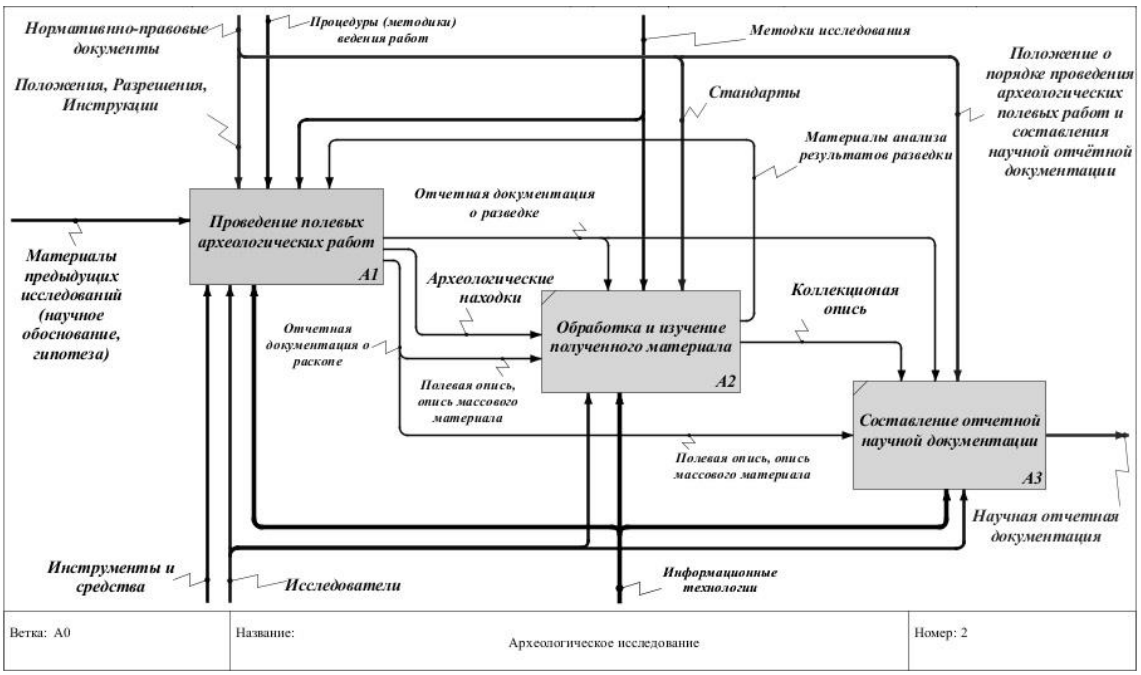


Рис.1. Диаграмма декомпозиции А0.

Каждая из этих групп, должна быть снабжена определенным набором информационных технологий, которые смогут обеспечить, непрерывный сбор (фиксацию и описание),

организацию передачи, обмена и хранения данных, а также обработку материалов исследования.

Дальнейшая детализация процессов дает возможность конкретизировать вид информационной технологии способной автоматизировать многие виды работ, предусмотренных в Положении о порядке проведения археологических полевых работ (археологических раскопок и разведок) и составления научной отчётной документации.

В качестве дополнения модели бизнес-процессов, выполненной в IDEF0 строится диаграмма DFD. Диаграммы потоков данных (Data Flow Diagrams – DFD) представляют собой иерархию функциональных процессов, связанных потоками данных. Цель такого представления – продемонстрировать, как каждый процесс археологического исследования преобразует свои входные данные в выходные, а также выявить отношения между этими процессами. Диаграммы верхних уровней иерархии (контекстные диаграммы) определяют основные процессы или подсистемы с внешними входами и выходами. Они детализируются при помощи диаграмм нижнего уровня. Такая декомпозиция продолжается, создавая многоуровневую иерархию диаграмм, до тех пор, пока не будет достигнут уровень декомпозиции, на котором детализировать процессы далее не имеет смысла [6].

Построение функциональной модели DFD начинается с разработки контекстной диаграммы. На ней отображается основной процесс (сама система в целом) и ее связи с внешней средой (внешними сущностями). Это взаимодействие показывается через потоки данных. Внешняя сущность представляет собой материальный предмет или физическое лицо, представляющее собой источник или приемник информации. Определение некоторого объекта или системы в качестве внешней сущности указывает на то, что она находится за пределами границ анализируемой ИС. На контекстной диаграмме в нотации DFD (рис.2) в качестве сущностей выступают «Исследователь» и «Научный фонд». Основным блоком диаграммы будет выступать система «Археологическое исследование». Поток данных определяет информацию, передаваемую через некоторое соединение от «Исследователя» к «Научному фонду». Поток данных на диаграмме изображается линией, оканчивающейся стрелкой, которая показывает направление потока. Каждый поток данных имеет имя, отражающее его содержание.

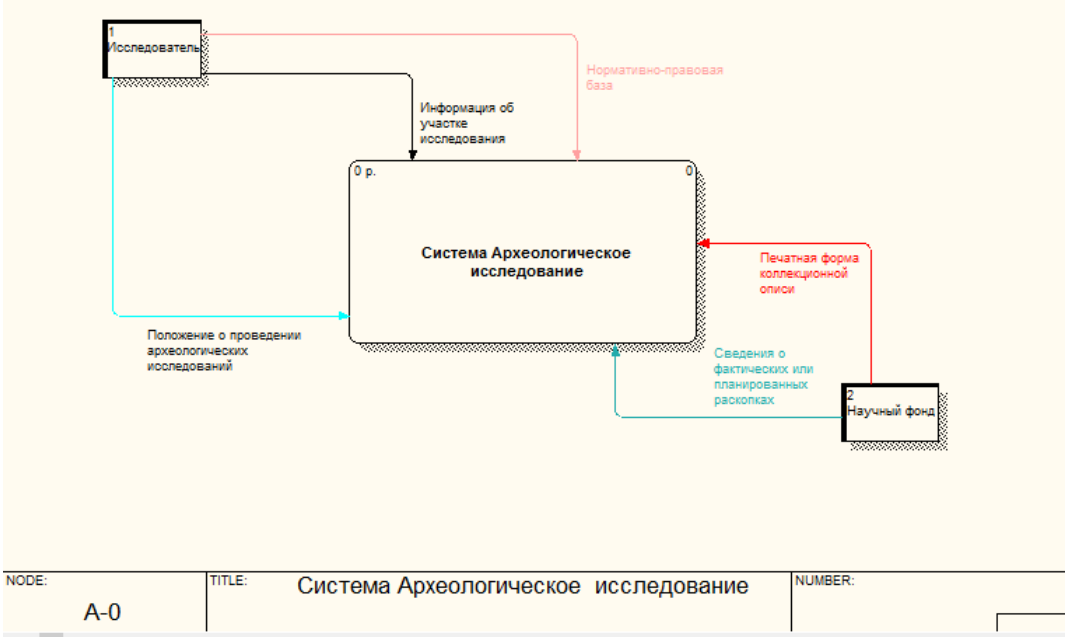


Рис.2. Контекстная диаграмма в нотации DFD.

Следующий этап описания в рамках нотации DFD предполагает декомпозицию контекстной диаграммы, то есть разбиение сложного процесса на составляющие функции,

рис. 3. В данном случае были выделены семь больших групп процессов: получение разрешения на археологические исследования, проведение полевых работ, внесение классификаторов, положений и законов, проведение археологических исследований, подсистема импорта данных, создание коллекционной описи.

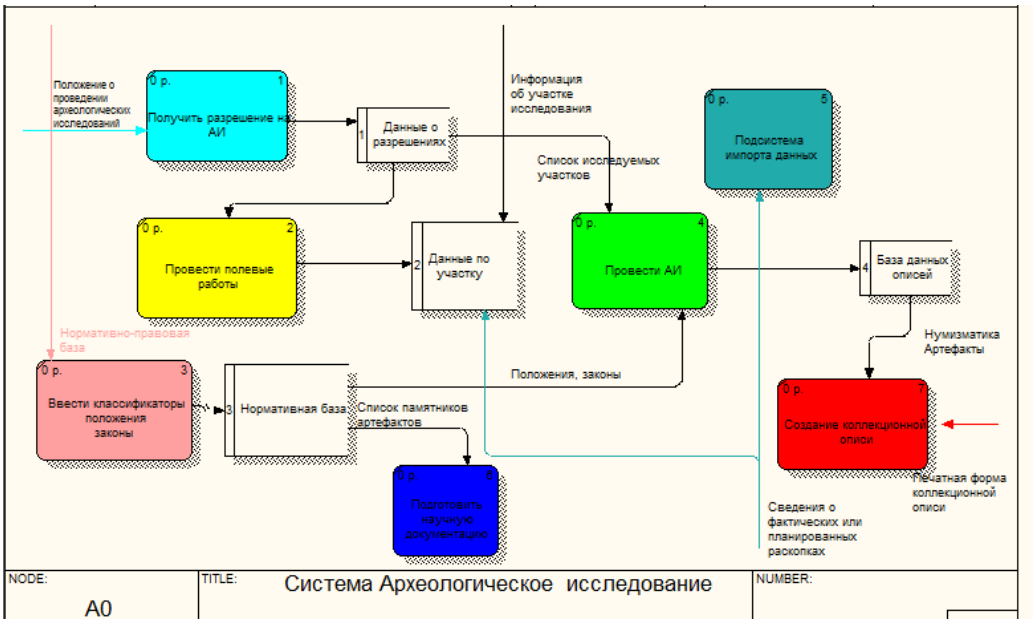


Рис. 3. Диаграмма декомпозиции в нотации DFD.

У каждого процесса (подсистемы) на диаграмме потоков данных должен быть как минимум один входящий и один выходящий поток. Процесс должен запускаться на выполнение либо через обрабатываемый, либо через управляющий поток данных. Работа каждого процесса должна завершаться конкретным результатом. Каждый накопитель данных также должен иметь как минимум один входящий и один выходящий поток. Наличие только входящих потоков в накопитель означает, что информация накапливается, но не используется.

Модель DFD, помимо описания функционального аспекта системы, содержит также сведения об информационном и компонентном аспектах. Совокупность накопителей данных является прообразом будущей БД, т.е. определяет состав и структуру информации. Построение диаграмм с использованием в качестве блоков подсистем показывает состав и связи компонентов будущей системы.

Для получения нотации, легко понимаемой всеми пользователями, воспользуемся спецификацией BPMN. Эта спецификация разработана организацией Business Process Management Initiative (BPMI) в 2001-2004 годах с учётом множества ранее существовавших диаграмм [7].

Диаграмма, описанная в нотации BPMN, представляет собой алгоритм (сценарий) выполнения процесса, а также отображение того, как процесс взаимодействует с другими процессами с точки зрения обмена сообщениями (информацией, документами и другими объектами деятельности). Алгоритм выполнения процесса представляется на диаграмме с помощью элементов потока (событий, процессов, шлюзов), которые связываются между собой потоками управления, определяющими ход выполнения процесса [8].

На начальном этапе модель представляется в виде трех подпроцессов: проведение полевых археологических работ, обработка и изучение полученного материала, составление научной отчетной документации; в исполнении которых заключается археологическое исследование, рис.4. Каждый подпроцесс ассоциируется с базой данных или ее фрагментом. Для обозначения слияния и/или ветвления процессов используются логические операторы.

Следующий этап описания предполагает декомпозицию подпроцессов на действия, рис.5. В данном случае подпроцесс проведения полевых археологических работ был разделен

на три действия: сбор научных сведений, ведение археологической разведки, ведение археологических раскопок.

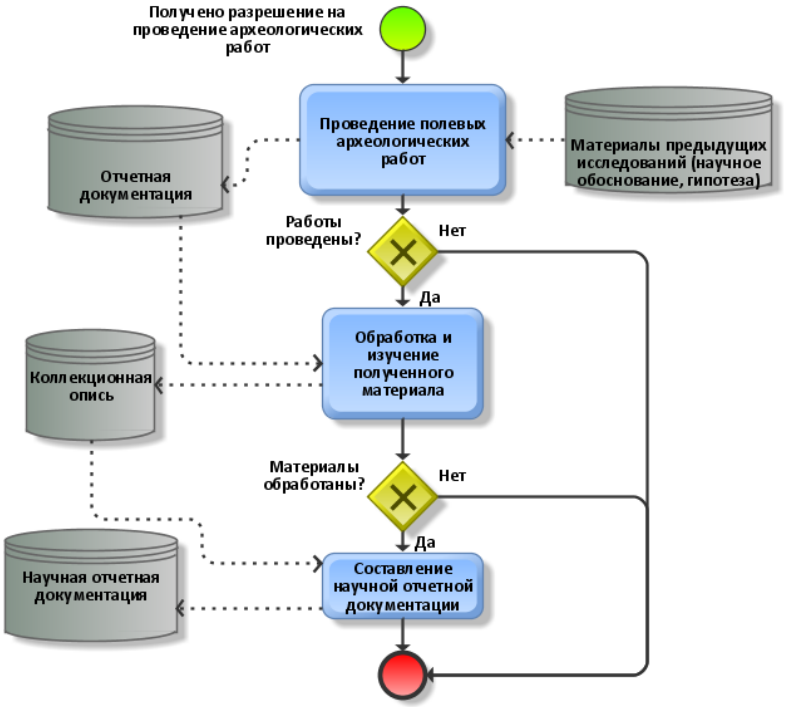


Рис.4. Контекстная BPMN-диаграмма археологического исследования

Каждое из действий осуществляется на основе информации взятой из баз данных. Наличие шлюзов показывает, что в ходе выполнения может быть активирован только один из предложенных маршрутов. Условия пропуска по исходящему маршруту задается рядом с соответствующей линией в виде логического выражения.

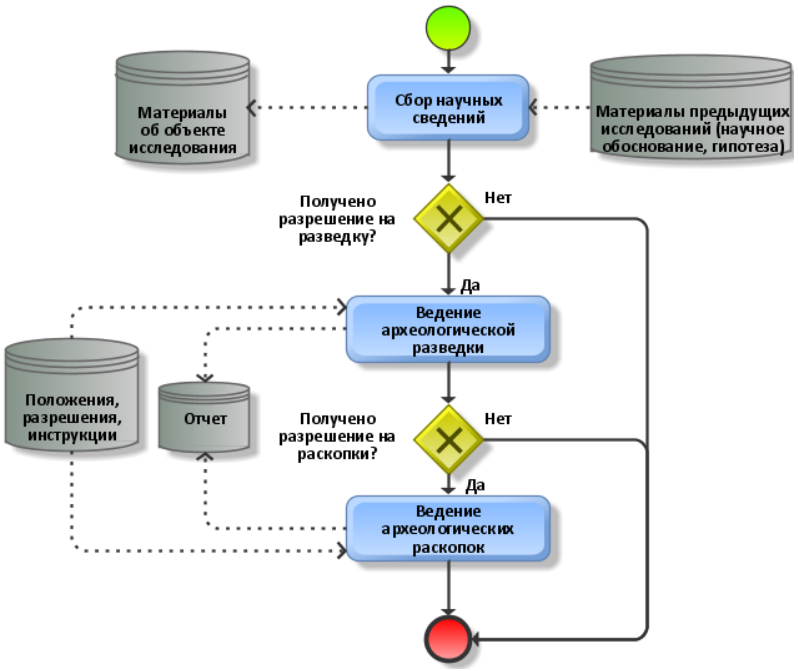


Рис.5. BPMN-декомпозиция подпроцесса «Проведение полевых археологических работ»

Дальнейшая детализация процессов показывает, что все процессы реализуются при взаимодействии с базами данных или их фрагментами. Это означает, что существует

необходимость внедрения технологии, формирующей единой информационное пространство с мобильным доступом.

Выводы

При структурно-функциональном описании процессов археологического исследования говорить о достоинствах и недостатках отдельных нотаций некорректно, так как возможны ситуации, при которых анализ по одной из методологий не обнаружит недостатков в функциональной структуре разрабатываемой системы, однако это не является гарантией отсутствия ошибок. Модели IDEF0 наиболее удобны при построении функциональных моделей, наглядно отражают функциональную структуру объекта: производимые действия, связи между этими действиями. Методология DFD позволяет проанализировать информационное пространство системы и используется для описания документооборота и обработки информации, поэтому диаграммы DFD целесообразно применять в качестве дополнения модели бизнес-процессов, выполненной в IDEF0.

Дальнейшие исследования предполагают детальное изучение процессов археологического исследования путем продолжения структурно-функционального и информационного описания, включающего моделирование предметной области при помощи методологий IDEF1X и моделирование процессов при помощи методологии IDEF3.

Итоговой целью является создание комплексной информационной технологии в рамках пилотного проекта «Археолог-Херсонес», поддерживающей современные тенденции мобильных, геоинформационных, инфокоммуникационных и web-технологий.

Библиографический список

1. Прохоров А. Компьютерные технологии в археологии // Web-сервер журнала КомпьютерПресс [Электронный ресурс]. – 2003. №7 Режим доступа: http://compress.ru/archive/cp/2003/7/3/
2. Васильев, С. Применение компьютеров в археологии / С. Васильев // Статья на сайте "Археология.РУ - Скифика-Кельтика" [Электронный ресурс]. – 2005. – Режим доступа: http://www.archaeology.ru/ONLINE/Vasiljev/pc.html.
3. Баранова Е.В. Очерк применения геоинформационных систем в исторических исследованиях // Лаборатория социальной истории [Электронный ресурс]. -2002. - Режим доступа: http://www.newinsight.msk.ru/index
4. Вендров А.М. CASE - технологии. Современные методы и средства проектирования информационных систем. [Текст] / А.М. Вендров – М.: Финансы и статистика, 2005. – 456 с.
5. Марка Д.А. Методология структурного анализа и проектирования. [Текст] / Д.А. Марка, К. МакГоуэн – М.: МетаТехнология, 2005. – 240 с.
6. Базы данных. Диаграмма потоков данных (DFD). Графический язык диаграммы. Примеры. Режим доступа: http://e-educ.ru/bd14.html
7. Волков Ю.О. " Диаграммы для описания бизнес-процессов". Режим доступа: http://yurivolkov.com/articles/Diagrams_for_business_processes_ru.html
8. Документация BusinessStudio. BPMN нотация. Режим доступа: http://www.businessstudio.ru/wiki/docs/v4/doku.php/ru/manual/creating_model/bpmn_notation

УДК 681.3
И.А. Спицын, магистрант
Научный руководитель: С.Г. Лелеков
канд. техн. наук, доц.
Севастопольский государственный университет

ОЦЕНКА НАДЕЖНОСТИ РАСПРЕДЕЛЕННЫХ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ С ПОМОЩЬЮ ИМИТАЦИОННОГО МОДЕЛИРОВАНИЯ

Аннотация: Данная статья посвящена рассмотрению основных методов оценки надежности распределенных вычислительных систем. Учтены все преимущества и недостатки каждого из методов. Рассмотрен процесс оценки надежности с помощью средств имитационного моделирования. Помимо существующих методов предлагается свой, уникальный. Делается вывод о наилучшем методе оценки надежности.
Ключевые слова: Вычислительная система, надежность, имитационное моделирование, оценка надежности, метод, вероятность.
Annotation:This article is devoted to main methods of assessing the reliability of distributed computing systems. Taking into account all the advantages and disadvantages of each method. The process of estimating the reliability using simulation tools. In addition to the existing methods offers a unique. The conclusion about the best method of assessing reliability.
Key words:Computing system, reliability, simulation, reliability evaluation, method, the probability

1. Введение

При проектировании сложных систем рассматриваются одними из основных параметров три важнейшие целевые размерности разработки – стоимость, производительность и надёжность.

Понятие надёжности является комплексной характеристикой. Оценка надёжности осуществляется множеством единичных и/или обобщённых количественных показателей, зависящим как от особенностей рассматриваемой системы, так и условий ее эксплуатации.

Надежность вычислительной системы представляет собой вероятность ее работы без отказов в течение определенного периода времени, рассчитанная с учетом весомости отказа для пользователя[1,2,3]. Из этого следует, что надежность системы определяется не только частотой отказов, но и последствиями, к которым они приводят.

В данной работе приводятся возможные варианты оценки надёжности методом имитационного моделирования, а также проводится их сравнение. Обосновывается выбор наиболее перспективных вариантов для реализации.

2. Оценка надежности вычислительной системы с помощью имитационного моделирования.

На этапе проектирования распределенных вычислительных систем можно применить имитационное моделирование. Оно необходимо для решения некоторых задач. Например:

- Оценка производительности системы на основе прогноза. Анализ структуры системы.
- Создание модели аппаратного инструмента и отладка программы на нем.
- Проверка алгоритмов и их свойств, используя одно и то же описание системы
- Используя заданные критерии оптимизировать структуру вычислительной системы

Создавая на этапе проектирования модель, которая будет учитывать структуру разрабатываемой вычислительной системы, ее аппаратную и программную составляющую, появляется мысль об использовании имитационной модели для оценки надежности вычислительной системы[4,5].

При моделировании системы исходные данные могут быть представлены в двух видах:

1. Для каждого компонента распределенной вычислительной системы присутствуют все необходимые характеристики надежности.
2. Характеристики надежности отсутствуют для тех или иных компонентов.

Для первого случая можно использовать уже существующие методы оценки надежности, такие как:

- метод минимальных путей и сечений. Этот метод применяется только для систем с монотонной структурой, свойства которых формулируются следующим образом:
  - система работоспособна, если все элементы работоспособны;
  - система находится в состоянии отказа, если все элементы отказали; при отказе элемента состояние системы только ухудшается.

Таким образом, для оценки надежности систем методом минимальных путей и минимальных сечений необходимо осуществлять следующее:

1. Построить все минимальные пути и минимальные сечения
2. Оценить вероятности работоспособного состояния системы.

- логико-вероятностный метод. В основе постановки и решения всех задач моделирования и расчета надежности систем с помощью ОЛВМ лежит так называемый событийно-логический подход.

Сущность заключается в использовании функций алгебры логики для аналитической записи условий работоспособности системы и переходе к вероятностным функциям,объективно выражающим безотказность системы.

- Метод псевдоэлементов. Идея метода: исходная надёжностная схема системы поэтапно преобразовывается таким образом, что схема постепенно укрупняется (в соответствии с какими-то алгоритмами). При этом параметры надёжности остаются неизменными. Укрупнение происходит до тех пор, пока она не будет укрупнена до первого элемента[6].

Однако второй случай, когда не хватает тех или характеристик надежности встречается гораздо чаще. Для решения этого вопроса будем использовать метод, так называемого внедрения отказов. Суть метода заключается в том, что мы сами внедряем отказы в вычислительную систему и наблюдаем ее способность или же не способности с этими отказами бороться.

Есть 2 способа внедрения отказов в систему во время моделирования:

1. Внедрять отказы в систему необходимо во время прогона модели. Недостаток этого метода состоит в том, что при смене типа отказа необходимо будет изменять саму суть имитационной среды, переделывать ее.
2. Внедрять отказы в систему необходимо во время анализа результатов. Недостаток этого метода состоит в том, что невозможно промоделировать работу системы во время функционирования отказоустойчивых алгоритмов[7,8].

Однако, в своей работе я буду использовать ситуацию, когда у каждого компонента системы известна одна или несколько характеристика надежности. В этом случае мы можем не только промоделировать систему, увидев процесс отказоустойчивости, но и рассчитать существующими методами надежность распределенной вычислительной системы.

В разрабатываемой среде, распределенная вычислительная система представляется в виде ориентированного графа. За веса данного графа принимаются показатели надежности: время безотказной работы, интенсивность отказа, коэффициент наработки до отказа, и т. д. Если представить вершины графа, как узлы сети, а ребра как линии связи, то анализируемую сеть можно рассматривать как систему массового обслуживания. Будем считать, что каждый элемент системы может находится в одном из двух состояний: работоспособном или отказа. Для обозначения состояния i-го элемента системы вводится булева переменная xi, такая, что:

$$x_i = \begin{cases} 1, \text{если } i - \text{й элемент работоспособен;} \\ \\ 0, \text{если } i - \text{й элемент отказал.} \end{cases}$$

Будем считать, что состояние системы полностью определяется состоянием ее элементов x = (x_1, x_2,…, x_n), где n – число элементов системы. В таком случае можно ввести булеву функцию φ состояния системы в целом:

$$\varphi(x) = \begin{cases} 1, \text{если система работоспособна;} \\ 0, \text{если система отказала.} \end{cases}$$

Функция $\varphi(x)$ называется структурной функцией системы.

Т.е. сначала выделяются ресурсы компьютера. Для всех них известны характеристики надежности. Затем запускается этап моделирования. В процессе моделирования собирается информация о тех или иных элементах, их состояние. Далее эта информация используется для расчета коэффициента надежности всей системы в целом.

В данном случае (табл. 1) $\mu_{nm}(t)$ представляет интенсивность потока атак n–ого дестабилизирующего фактора на m–й элемент информационной системы.

	E_1	E_2	…	E_m	…	E_M
Y_1	$\mu_{11}(t)$	$\mu_{12}(t)$	…	$\mu_{1m}(t)$	…	$\mu_{1M}(t)$
Y_2	$\mu_{21}(t)$	$\mu_{22}(t)$	…	$\mu_{2m}(t)$	…	$\mu_{2M}(t)$
…	…	…	…	…	…	…
Y_n	$\mu_{n1}(t)$	$\mu_{n2}(t)$	…	$\mu_{nm}(t)$	…	$\mu_{nM}(t)$
…	…	…	…	…	…	…
Y_N	$\mu_{N1}(t)$	$\mu_{N2}(t)$	…	$\mu_{Nm}(t)$	…	$\mu_{NM}(t)$

Табл.1. Таблица интенсивностей атак ДФ.

Данный метод более точен, чем существующие методы оценки надежности, т.к. является неким гибридом и вбирает в себя самое лучшее, что используется в каждом методе по отдельности.

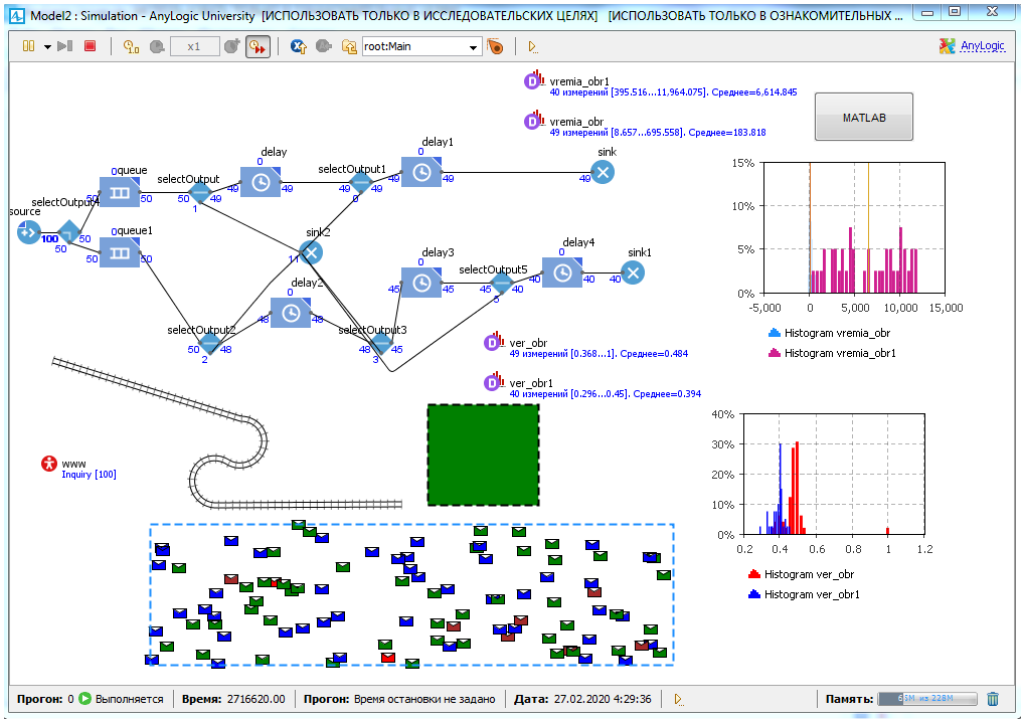


Рис.1. Исследование надежности РВС в Anylogic

3. Выводы и перспаективы исследований

В работе приведены и проанализированы основные методы оценки надежности распределенных вычислительных систем. Выделены преимущества и недостатки всех способов. Выбран и частично реализован гибридный метод оценки надежности. Проведена отладка и тестирование компьютерного модуля, которые с достаточно большой вероятностью показывают его работоспособность и адекватность решения поставленной задачи

Ближайшие перспективы развития исследований: 1) Программная реализация существующих методов, в первую очередь реализация гибридного подхода; 2) Сравнение результатов методов оценки надежности; 3) определение наилучшего метода по полученным результатам.

Библиографический список

1. ГОСТ Р ИСО/МЭК 9126 – 93. Информационная технология. Оценка программной продукции. Характеристики качества и руководства по их применению. // М.: Издательство стандартов, 1994.
2. Пугачев В.С. Методы введения избыточности для вычислительных систем / В.С. Пугачев. – М.: Советское радио, 2006. – 240 с.
3. Липаев В. В. / Программная инженерия. Методологические основы. // М.: ТЕИС, 2006.
4. Киселева М. В. К44 Имитационное моделирование систем в среде AnyLogic : учебно-методическое пособие / М. В. Киселёва. Екатеринбург : УГТУ - УПИ, 2009. 88 с.
5. Боев В. Д. Компьютерное моделирование: Пособие для практических занятий, курсового и дипломного проектирования в AnyLogic7:. — СПб.: ВАС, 2014. — 432 с.
6. Кузовлев В.И. Математические методы анализа производительности и надежности САПР / В.И. КузовлевП.Н. Шкатов. – М.: Высшая школа, 1990. – 144 с.
7. Чистолинов М.В., Бахмуров А.Г. “Среда моделирования многопроцессорных вычислительных систем”. Программные системы и инструменты №1(тематический сборник). Москва. МАКС Пресс. 2000.
8. Thorhuus R. “Software Fault Injection Testing”, Master of Science Thesis in Electronic System Design, Stockholm, February, 2000.

УДК 517.978
Е. Э. Фабиянский, магистрант.
Научный руководитель А. С. Миненко, доктор физ.-мат. наук, профессор
Донецкий национальный технический университет

ОСНОВНЫЕ ФУНКЦИИ И ЗАДАЧИ СИСТЕМЫ УЧЕТА РЕЕСТРА ПРОДАЖ АВТОТРАНСПОРТА В АВТОСАЛОНЕ

Аннотация: Рассматриваются существующие системы автоматизации продаж, проводится сравнительный анализ их основных характеристик. Проводится системный анализ автосалона «АИС Донецк». Выделяются основные требования, задачи и характеристики информационной системы реестра продаж автотранспорта в автосалоне «АИС Донецк».
Ключевые слова: системный анализ, информационная система, бизнес-процессы.
Annotation: We consider the existing sales force automation system, a comparative analysis of their main characteristics. Conduct a systems analysis of the car showroom "AIS Donetsk.". Stand out the basic requirements, objectives and characteristics of vehicles sales registry information system in the car showroom "AIS Donetsk."
Key words: systems analysis, business process, information system.

В настоящее время процесс автоматизации затронул не только производственную, техническую и технологическую сферы деятельности человечества, но и информационное пространство. Системы автоматизации помогают сократить избыточность хранимых данных, также влияют на уменьшение затрат на многократные операции обновления избыточных копий и устранение возможности возникновения противоречий из-за хранения в разных местах сведений об одном и том же объекте, помогают ускорить оперативность получения необходимой информации, что значительно улучшает качество обслуживания клиентов.

Целью данной работы является усовершенствование информационной системы автосалона «АИС Донецк» при помощи рассмотрения уже существующих систем, учета их недостатков и достоинств, а также проведения системного анализа автосалона «АИС Донецк», для последующего выделения основных функций и бизнес-процессов.

Практическое значение полученных результатов заключается в увеличении объема продаж, получении оперативной информации, интересующей покупателя, тем самым, повышая качество обслуживания, что повлечет за собой привлечение новых клиентов в автосалон.

Необходимо рассмотреть существующие системы автоматизации и провести их сравнительный анализ, для выделения их основных преимуществ и недостатков, что поможет улучшить информационную систему автосалона «АИС Донецк».

1. Программный продукт «Альфа-Авто: Автосалон + Автосервис + Автозапчасти» предназначен для комплексной автоматизации учета на предприятиях автобизнеса (магазинов оптовой и розничной продажи запчастей, автосервисов и станций технического обслуживания автомобилей, автосалонов и дилерских центров).

2. Программа АвтоСалон предназначена для автоматизации деятельности автосалонов, торгующих автомобилями, в части, касающейся ведения учета, продаж автомобилей и формирования договорных документов. Также предусмотрено формирование и печать документов, специфичных для данного вида деятельности (справка–счет, ПТС, транзитный номер).

3. Система «1С: Предприятие 8.2» имеет в своей основе ряд механизмов, определяющих концепцию создания прикладных решений.

В качестве ключевых моментов можно выделить изоляцию разработчика от технологических подробностей, алгоритмическое программирование только бизнес–логики приложения, использование собственной модели базы данных и масштабируемость прикладных решений без их доработки.

Состав прикладных механизмов «1С: Предприятия» ориентирован на решение задач автоматизации учета и управления предприятием. Использование проблемно-ориентированных объектов позволяет разработчику решать самый широкий круг задач

складского, бухгалтерского, управленческого учета, расчета зарплаты, анализа данных и управления на уровне бизнес-процессов.

Потребность в удобных и эффективных средствах разработки программного обеспечения, а также быстрое развитие вычислительной техники, привели к появлению систем программирования, ориентированных на так называемую «быструю разработку», среди которых можно выделить Borland Delphi и Microsoft Visual Basic.

4. Delphi 7 представляет собой современную и мощную систему программирования и имеет определенные особенности.

У существующих аналогов были выявлены следующие недостатки:

- нет гибких механизмов разграничения доступа к данным;
- недостаточная функциональность;
- нет многопользовательского доступа;
- высокая цена.

Выше указанные программы «1С: Предприятие 8.2», а так же «Borland Delphi» могут быть использованы для улучшения работы автосалона «АИС Донецк», но было принято решение разработать автоматизированную систему учета реестра продаж автотранспорта в автосалоне, с учетом конкретных проблем и нюансов работы «АИС Донецк».

Для того, чтобы обозначить бизнес-процессы, которые необходимы при разработке информационной модели, необходимо разобраться в организации, составе и функциях рассматриваемого автосалона. Для этого проведем системный анализ «АИС Донецк».

Для описания модели «черный ящик» необходимо составить модели входов, выходов. Входами в системе «Автосалон «АИС Донецк»» являются:

- продукция с завода;
- клиент на обслуживание;
- каталоги продукции завода;
- финансовые средства.

В системе «Автосалон «АИС Донецк»» можно выделить такие выходы:

- каталоги продукции автосалона;
- накладные;
- обслуженный клиент;
- реализованная продукция.

Модель «черный ящик» приведена на рисунке 1.

Рис. 1. Модель «Черный ящик»

Для разработки модели состава проведем субстратный анализ, определив из каких основных элементов состоит автосалон «АИС Донецк». Автосалон «АИС Донецк» можно разделить на: склад, руководство автосалона, отдел продаж и бухгалтерию. Склад состоит из отдела тестирования, отдела приема продукции и отдела выдачи продукции. Руководство автосалона «АИС Донецк» это – генеральный директор автосалона и директор. Отдел продаж состоит из главного менеджера, ведущего менеджера и менеджера. А бухгалтерия состоит из главного бухгалтера и бухгалтера.

На рисунке 2 приведена модель состава системы.

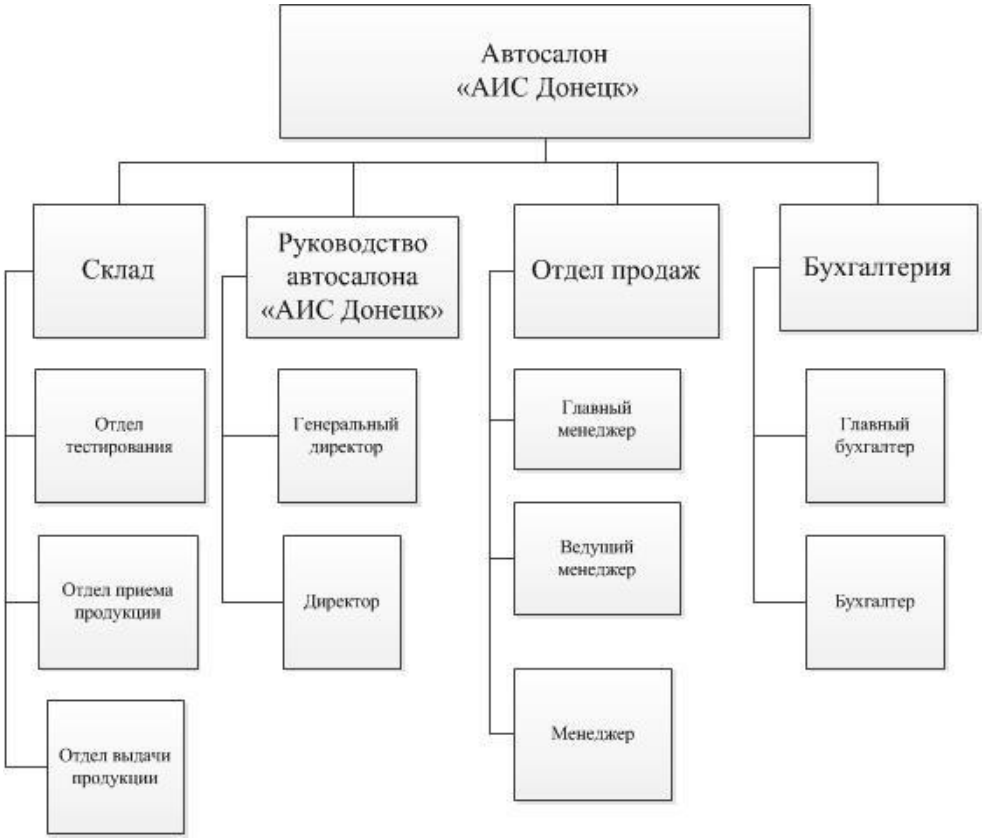


Рис. 2. Модель состава системы

Осуществим функциональное описание системы, которое включает в себя раскрытие ее внешнего и внутреннего функционирования. Для разработки функциональной модели системы нам необходимо составить контекстную диаграмму, которая представлена на рисунке 3.

Рис. 3. Контекстная диаграмма А-0

Декомпозированная контекстная диаграмма, которая необходима для более полного рассмотрения способности выполнении системой своей внешней функции, представлена на рисунке 4.

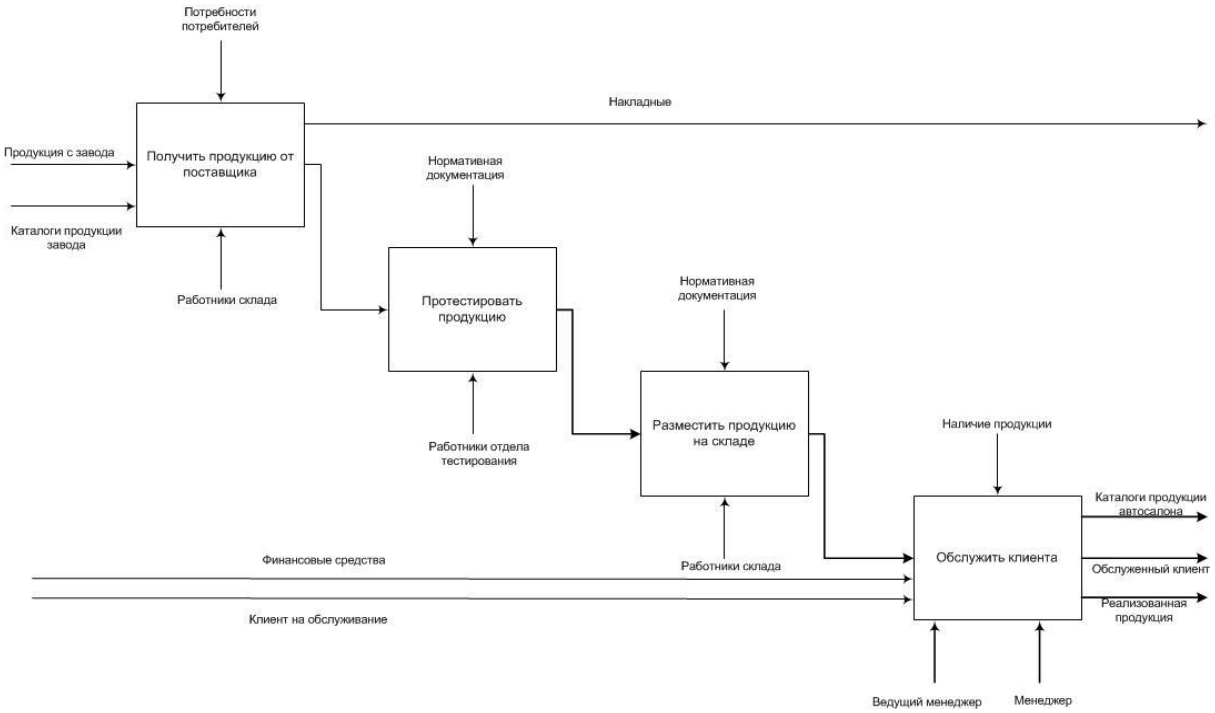


Рис. 4. Декомпозированная диаграмма А0

На основе проведенных исследований необходимо выделить основные характеристики и задачи, а также бизнес-процессы, необходимые для создания информационной системы учета и реестра продаж автотранспорта «АИС Донецк».

Рассмотрим бизнес – процессы, необходимые для учета реализации автомобилей в автосалоне:

1) заказ автомобиля – после выбора автомобиля оформляется заказ на выбранную модель, подготавливается и отправляется запрос на завод - изготовитель, принимается предоплата и выдается квитанция о предоплате;

2) прием автомобиля – принятие автомобиля на внутренний учет, проведение предпродажной подготовки и диагностики автомобиля, оповещение покупателя;

3) реализация автомобиля – осмотр автомобиля покупателем, оформление договора купли-продажи;

4) регистрация оплаты;

5) формирование отчетных документов:

- формирование отчета «Прайс-лист»;
- формирование отчета «Анализ продаж»;
- формирование отчета «Заказы автомобилей»;
- формирование отчета «Состояние заказов».

АИС «Учет автомобилей в автосалоне» должна обеспечивать выполнение следующих функций:

1) ввод справочных данных;

2) ввод сведений о поставщиках;

3) ввод сведений о покупателях;

4) ввод сведений об автомобилях;

5) ввод сведений о заказах;
6) внесение изменений сведений о автомобилях;
7) внесение изменений сведений о поставщиках;
8) формирование различных отчетов: прайс-лист автомобилей, анализ продаж, реестр справок счетов, заказы автомобилей, состояние заказов автомобилей;
9) внесение сведений о проданных автомобилях;
10) выписка счета на оплату.

В работе были рассмотрены существующие системы автоматизации, проведен их сравнительный анализ, были выделены основные достоинства и недостатки автоматизированных систем. Также был проведен системный анализ автосалона «АИС Донецк», обозначены основные задачи, а также приведено описание бизнес процессов и этапов работы. Данные исследования, в дальнейшем, помогут разработать модель автоматизированной информационной системы, которая сможет повысить эффективность работы автосалона, поможет улучшить качество обслуживания и, как следствие, увеличить прибыль «АИС Донецк».

Библиографический список

1. Спицнадель В. Н. Основы системного анализа: Учеб. пособие. — СПб.: «Изд. дом «Бизнесс-пресса», 2000 г. — 326 с.
2. Лямец Владимир Ильич. Системный анализ. Вводный курс : Учебное пособие / Владимир Ильич Лямец, Андрей Дмитриевич Тевяшев . – 2-е изд., перераб. и доп. – Харків : Издательство ХНУРЭ, 2004 . – 448 с.
3. Злупко С. Принципи дослідження міждисциплінарних проблем : Згуровский М., Панкратова Н. Системный анализ: Проблемы, методология, приложения. — К.: Наук. думка, 2005. — 743 с. / С. Злупко // Вісник Національної академії наук України. - 2007. - № 2. - С. 86-88.
4. Саати Т., Кернс Л. Аналитическое планирование. Организация систем. - М.: Радио и связь, 1991. - 224 с.

УДК 004.415
В.В. Чижевский, студент 2-го курса магистратуры
Научный руководитель: В.И. Шевченко, к.т.н., доцент
Севастопольский государственный университет

ИССЛЕДОВАНИЕ ПРОЦЕССОВ ПОДДЕРЖКИ ИТ-СЕРВИСОВ МЕТОДАМИ КЛАСТЕРНОГО АНАЛИЗА

Аннотация: разработана концептуальная модель классификации систем автоматизации бизнес-процессов по степени критичности ИТ-сервисов. Приведено описание модели в рамках нотации языка UML. Предложена модель программной системы «Кластеризация ERP», смоделированная при помощи программного обеспечения AnyLogic 7.2. Представлено исследование характеристик модели системы «Кластеризация ERP» при произвольных потоках данных о наступлении событий трёх видов: инцидент, проблема, запрос на обновление. Проведён сравнительный анализ коэффициента эффективности поддержки SLA в зависимости от законов распределения потоков данных.
Ключевые слова: кластерный анализ, бизнес-критическая система, качество ИТ-сервиса, закон распределения.
Annotation: developed a conceptual model of the classification systems of the automation of business processes on the degree of criticality of IT services. The description of the model within the UML notation. The model of "Clustering ERP" software system, modeled using AnyLogic 7.2 software. Presented study characteristics "Clustering ERP" system model for pro-freestyle flows of the occurrence of the events of three types of data: incident, problem, request an upgrade. The comparative analysis of the efficiency coefficient SLA support, depending on the data flow distribution laws.
Key words: cluster analysis, business-critical system, the quality of IT services

1. Введение

На сегодняшний день информационные технологии (ИТ) оказывают существенное влияние на бизнес-процессы. Появление персональных компьютеров, локальных сетей, технологии клиент-сервер и Интернета позволили организациям быстрее выводить на рынок свои продукты и услуги. Это привело к тому, что в рамках традиционных иерархических организаций часто бывает трудно реагировать на условия быстроменяющихся рынков и возникает необходимость к переходу на более гибкий, адаптивный режим управления бизнес-компаний с меньшей степенью иерархичности и большей децентрализацией. Организация эффективной работы бизнеса напрямую зависит от поддерживающих его ИТ-сервисов[1].

С целью эффективной поддержки ИТ-сервисов ИТ-отделами предприятий устанавливаются специальные системы мониторинга, позволяющие собирать информацию о качестве работы информационных систем, обслуживающих бизнес. При этом возникает новая проблема, объем этих данных настолько велик, что обработать их без применения информационных технологий не возможно.

Для решения этих проблем существуют методы классификации и кластеризации, применяемые в процессе анализа «больших данных». Особенностью данных, к которым применяются эти методы – многомерность, т.е. несводимость к одному единственному аспекту или характеристике, невозможность прямого измерения их существенных характеристик, таких, как «эффективность производства» или «выработка установленного рабочего времени»[2].

В данной работе рассматривается задача кластеризации «больших данных». Результатом обработки исходных данных с помощью проектируемой программной системы является структурированная совокупность данных о составе кластеров программного обеспечения (ПО) по степени критичности [3,4].

2. Описание программной системы

Программная система состоит из двух основных модулей – модуль интеграции лог-журналов и модуль обработки данных, реализованных на языке 1С платформы 1С:Предприятие 8.2. Данная система позволяет пользователю получать структурированные данные о критичности исследуемых объектов на основе лог-журналов.

Компьютерная система работает по следующему принципу: на начальном этапе пользователю необходимо при помощи модуля интеграции загрузить лог-журнал в разработанную систему, это происходит при помощи преобразования лог-журнала в формат xml; затем, после конвертации в xml, происходит загрузка данных, для последующей

кластеризации в модуль обработки. Далее пользователю необходимо ввести параметры кластеров и послать запрос системе на обработку данных. Структурная схема взаимодействия между модулями изображена на рис. 1.

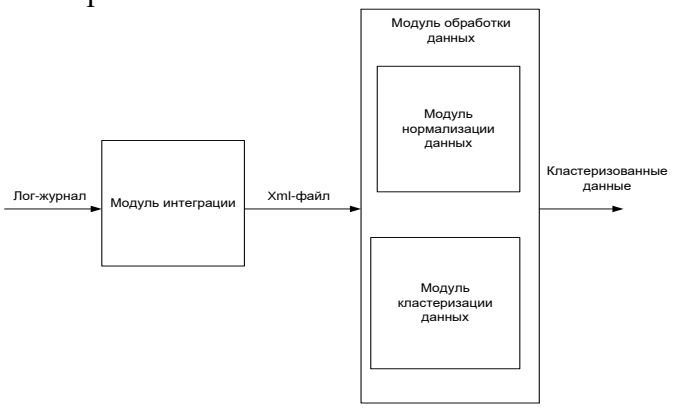


Рис.1. Структура взаимодействия модулей программной системы

Концептуальная схема компьютерной системы представлена в виде диаграммы классов. На таких диаграммах, как правило, показывается множество классов, интерфейсов, кооперaций и отношений между ними. Диаграммы классов используются для моделирования статического вида системы с точки зрения проектирования. Диаграмма концептуальных классов разработанной программной системы приведена на рис. 2.

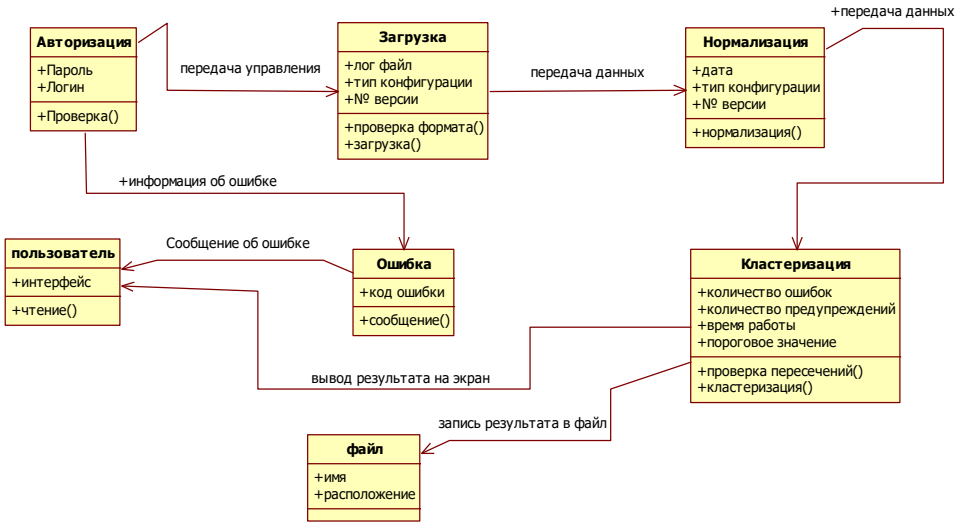


Рис.2. Диаграмма концептуальных классов программной системы

3. Исследование характеристик модели системы «Кластеризация ERP»

Проведено исследование характеристик эффективности работы системы динамической кластеризации «Кластеризация ERP». С этой целью была разработана имитационная модель компьютерной системы, исследующая процессы классификации при произвольных потоках данных о наступлении событий трёх видов: инцидент, проблема, запрос на обновление. В качестве законов распределения для каждого потока рассматривается: равномерное распределение; нормальное распределение; гамма распределение.

Последовательность обработки заявок в системе: заявки поступают от источника «Лог в очередь», «Регистр загрузки», после чего, поступают в устройство «Загрузка», далее в очередь «Регистр нормализации» и после на устройство «Нормализация». Далее при помощи вероятностного условия организован процесс имитации поступления инцидентов в систему, при этом заявка из условного блока переходит либо в блок уничтожения транзактов (время обработки инцидента превышает допустимые пределы), либо к блокам, имитирующим обработку одним из трёх методов кластеризации.

Особенности настройки блоков моделирования при равномерном распределении входного потока заявок:

1) Блок Sourse: Лог – заявки прибывают согласно равномерному закону распределения. Интенсивность: uniform(10,15) в секунду. Агенты, которые не могут выйти: уничтожаются.

2) Блоки Delay. Имя: «загрузка», «нормализация», «полныхсвязей», «ксредних», «средних», «евклидовоп», «манхэттенскоеп», «степенноеп», «евклидовок», «манхэттенскоек», «степенноек», «евклидовос», «манхэттенскоес», «степенноес». Тип: определённое время. Время задержки: uniform(40,50), uniform(40,45), uniform(40,100), uniform(40,130), uniform(40,150), uniform(20,40), uniform(40,50), uniform(40,50), uniform(20,40), uniform(40,50), uniform(40,50), uniform(20,40), uniform(40,50), uniform(40,50) соответственно порядку представленных имён.

3)Боки SelectOutput: выход truevыбирается согласно закону распределения вероятностей – uniform(0, 0.2). Использует функцию *statsSize.mean()* для определения средней длины очереди, а также функцию *statsUtilization.mean()* для определения коэффициента использования устройства.

Результат моделирования системы представлен на рис.3.

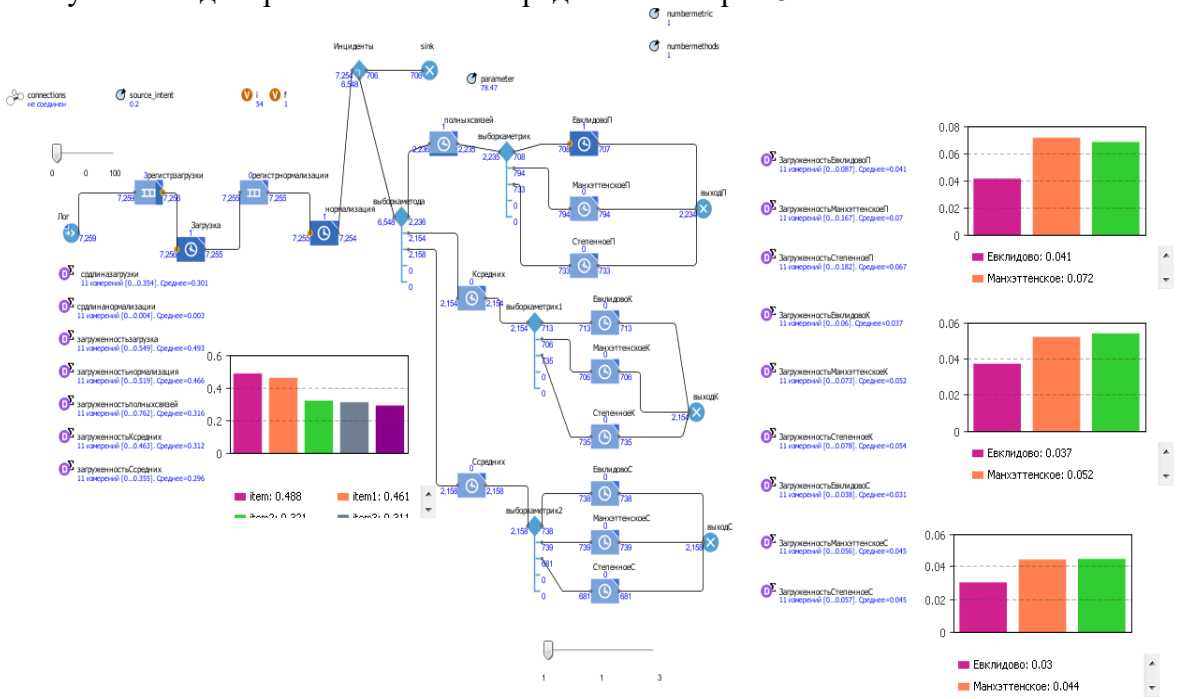


Рис.3. Результат моделирования системы при равномерном законе распределения

Аналогичным образом имитационная модель может быть перенастроена для других видов распределения. Результаты моделирования по трём законам распределения: равномерному, гамма, нормальному сведены в таблицы 1−3 соответственно.

Таблица 1. Статистика моделирования с использованием равномерного закона распределения

Элемент	Коэф. использования	Вход	Выход	Ср. длина очереди
Лог	-	-	7259	-
Инциденты	-	7254	6548	-
Полных связей	0.316	2236	2235	-
К-средних	0.312	2154	2154	
С-средних	0.296	2158	2158	
Регистр загрузки	-	7259	7256	0.301
Регистр нормализации	-	7255	7255	0.003

Таблица 2. Статистика моделирования с использованием гамма распределения

Элемент	Коэф. использования	Вход	Выход	Ср. длина очереди
Лог	-	-	8477	-
Инциденты	-	8474	6760	-
Полных связей	0.337	2418	2418	-
К-средних	0.311	2216	2213	
С-средних	0.3	2126	2126	
Регистр загрузки	-	8477	8477	0.365
Регистр нормализации	-	8477	8475	0.588

Таблица 3 Статистика моделирования с использованием нормального закона распределения

Элемент	Коэф. использования	Вход	Выход	Ср. длина очереди
Лог	-	-	7127	-
Инциденты	-	7126	5646	-
Полных связей	0.119	1885	1885	-
К-средних	0.161	1898	1898	
Ссредних	0.137	1863	1862	
Регистр загрузки	-	7127	7127	0.041
Регистр нормализации	-	7126	7126	0.029

Был рассчитан коэффициент эффективности уровня поддержки SLA для каждого закона распределения как $K_{эф} = N_{вых} / N_{вх}$, где $K_{эф}$ – коэффициент эффективности, $N_{вх}$ – количество заявок на входе системы, $N_{вых}$ – количество заявок на выходе системы.

Для равномерного закона распределения $K_{эф} = 0.901$. Для гамма распределения $K_{эф} = 0.797$. Для нормального закона распределения $K_{эф} = 0.792$.

Следует отметить, что в случае если входной поток запросов по обслуживанию пользователей будет иметь равномерный закон распределения, то при прочих равных параметрах обработки запросов, коэффициент эффективности уровня поддержки принимает максимальное значение.

В рамках дальнейших исследований предполагается разработать сценарий информационной технологии поддержки ИТ-сервисов в виде диаграмм UML, а также определить комплексный критерий эффективности поддержки ИТ-сервисов. С учётом выбранных критериев эффективности для уже реализованной модели в среде AnyLogic планируется провести серию экспериментальных исследований с целью определения поля эффективных решений при различных коллизионных ситуациях.

Библиографический список

1. Радченко М. Г. 1С:Предприятие 8.2. Практическое пособие разработчика. 2009. − 874 с.
2. Мандель И. Д. Кластерный анализ [Текст]/ И. Д. Мандель. — М.: Финансы и статистика, 1988 – 176 с.
3. Луговская Л.П. Динамическая кластеризация информационных потоков [Текст]/Л.П. Луговская, А.В. Скатков, В.И. Шевченко // Вестник СевНТУ. Информатика, электроника, связь: сб. науч. тр. Севастопольского нац. техн. ун-та. – Севастополь, 2011.–№114.– с. 14 – 20.
4. Мащенко Е.Н. Исследование критических ситуаций в ИТ-инфраструктурах методами кластерного анализа [Текст] / Е.Н. Мащенко, В.И. Шевченко // Радиоэлектронные и компьютерные системы. – 2012. – No. 5(57). – с. 191 – 196.

СЕКЦИЯ 2

Робототехника. Технологии искусственного интеллекта

УДК 004.032.26
С.Е. Дядюшенко, студент 4-го курса
Научный руководитель: В.А. Строганов, ст. преподаватель.
Севастопольский государственный университет

ПРОЕКТИРОВАНИЕ ОБЪЕКТНО-ОРИЕНТИРОВАННОЙ СТРУКТУРЫ ВЕЙВЛЕТ-НЕЙРОННОЙ СЕТИ.

Аннотация: Проведён анализ функционирования вейвлет-нейронной сети. Предложена объектно-ориентированная структура классов, реализующих построение и обучение сети.
Ключевые слова: Вейвлет-нейронная сеть, вейвлеты, искусственные нейронные сети, проектирование.
Annotation: Performed analysis of wavelet network functioning. Proposed object-oriented structure of classes, that implements building and learning of wavelet networks.
Keywords: Wavelet network, wavelets, artificial neural networks, design.

В последнее время для решения разнообразных типов задач широко применяются искусственные нейронные сети.

Нейронная сеть – это совокупность соединённых абстрактных объектов, называемых нейронами. Каждый нейрон выполняет обработку сигналов, поступающих на его вход, и рассматривается как некоторый процессорный элемент [1]. Название «нейронные сети» появилось из-за того, что алгоритм их функционирования подобен алгоритмам преобразования информации в клетках-нейронах головного мозга.

В вейвлет-нейронных сетях функцией активации нейрона является вейвлет-функция, используемая для вейвлет-преобразования. Вейвлет-преобразование является аналогом преобразования Фурье. В отличие от преобразования Фурье, которое позволяет обнаружить только наличие или отсутствие той или иной гармоники, вейвлет преобразование за счет временной локализации базисных функций позволяет выполнять анализ локальных особенностей сигнала. Вейвлет-нейронная сеть является дальнейшим развитием вейвлет-анализа и объединяет в себе свойства вейвлет-преобразования и искусственных нейронных сетей [2]. ВНС представляет собой трехслойный персептрон и может быть использована в качестве универсального аппроксиматора сигналов. Достоинством таких сетей является возможность динамического обучения, то есть функция, описывающая сигнал, изменяется на лету, после момента поступления сигнала в сеть. При получении новых данных о сигнале корректируются веса и коэффициенты сети.

Вейвлет-нейронные сети широко используются, в частности, для очищения сигналов от шумов, распознавания сигналов, решения задач прогнозирования.

Однако, не смотря на широкое распространение, не существует открытых для использования программных продуктов, специализирующихся на вейвлет-нейронных преобразованиях. Целью работы является устранение этого пробела и разработка объектно-ориентированной структуры вейвлет-нейронной сети для преобразования сигнала.

Очевидно, что разработанная структура должна быть в достаточной степени гибкой и расширяемой, с возможностью добавления вариантов вейвлет-функций и методов обучения.

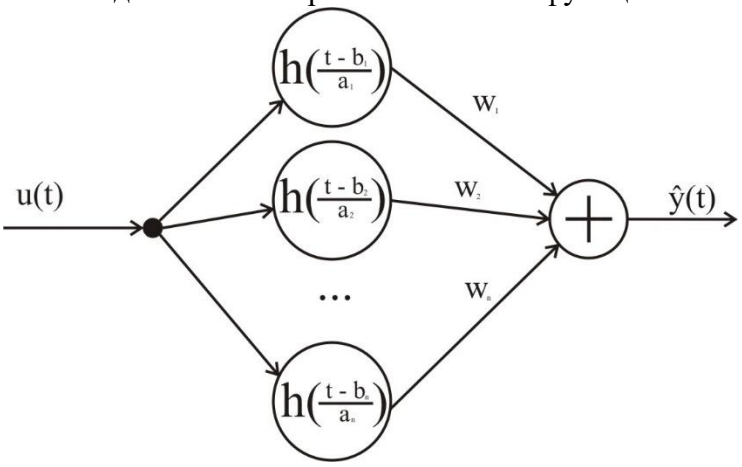


Рис.1. Вейвлет-нейронная сеть

Рассмотрим структуру вейвлет-нейронной сети. На вход поступает сигнал *u(t)*. Его возможное применение – задание множителя выхода. Нейронам входного слоя необходимо явно задавать вход. Остальные нейроны используют выход нейронов предыдущего слоя. Каждый нейрон скрытого слоя содержит некоторую функцию $h(\tau)$. Параметр τ зависит от коэффициентов масштабирования a_i и сдвига b_i, индивидуальных для каждого нейрона. Нейрон выходного слоя выполняет стандартное суммирование. По своей сути он не отличается от нейрона обычной нейронной сети. Выходом сети является функция $\hat{y}(t)$ воспроизводящая с определённой точностью описываемый сигнал:

$$\hat{y}(t) = u(t)\sum_i w_i h(\frac{t-b_i}{a_i}). \quad (1)$$

Параметры нейронной сети w_k, a_k и b_k могут быть оптимизированы путём минимизации функции квадратичной ошибки E за время t [3]. Функция стоимости определяется как

$$E = \frac{1}{2}\sum_{i=1}^{T} e^2(t), \quad (2)$$

где $e(t)$ – ошибка обучения в момент времени t

$$e(t) = y(t) - \hat{y}(t). \quad (3)$$

Реализация функции активации нейрона отделена от самого нейрона. Это позволяет назначать одинаковым нейронам разные функции и в то же время использовать эти функции отдельно от нейронной сети.

В результате анализа структуры вейвлет-нейронной сети была разработана иерархия классов, реализующих различные типы нейронов. Диаграмма классов представлена на рисунке 2.

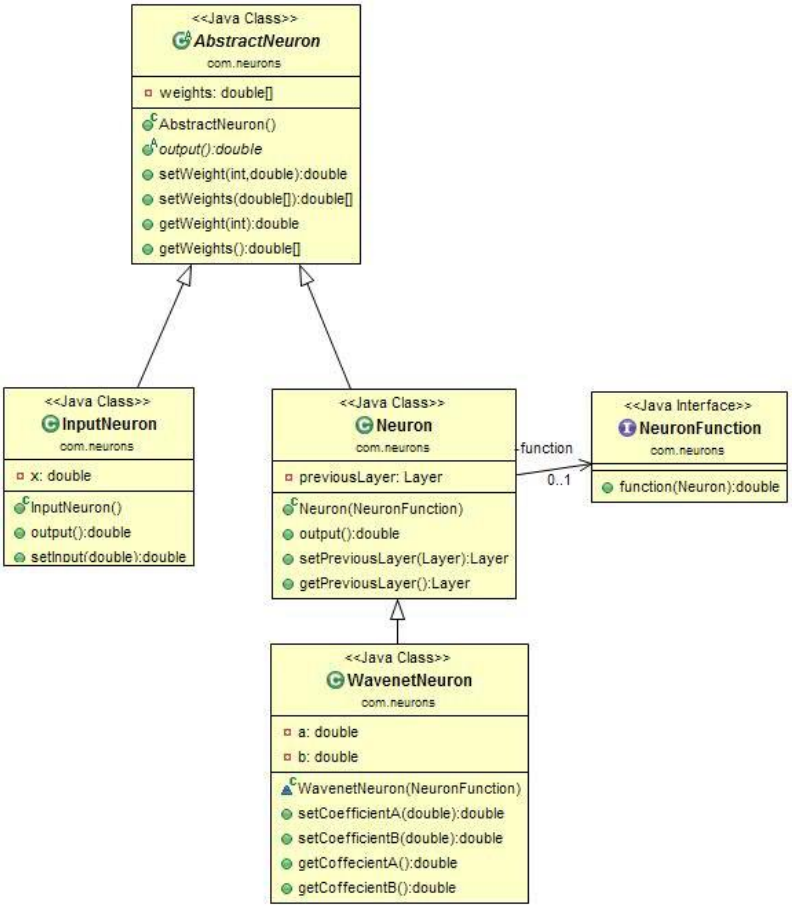


Рис.2. Диаграмма классов для описания нейронов

Выделенные функции должны иметь метод возвращения своего результата с одной и той же сигнатурой. Этому методу необходима информация вызывающего нейрона (выхода предыдущего слоя, весов связей, коэффициентов вейвлет-нейронной сети). Таким образом, для множества нейронных функций создан общий метод возвращения результата, перегруженный в различных реализациях, принимающий параметром нейрон. Этот метод был выделен в отдельный интерфейс, ссылка на экземпляр которого помещена в нейрон. Конкретными реализациями данного интерфейса могут быть различные нейронные функции, как вейвлеты, так и любые другие.

После построения структуры сети необходимо инициализировать веса связей между нейронами, а также коэффициенты. Эти обязанности возложены на отдельные классы. Инициализация весов и/или коэффициентов будет проводиться отдельно для каждого нейрона в зависимости от его типа. Инициализаторы связаны с необходимыми нейронами посредством класса слоя, который также используется для инициализации нейронных функций и проверки правильности построения модели.

Далее необходимо проверить на правильность структуру и обучить построенную модель. Для этого введён класс модели, объединяющий в себе все слои сети. Модель имеет доступ ко всем слоям и все нейронам, поэтому может проверить правильность их взаимодействия и модифицировать веса и коэффициенты во время обучения.

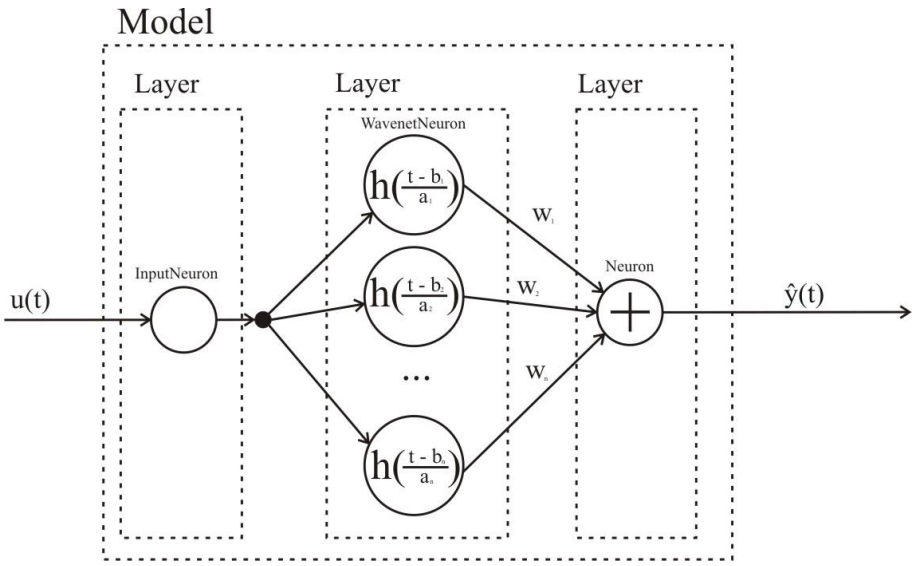


Рис.3. Обозначение конкретных классов на вейвлет-нейронной сети

Время обучения едино для всей системы. Получать его значение необходимо в совершенно разных её частях. Поэтому класс, отвечающий за хранение и получение времени в системе, объявлен статическим.

Существуют разные способы обучения нейронной сети, по разному воздействующие на нейроны. Во время обучения также могут использоваться различные функции пересчёта весов и коэффициентов. Для сохранения возможности быстрого создания новых классов-методов посредством введения абстрактного класса реализован единый интерфейс обучения.

Аналогично ситуации с нейронными функциями, целесообразно выделить функции пересчёта весов и коэффициентов. Для получения текущего значения сигнала вводится класс-обёртка для потока/массива значений, при помощи которых будет происходить обучение вейвлет-нейронной сети. Метод обучения получает выход модели и текущее значение из класса значений, вычисляет разницу и производит соответствующие изменения в сети.

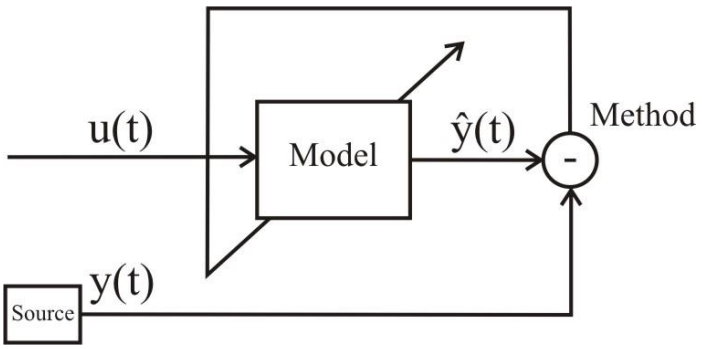


Рис.4. Обозначение конкретных классов на схеме обучения

В результате была спроектирована структура классов, реализующих построение и обучение нейронной сети. Диаграмма классов представлена на рисунке 5.

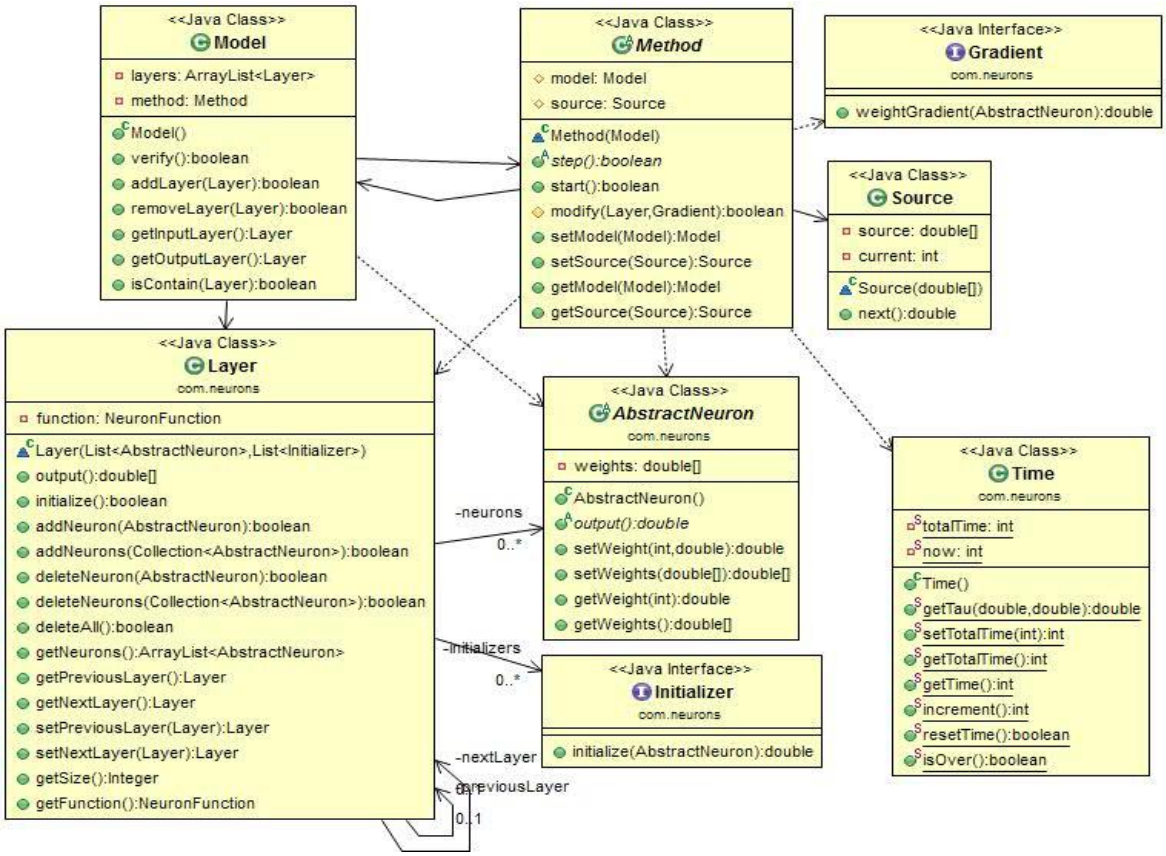


Рис.5. Диаграмма взаимосвязанных классов построения и обучения нейронной сети

Таким образом в данной работе решена проблема создания гибкой объектно-ориентированной структуры вейвлет-нейронной сети. На основании анализа структуры вейвлет-нейронной сети создана система взаимосвязанных между собой классов, реализующих построение и обучение сети.

Библиографический список

1. Бондарев В.Н. Искусственный интеллект: Учеб. Пособие для вузов/ В.Н. Бондарев, Ф.Г. Аде – Севастополь: Изд-во СевНТУ, 2002 – 615с.
2. Строганов, В. А. Использование центральных частот вейвлетов при инициализации вейвлет-нейронных сетей [Текст] : сб. науч. тр. / В. А. Строганов // Вестн.СевНТУ. Сер. Информатика, электроника, связь. – 2010. – Вып. 101. – С. 33–36.
3. Lekutai G. Adaptive Self-Tuning Neuro Wavelet Network Controllers / G. Lekutai.-Blacksburg:Virginia Polytechnic Institute and State University, 1997.-122 p.

УДК 681.521.2
Т.Г. Москалюк, студент 3 курса, кафедры КСМ
Научный руководитель: А.Ю.Харитонов, ст.пр каф КСМ
Донецкий национальный технический университет

ПРИМЕНЕНИЕ РОБОТИЗИРОВАННОЙ СИСТЕМЫ НА ОСНОВЕ "LEGO NXT" ДЛЯ АВТОМАТИЗАЦИИ СОСТАВЛЕНИЯ ЭЛЕКТРОННОГО КАТАЛОГА ГОРНО - ГЕОЛОГИЧЕСКОГО МУЗЕЯ ДонНТУ

Аннотация: В этой статье будет рассмотрено применение роботизированной системы на основе контроллера "Lego NXT" для автоматизации и упрощения составления трёхмерного электронного каталога минералов и окаменелостей горно-геологического музея Донецкого национального технического университета. В статье описана конструкция и алгоритм работы системы, поднимаются вопросы соотношения качества и объёма отснятого материала, рассмотрена проблематика съёмки и редактирования больших объёмов исходного материала, также, метод решения этой проблемы. Актуальность статьи связана с упрощением трудоёмкого процесса съёмки изображений в высоком разрешении, а также повышением качества и точности решения задачи в условиях больших объёмов работы.
Ключевые слова: Автоматизация, робот, съёмка, музей, объёмные снимки, контроллер.
Annotation: The purpose of this article is to review the creation, construction, programming and usage of robotic system Multiframe Automatic Universal Device", based on " Lego Mindstorms NXT " platform and "Lego NXT" controller , for automation and simplification of creation of digital catalogue of exhibits of mineralogical museum of Donetsk national technical university. This article provides information about the construction, principles and algorithms of system's functioning, the concepts of programming in "Lego MINDSTORMS NXT" development environment, based on "National Instruments LabVIEW". Also, the questions of ratio between quality and volume of captured material, problematics of capturing and processing of large volumes of input materials, and methods of solution of those problems, lossless on the resulting material, are being discussed. The perspectives of further development in the field of creation and usage of three-dimensional images of real objects, in medicine including plastic surgery, orthopedics, traumatology and prosthetics, science and the entertainment field are being discussed too. The relevance of an article is connected to simplification of time consuming process of capturing three-dimensional images, and improvement of the quality and preciseness of solution of the problem in situation of big volumes of work.
Keywords: Automation,robot,capturing,museum,3-Dimensional capture, controller.

Введение

Горно-геологический музей Донецкого национального университета является уникальным проектом, основанным в 1924 году. В текущий момент музей насчитывает более десяти тысяч экспонатов. В его коллекции собраны образцы полезных ископаемых и окаменелостей региона Донбасса, а также Германии, Болгарии, Чехословакии, Канады, Китая, Советского союза, и стран СНГ. Музей университета проводит экскурсии для школьников, и является лучшей предметной аудиторией по обучению студентов: образцы материалов применяются для проведения лекционных и лабораторных занятий. К юбилею вуза - 95-летию со дня основания, было принято решение о создании виртуального каталога музея [1]. Каталог станет энциклопедическим справочником, упрощающим работу с музеем. Для создания каталога было принято решение о внедрении технологии объёмного фотографирования экспонатов. Фотографирование осуществлялось вручную, когда объект съёмки поворачивался на заданный угол, и фотографировался.

Постановка задачи

Следует обратить внимание на факт, что такой подход к созданию объёмных моделей серьёзно снижает качество, резкость, и глубину фокуса фотографий, усложняет обработку изображений, и, соответственно, точность результирующих моделей, а, также, является излишне трудоёмким и требующим большого ресурса времени. В связи с вышеописанными трудностями, была поставлена задача автоматизации и оптимизации съёмки объёмных моделей, тем самым повысив общее качество электронного каталога.

Результаты

Исходя из поставленной задачи, мы имеем проблему улучшения качества электронного каталога, которая будет решена с использованием автоматизированной системы. Для реализации решения проблемы была выбрана учебная система "Lego NXT". В составе системы содержится контроллер, оснащённый набором различных датчиков и прецизионных

сервоприводов с обратной связью, а, также, набор конструкционных деталей, включая различные зубчатые колёса, оси и валы. На основе выбранной системы была сконструирована роботизированная система Multiframe Automatic Universal Device (далее - MAUD). Конструктивно, MAUD представляет собой платформу, на которой располагается объект съёмок, оснащённую приводами для точного его вращения, и универсальный обвес для устройства съёмки, оснащённый приводом управления спускового механизма устройства и датчиком освещённости. MAUD производит вращение платформы на заданный градус, происходит спуск устройства съёмки, проверка того, был - ли отснят кадр, и продолжение работы аппарата до полного оборота экспоната. На рисунке 1 представлено схематическое строение устройства. Рисунки 2 и 3 — фотографии реального прототипа MAUD.

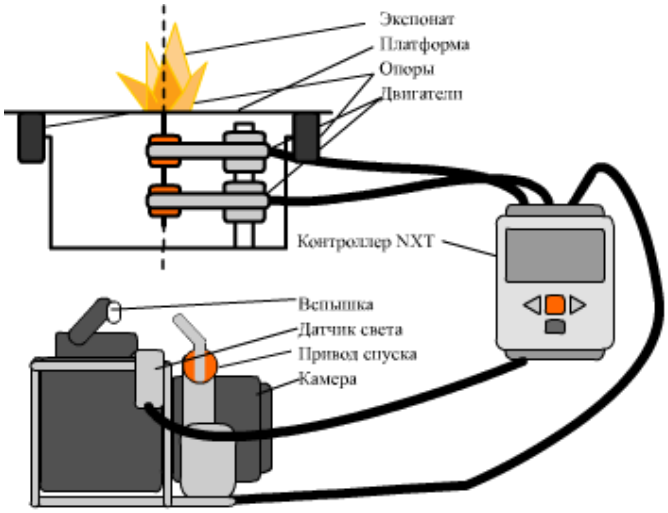


Рис. 1. Схематическое строение MAUD

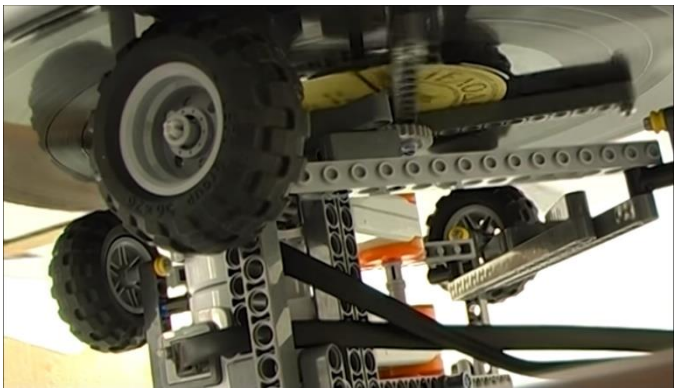


Рис. 2. Основной модуль MAUD. Вид снизу

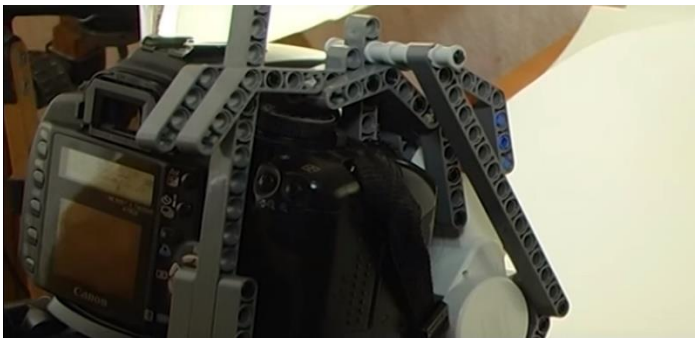


Рис. 3. Дополнительный модуль MAUD — универсальный обвес на снимающем устройстве.

Работа MAUD состоит из двух функциональных блоков:

- Блок вращения платформы на заданный угол.
- Блок съёмки изображения, следующий за вращением объекта.

Особое внимание следует обратить на блок съёмки объекта, т.к этот блок содержит проверку правильности поворота платформы, проверку события съёмки, с использованием датчика освещённости, и команду на саму съёмку объекта.Исходя из изученных функций, и справочного руководства [3], построим блок-схему алгоритма работы программы, загружаемой в контроллер (см.рис 3)

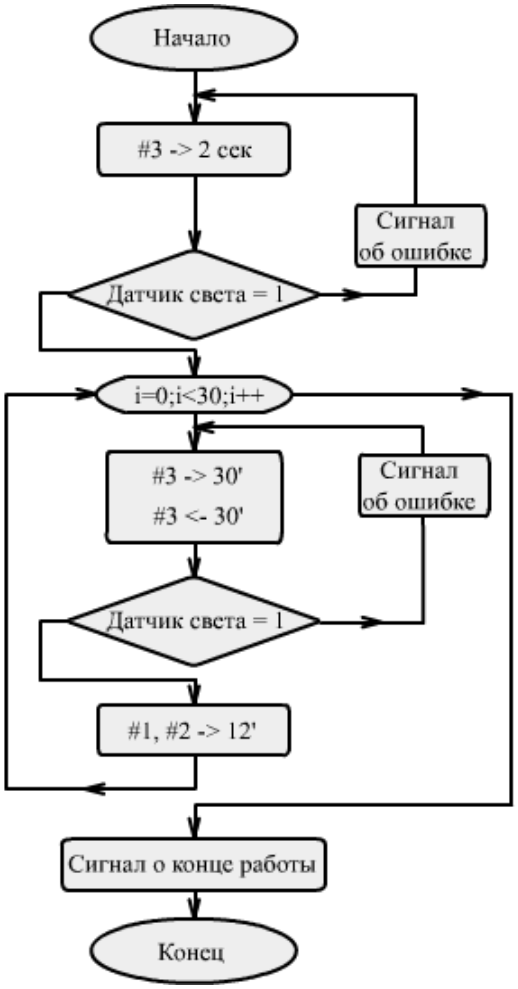


Рис. 4. Блок-схема алгоритма работы

Алгоритм, который изображён на блок-схеме (см. рис. 4), можно условно разделить на две части: отладочно -калибровочная часть, и рабочий цикл устройства. В отладочно - калибровочной части алгоритма, система производит проверку сервоприводов и датчиков на работоспособность, а так же проверяет исправность работы устройства съёмки. Это можно увидеть на первом, после начала, блоке алгоритма. Сервопривод №3, отвечающий за спусковой механизм снимающего устройства, производит вращение в прямом направлении в течение двух секунд до момента регистрации фотовспышки датчиком света. В случае, если фотовспышка не произошла, система сообщает о неполадке в работе системы, и повторяет действие до того момента, когда произойдёт тестовый снимок.

Дальнейшая работа алгоритма- повторяющийся цикл, в котором происходит рабочий снимок экспоната с проверкой, и вращение экспоната. После завершения цикла, система выводит сигнал о завершении работы, и выходит в режим ожидания, пока оператор меняет экспонаты, или принимает решение внести какие-либо изменения в работу системы.

В результате многочисленных экспериментов, отладки работы системы, изучения литературы [2], было определено оптимальное соотношение количества кадров и угла поворота объекта к объёму изображений на дисковом пространстве, и трудоёмкости обработки исходных фотографий. Это значение- тридцать кадров на экспонат. Такое распределение позволяет производить фотографию объекта каждые 12 градусов, что даёт возможность отснять минимум кадров без потери детализации экспоната.

Выводы

Результатом работы автоматизированной системы MAUD, после обработки исходных изображений в графическом редакторе, является объёмная модель экспоната, которая, в дальнейшем, будет занесена в электронный каталог горно-геологического музея ДонНТУ.

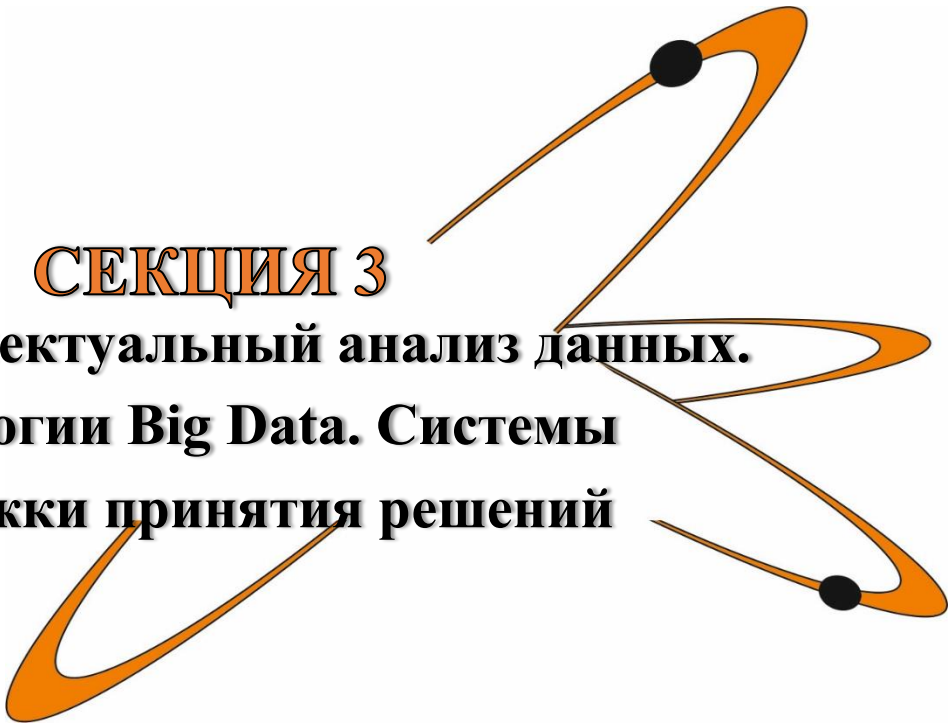
Теоретическое значение данного исследования состоит в способствовании ускорению создания каталога горно-геологического музея ДонНТУ. Данную технологию можно применить в любых областях, где могут потребоваться объёмные фотографии реально существующих объектов.

С практической точки зрения, разработка значительно ускорит процесс создания каталога, снизит затраты времени и трудового ресурса. В связи со сравнительно большим количеством экспонатов, доступных для реализации как части каталога, затраты могут превысить любую рентабельность. Разработка предназначена для упрощения деятельности по созданию каталога.

В дальнейших перспективах планируется реконструкция системы для увеличения точности и качества снимаемого материала, а также для увеличения массового порога снимаемых объектов. Также возможна реализация съёмки объектов с нескольких ракурсов, создания объёмных моделей человеческих лиц и частей тела для реализации создания индивидуальных протезов.

Библиографический список

1. Н.Столяров Ученые ДонНТУ приступили к созданию виртуального каталога геологического музея. // Донецкий Политехник, Спецвыпуск , Октябрь 2010. - стр.8.
2. В. В. Яншин. Анализ и обработка изображений: принципы и алгоритмы : учеб. пособие для студ. вузов, обуч. по направлениям "Информатика и выч. техника", "Радиотехника" / В- М. : Машиностроение, 1995. - 112 с.:ил. - (Для вузов).
3. Гагарина Л.Г. Алгоритмы и структуры данных: практикум, : М.:Финансы и статистика, 2009 год, 304 с.
4. Юревич Е. И Основы робототехники.. - 2-е изд., перераб. и доп. - СПб.: БХВ-Петербург, 2005. - 416 с: ил.
5. М. Предко Устройства управления роботами. Схемотехника и программирование (пер. с англ). ДМК Пресс. - 2005. - 404 с.



СЕКЦИЯ 3

Интеллектуальный анализ данных. Технологии Big Data. Системы поддержки принятия решений

УДК 681.5
Н.В. Балясный, магистрант
Научный руководитель: Е.Л. Первухина, д-р.техн. наук, проф.
Севастопольский государственный университет

ИНФОРМАЦИОННО-ВЫЧИСЛИТЕЛЬНАЯ ТЕХНОЛОГИЯ ДЛЯ ОПТИМИЗАЦИИ КОМПОНОВКИ ТЕХНИЧЕСКОЙ СИСТЕМЫ

Аннотация: Решается актуальная задача нахождения оптимальной компоновки автоматизированной производственной линии. Разработанная информационно-вычислительная технология позволяет избежать лишних трудовых и временных затрат.
Ключевые слова: многокритериальная оптимизация, компоновка системы, глобальный поиск
Annotation: An urgent problem of optimal automatic industrial line assembly solution search is to be solved. The proposed information and computing technology would avoid extra work and time costs.
Key words: multi-criteria optimization, system assembling, global search.

Рациональный выбор устройств для достижения оптимальной компоновки технической системы представляет сложную и актуальную задачу многокритериального выбора, когда критерий эффективности объектов представляется в векторном виде. Для целого ряда производственных приложений (например, компоновки автоматизированной производственной линии) поиск решения усложняется недостатком априорной информации об области поиска и нечеткой формулировкой цели, в том числе в силу противоречивости отдельных критериев. В этих условиях не всегда очевидно, какие функционалы могут быть использованы в качестве критериев, наилучших с точки зрения рационального решения производственных задач [2].

Целью работы является разработка информационно-вычислительной технологии для анализа компоновки технической системы по критериям совместимости составляющих устройств на основе векторов их характеристик, в том числе надежности и стоимости компоновки оборудования.

Обзор научно-технической литературы последних лет [3, 4], показал, что решения таких задач в открытых источниках отсутствуют, однако описаны математические подходы к их решению на основе методов глобальной оптимизации. Современное развитие веб-технологий и автоматизированных систем поддержки принятия решения позволяют упростить поиск возможных путей решения и создают необходимые условия для создания новой информационно-вычислительной технологии.

Предлагается декомпозировать задачу нахождения оптимальной компоновки на две подзадачи, последовательное решение которых обеспечивает нахождение наиболее эффективного компоновочного решения.

Подзадача 1.

$\exists N$классов устройств, элементов, i – номер класса,

Из каждого класса X_i выбирается только один механизм, вектор параметров которого x_i^j, где j – порядковый номер конкретного элемента в своем классе. Пусть все устройства работоспособны. Выбирается совокупность числа N для создания нового устройства.

Существует множество G, определяющее ограничения g_k, накладываемые на итоговую сборку пользователем.

Пусть Y– множество всех возможных компоновок, реализуемых системой. Описать процесс формирования этого множества можно следующим образом:

$$Y = f(X_1, X_2 \dots X_N), \quad (1)$$

где f – функция выбора.

Решение подзадачи ориентировано на определение множества поиска оптимума Y' пошаговым сужением всего множества альтернатив с учетом ограничений g, задаваемых пользователем, и требований совместимости h, вычисляемых для каждого устройства с использованием уравнений кинетической теории механизмов.

$$Y' = \varphi(Y, h, g), \quad (2)$$

где φ – функция ограничения множества альтернатив.

Подзадача 2.

Существует множество F, задающее векторный критерий эффективности, включающий показатели стоимости сборки rи надежности c. Показатель (критерий) стоимости для каждой компоновки вычисляется как суммарное значение всех затрат при эксплуатации набора устройств: стоимости приобретения, обслуживания и энергозатраты. Критерий надежности определяется как вероятность безотказной работы системы, вычисляемая на основе статистической информации о проблемах совместимости устройств.

Решением подзадачи является оптимальная компоновка на множестве Y', полученном на предыдущем шаге с использованием критериев rи c. Предлагаемая информационно-вычислительная технология обеспечивает компоновку, характеризуемую вектором параметров y', более эффективным (с позиции заявленного векторного критерия) чем исходный вектор y:

$$y' = Opt(Y', r, c) \quad (3)$$

Предположим, при определенных сочетаниях параметров отдельных элементов x_i^j новое устройство не может быть реализовано или реализовано, но не является работоспособным.

В этом случае при подаче на вход модели совокупности требований к системе, на выходе получится только часть ожидаемых вычислительных элементов y. Описание системы дополняется вектором признаков $\xi(y)$, указывающим для каждого функционала, вычислен он или нет. Такая модель будет считаться частично вычислимой, а модель, описанная ранее, – полностью вычислимой.

Для иллюстрации технологии предлагается поиск оптимума при осуществлении компоновки устройства для передвижения и обработки объектов, состоящего из трех основных механизмов компании «*Festo*»: конвейерной ленты (класс устройств *А*), выступающей в роли движущего и позиционирующего устройства, двигателя (класс устройств *Б*) и контроллера (класс устройств В).

Каждый элемент класса *А* характеризуется следующим вектором характеристик: скорость ленты, нагрузка, тяговая сила ленты, начальная цена, цена обслуживания, энергозатраты.

Каждый элемент класса *Б* можно задать параметрами: мощность, максимальный крутящий момент, тип, исходная цена, цена сервиса и энергозатрат.

Каждый элемент класса *В* характеризуется: мощностью, током, типом подходящего двигателя, ценой, ценой сервиса и энергозатрат.

В ходе системотехнического анализа с помощью выбранной в качестве основной нотации семейства методологий *IDEF* спроектирована диаграмма декомпозиции (рис.1).

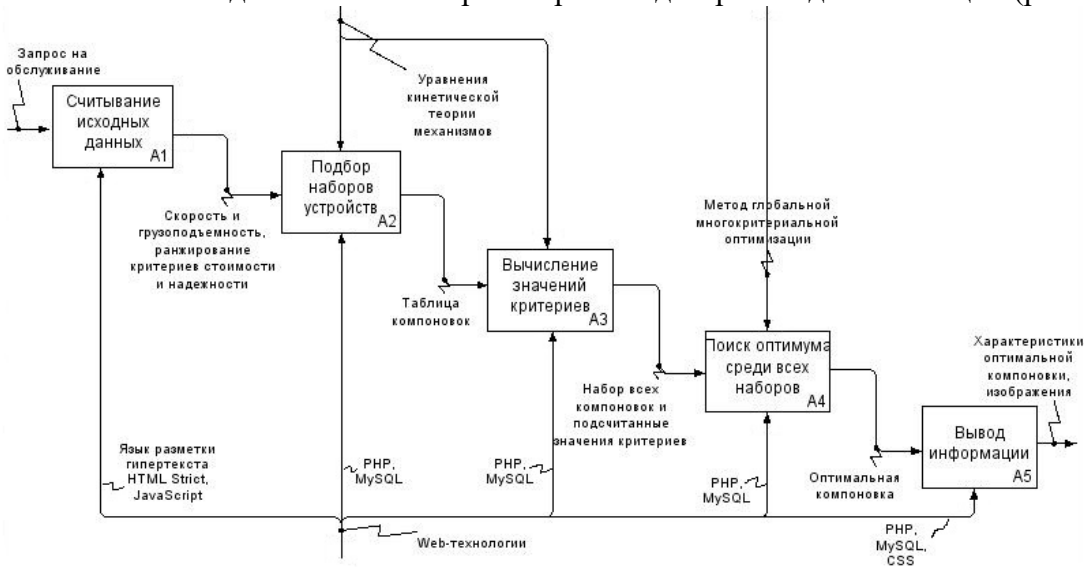


Рис.1. Диаграмма декомпозиции первого уровня

Представленные на диаграмме блоки характеризуют основные функции предлагаемой информационно-вычислительной технологии в виде последовательности шагов рис. 2:

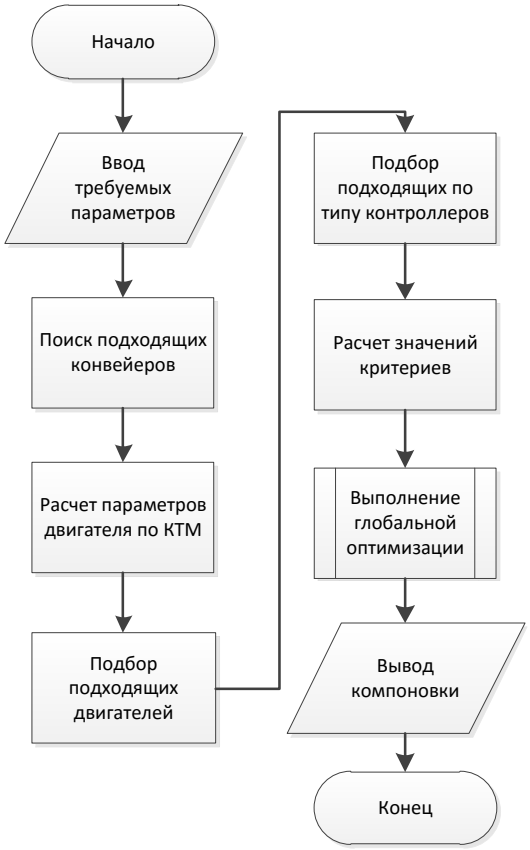


Рис.2. Последовательность шагов предлагаемой технологии

1. С помощью входной информации о скорости передвижения (V) и грузоподъемности (m), которые должно иметь итоговое устройство, система определяет список подходящих конвейеров из базы.

2. По уравнениям кинетической теории механизмов высчитывается требуемая мощность двигателя, способного поддерживать корректную работу автоматизированной линии.

Для определения параметров требуемых двигателей используется дополнительный параметр, хранящийся в базе данных для каждого конвейерного устройства. Это тяговая сила ленты F, необходимая для расчета потребляемой мощности привода по формуле:

$$P_{\text{птр}} = F * V \quad (4)$$

По потребляемой мощности привода определяется требуемая мощность двигателя по формуле:

$$P_{\text{н}} = \frac{P_{\text{птр}}}{\eta}, \quad (5)$$

где η – общий КПД привода, в данной работе принято $\eta = 0{,}96$ [1].

3. По типу и модели подобранного двигателя приложение находит в классе контроллеров подходящее устройство.

4. Анализируется оптимальность всего набора компоновок, удовлетворяющих ограничениям пользователя, с использованием критериев стоимости и надежности.

5. На заключительном этапе итоговый набор механизмов с конечными характеристиками, ценой и уровнем надежности представляется пользователю.

Как было описано выше, при анализе эффективности каждой конкретной компоновки используются два критерия. Критерий стоимости для каждого из составных частей итогового устройства описывается как сумма затрат на приобретение и обслуживание устройства.

Критерий надежности определяется на основе эмпирических данных статистики эксплуатации участвующих в анализе устройств о возникающих проблемах, связанных с их совместимостью.

Разработанная система относится к классу систем управления взаимоотношениями с клиентами (*CRM*) и может быть встроена в интернет-сайты компаний, содержащие каталог предлагаемых товаров. Любой пользователь, заходя на сайт, может с ее помощью получить объективную оценку эффективности выбранного комплекса устройств, его надежности и стоимости. Это позволит снизить уровень обращений в сервис предприятия, благодаря превентивному анализу возможных ошибок совместимости устройств.

Система протестирована на различных наборах входных данных, предусмотрена защита от ввода некорректных запросов, ошибки в поведении системы не выявлены.

Работа выполнена при частичной поддержке Министерства Образования и Науки РФ в рамках базовой части государственного задания.

Библиографический список

1. Решетов Д.Н. Детали машин: Учебник для студентов машиностроительных и механических специальностей вузов/Д.Н.Решетов. – 4-е изд., перераб. и доп. – М.: Машиностроение, 1989. – 496 с.
2. Сысоев А.В. Об одной информационно-алгоритмической модели процесса параллельного глобального поиска // Информационные технологии, Вестник Нижегородского университета им. Н.И. Лобачевского, 2011, №3, с. 293-300.
3. Jamian J.J. Global partical swarm optimization for high dimension numerical function analysis/J.J. Jamian, M.N. Abdullah, H.Mokhlis, M.W. Mustafa, A.H.A.Bakar // Int. J. Appl. Math., 2014. – p.14.
4. Weise T. Global optimization algorithms: Theory and application/T. Weise. – 3rd ed., 2011. – p.1223.

УДК 621.391
Ю.Э. Криворучко, студентка 4-го курса
Научный руководитель: В.С. Чернега, канд. техн. наук, доц.
Севастопольской государственный университет

ИССЛЕДОВАНИЕ СТАТИСТИЧЕСКИХ ХАРАКТЕРИСТИК ПРОБЛЕМНО-ОРИЕНТИРОВАННЫХ ТЕКСТОВЫХ СООБЩЕНИЙ

Аннотация: получены статистические характеристики лексем текстовых сообщений банковской тематики, которые могут быть использованы для повышения степени сжатия текстовых документов данной тематики.
Ключевые слова: статистическая характеристика, сжатие, частота, вероятность, банковский текст.
Annotation: obtained statistical characteristics of the lexemes text messages banking theme, which can be used for increasing the compression ratio of text documents on this theme.
Key words: statistical characteristics, compression, frequency, probability, bank text.

Системы сжатия информации используют статистические характеристики вероятности появления символов: слов, биграмм, триграмм, слов. Современные компрессоры рассчитаны на усредненные значения вероятности, полученные при анализе литературы на русском либо ином языке [1, 2]. Такая статистика отличается от статистических характеристик специализированных текстов, например, технические, экономические, медицинские и тому подобные. Поэтому компрессоры, которые рассчитанные на усредненную статистику будут сжимать статистические тексты неэффективно. В работе [3] показано, что использование полуадаптивных архиваторов проблемно-ориентированных сообщений с предварительно занесенными в таблицу кодирования наиболее часто используемых *n*–грамм, отдельных слов и словосочетаний повышает степень сжатия таких сообщений по сравнению с адаптивными архиваторами.

Целью работы является исследование статистических характеристик текстовых документов, используемых в банковской сфере и построения моделей банковских сообщений для применения их в специализированном архиваторе банковских сообщений.

При исследовании статистических характеристик анализировались электронные банковские тексты на русском языке. При анализе текста не учитывался регистр символов. Анализируя слова не учитывались пробельные символы и знаки пунктуации. Для исследования текстовых сообщения были выбраны следующие лексические единицы: русский алфавит, n-грамм (биграммы, триграммы), слова. Полученные результаты исследований оформлены в виде таблиц, графиков и гистограмм.

Моделью проблемно-ориентированного текста может быть совокупность вероятностей символов, n-грамм, слов. Представим банковский документ в виде вектора $d = (w_1, w_2, ..., w_m)$, где m - множество всех уникальных термов (символов, n-грамм, слов). w_i - вес i-го терма. Существуют различные способы определения веса терма. В данном случае будем использовать частотное вхождение терма в документ. Частота вхождения терма определяется как отношение числа вхождения терма к общему числу термов текста.

$$w_i = \frac{n_i}{\sum n_k}, \quad (1)$$

где n_i - количество вхождений i-го терма в d.

При больших объемах текста частота вхождения терма и вероятность появления терма в тексте равны. Таким образом модель текста выглядит следующим образом:

$$P(d) = \sum P_{симв}(w_i) + \sum P_{n-грамм}(w_i) + \sum P_{слов}(w_i). \quad (2)$$

Для исследования взят банковский текст, содержащий более 140 тысяч символов. Исследуем зависимость частоты встречаемости символов и их последовательности от количества символов в тексте. На рисунке 1 отображено изменение частоты символов «пробел», «а», «б». Изменение частоты появления биграмм «ст», «но» и триграмм «аци», «бан» от количества символов в тексте изображено на рисунке 2. На основе графиков можно

сделать вывод о том, что выбранный объем текста подходит для исследований, так как частота появления символа, биграмм и триграмм принимает устойчивое значение.

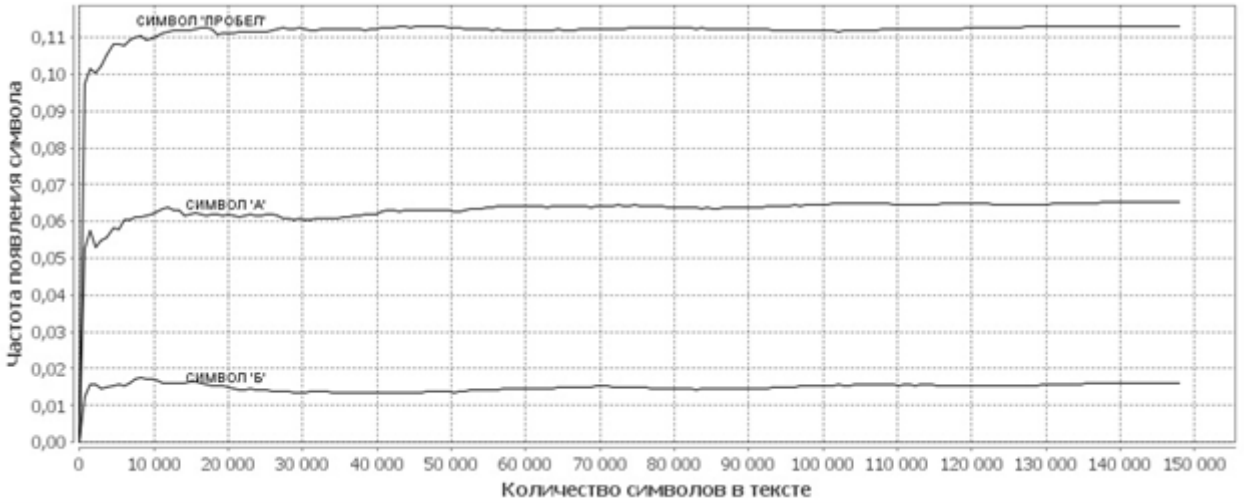


Рис.1. Зависимость изменения частоты появления символов от объема текста

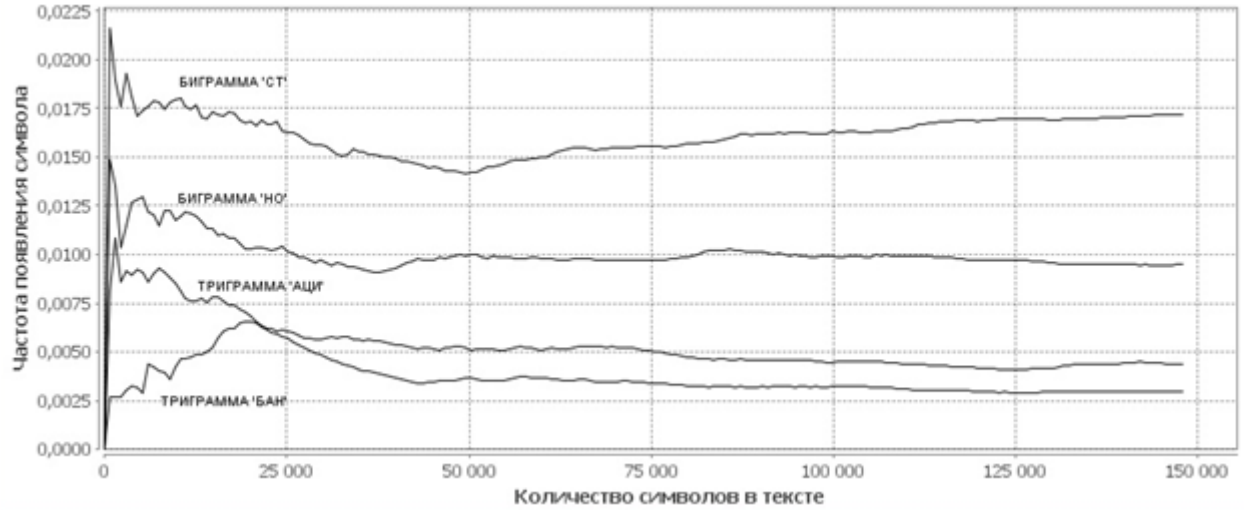


Рис.2. Зависимость изменения частоты появления биграмм и триграмм от объема текста

Статистические характеристики символов полученные в ходе исследования отображены в таблице 1. Символы, представленные в таблице, отсортированы в порядке убывания частоты появления символа в тексте.

Таблица 1 - Частота появления символов в текстовых сообщениях банковской тематики

Символ	Частота появления	Символ	Частота появления	Символ	Частота появления
пробел	0.11694	д	0.02555	,	0.00777
о	0.07766	м	0.02162	ж	0.00646
е	0.07741	ы	0.02016	ю	0.00550
и	0.07041	у	0.01760	щ	0.00452
а	0.06504	я	0.01759	)	0.00436
н	0.06350	б	0.01505	ф	0.00416
т	0.05702	ч	0.01126	–	0.00221
с	0.04797	г	0.01055	(	0.00217
р	0.04386	й	0.01038	;	0.00184
в	0.03778	ь	0.00848	э	0.00179
л	0.03093	ц	0.00846	ш	0.00168
к	0.03017	х	0.00834	:	0.00153
п	0.02735	.	0.00818	-	0.00118

Наиболее часто встречаемые биграммы, а также частота их появления в тексте отображены в таблице 2. Биграммы включают буквенные символы, а также пробельные символы и знаки препинания.

Таблица 2 – Частота появления биграмм в текстовых сообщениях банковской тематики

Биграмма	Частота появления	Биграмма	Частота появления	Биграмма	Частота появления
ен	0,01725	_д	0,00646	т_	0,00458
ст	0,01661	нн	0,00645	во	0,00454
_п	0,01401	ал	0,00638	ба	0,00436
и_	0,0124	то	0,00638	._	0,0043
ре	0,01225	В_	0,00626	ес	0,00425
_с	0,01172	ли	0,00624	ди	0,00421
ни	0,01171	_б	0,00621	ме	0,0042
е_	0,01134	х_	0,00615	ый	0,00407
ны	0,01132	ор	0,00609	не	0,00401
ов	0,01101	на	0,00607	ог	0,004
ан	0,01047	ва	0,00594	до	0,00398
_о	0,00986	че	0,0059	кр	0,00398
но	0,00971	ко	0,00567	ри	0,00397
_к	0,00961	_р	0,00566	ых	0,00385
ра	0,00958	ия	0,00558	ол	0,00379
та	0,0092	де	0,00553	нт	0,00367
ит	0,00907	го	0,00544	вл	0,00362
я_	0,00903	ел	0,00537	ом	0,0036
по	0,00879	об	0,00536	ля	0,00358
й_	0,00842	ер	0,00526	ые	0,00357
ед	0,00842	ль	0,00518	тн	0,00356
а_	0,00809	_н	0,00516	ав	0,00352
о_	0,00804	ие	0,00512	ас	0,00347
пр	0,00798	со	0,00504	ми	0,00347
ос	0,00795	ат	0,00501	тр	0,00344
_и	0,00792	нк	0,005	ла	0,00339
_в	0,00774	ти	0,00494	из	0,00338
,_	0,00759	ве	0,00492	од	0,00332
ка	0,00751	ци	0,00481	ем	0,0033
тв	0,00735	ле	0,00471	ии	0,0032
ет	0,00671	м_	0,0047	пи	0,00318
те	0,00668	за	0,00469	ся	0,00315
ро	0,00647	от	0,00463	ин	0,0031

Наиболее часто встречаемые слова, а также частота их появления в тексте отображены в таблице 3. В таблице так же присутствуют и отдельные символы, такие как «в», «и» так как разделение текста на слова выполнялось по разделителям (пробелам, знакам препинания).

Таблица 3 – Частота появления слов в текстовых сообщениях банковской тематики

Слово	Частота появления	Слово	Частота появления	Слово	Частота появления
и	0,03087	за	0,00365	как	0,00206
в	0,02875	к	0,00359	до	0,00194
по	0,01641	центральный	0,00329	является	0,00188

на	0,01605	ценные	0,00318	кредита	0,00188
банк	0,01	бумаги	0,00306	из	0,00182
с	0,00958	банки	0,003	через	0,00182
средств	0,00859	также	0,00288	кредитной	0,00176
банка	0,00688	о	0,0027	быть	0,00176
для	0,00647	счета	0,00259	документов	0,00171
при	0,00582	банков	0,00241	капитала	0,00171
от	0,00523	расчетов	0,00241	плательщика	0,00165
капитал	0,00523	срок	0,00235	клиента	0,00159
или	0,00512	организации	0,00235	договора	0,00159
собственный	0,00488	счет	0,00235	кредит	0,00159
а	0,00453	кредитных	0,00235	расчетных	0,00159
операций	0,00435	может	0,00235	ссуды	0,00159
их	0,00412	банком	0,00229	могут	0,00159
не	0,00412	между	0,00218	деятельности	0,00159
операции	0,00406	средства	0,00212	обязательств	0,00159
денежных	0,00388	за	0,00365	клиентов	0,00153
его	0,00382	цб	0,00206	услуг	0,00153

На основе полученных вероятностных характеристики текстовых сообщений банковской тематики можно составить таблицу, с учетом рационального размещения лексических единиц, для сжатия банковских текстов методами семейства LZW.

Библиографический список

1. Ватолин Д. Методы сжатия данных. Устройство архиваторов, сжатие изображений и видео / Д.Ватолин, А.Ратушняк, М.Смирнов, В.Юкин. – М.: ДИАЛОГ-МИФИ, 2002. – 384 с.
2. Чернега В.С. Сжатие информации в компьютерных сетях: Учебное пособие для вузов / В.С.Чернега. – Севастополь: Изд-во СевГТУ, 1997. – 214 с.
3. Чернега В.С. Оценка снижения коэффициента сжатия текстовых сообщений во время переходного периода работы компрессора / В.С.Чернега // Вестник СевНТУ, Вып.74. – Севастополь: 2006, с.21-28.

УДК 004.421
В.А. Миколюк, студент 4-го курса
Научный руководитель: В.С. Чернега, канд. техн. наук, доцент
Севастопольский государственный университет

АЛГОРИТМ УНИФИКАЦИИ МЕДИЦИНСКИХ ДАННЫХ

Аннотация: в данной статье была рассмотрена проблема унификации данных, полученных в ходе исследования пациента. Предложен алгоритм, по которому осуществляется объединение все типов электронного оборудования в одну информационную структуру.
Ключевые слова: DICOM, обработка данных, технический прогресс, алгоритм, стандарт, исследование.
Annotation: this article considers the problem of the unification of the data obtained in the course of the patient's studies. Algorithm was proposed, which is adapted to combine all types of electronic equipment by means of which, produced the study.
Key words: DICOM, data processing, technical progress, the algorithm, standard, study.

Неотъемлемой частью в верной постановке диагноза и дальнейшем лечении, является сбор достоверной информации с объекта исследования. На данный момент существует множество технический решений, направленных на исследование различных органов и групп органов человека. Актуальной проблемой при проведении исследования является задача унификации полученных данных, с целью возможного манипулирования результатами. Эта проблема возникла в результате того, что существуют медицинские электронные приборы, построенные по различным технологиям, и использующие различное программное обеспечение [1]. Особенно это касается приборов, выпущенных в 90-х годах, так как производители электронного оборудования не придерживались единых правил разработки. Поэтому необходим единый стандарт, который будет включать в себя все аспекты обработки данных. Таким стандартом является Digital Imaging and Communications in Medicine (DICOM) [2] – [4] . DICOM — отраслевой стандарт для манипулирования медицинскими изображениями и документами. Стандарт был разработан Национальной ассоциацией производителей электронного оборудования (National Electrical Manufacturers Association, NEMA), взявшей за основу стандарт Open System Interconnection (OSI), который создала Международная организация International Standards Organization (ISO).

Стандарт DICOM содержит два информационных уровня:

- файловый уровень — объектно-ориентированный файл, имеющий теговую организацией, для представления кадра изображения и сопровождающей или управляющей информации;
- сетевой уровень — протокол для передачи DICOM файлов и управляющих DICOM команд по сетям с поддержкой набора сетевых протоколов TCP/IP.

На файловом уровне содержится информация об пациенте, оборудовании, дате и времени, персонале, которые участвовали в исследовании. Также содержится изображение, полученное в результате исследования. Данная структура позволяет производителям электронного оборудования следовать единому решению, благодаря которому имеется возможность объединить медицинское оборудование разных производителей в единую клиническую информационную систему для дальнейшего анализа результатов исследования. Такая возможность не только облегчает работу персонала с использованием вычислительной техники, но и выгодна с экономической точки зрения.

Сетевой DICOM протокол описывает порядок передачи медицинской информации от медицинского оборудование в PictureArchivingandCommunicationSystem. Система предполагает создание специальных архивов с результатами исследований на DICOM серверах, необходимые для быстрого поиска и просмотра интересующей информации. Это позволяет создать единую DICOM-сеть, с помощью которой возможен обмен данными.

Целью данной работы, являться разработка алгоритма и программного модуля, позволяющий преобразовывать данные, полученные в результате исследования, к единому стандарту DICOM. Данное решение позволяет не только объединить электронное оборудование в одну группу, но и создать единую информационную сеть, в рамка которой, возможен обмен информацией между медицинскими подразделениями и учреждениями.

Входными данными для данного модуля является изображение, поступающее с медицинского электронного оборудования, а также сопутствующая информация, вводимая врачом во время проведения исследования. На выходе работы модуля получается документ в формате DICOM 3.0. В соответствии с требованиями стандарта DICOM программный модуль состоит из следующих блоков:

1. Блок выбора типа оборудования
2. Блок сбора информации об исследовании
3. Блок анализа и подготовки данных
4. Блок формирования DICOM документа
5. Блок анализа адекватности полученной модели

Функцией блока выбора типа оборудования является формирование меню с перечнем существующего оборудования, используемого при исследовании, из которого врач выбирает оборудование необходимое на данный момент. Это позволит настроить программный модуль для работы с выбранным оборудованием. В стандарте DICOM прописано наличие сопутствующей информации, такой как: время проведения исследования, ФИО пациента, ФИО врача, название исследуемого органа и т.д. За ввод метаданных отвечает блок сбора информации. Блок позволяет вводить метаданные по средствам графического интерфейса в соответствии со стандартом DICOM. Введённая информация хранится во временной памяти и будет использована далее. Блок анализа и подготовки данных предназначен для преобразования полученных в ходе проведения исследования данных. Сначала сжимается текстовая информация, по средствам метода RLE. Затем сжимается изображение или серия изображений с помощью алгоритма JPEG. Использование данных алгоритмов позволяет существенно уменьшить объем занимаемого дискового пространства без значительной потери качества информации. На следующем шаге формируется документ в формате DICOM 3.0 (*.dcm). Название документа врач указывает сам, но для обратной совместимости с более старым программным обеспечением, рекомендуется использовать не более 8 символов без пробелов. Документ сохраняется во временной памяти для дальнейшей проверки. После формирования и сохранения DICOM документа, требуется проверка полученной модели. Врач анализирует получившийся результат, при этом он может исправить допущенные ошибки или же провести исследование ещё раз. Результат представлен в виде итогового отчёта, с возможностью полного редактирования информации. Если же врач подтверждает результаты исследования, DICOM документ сохраняется в постоянную память. Количество таки документов будет равное количеству сделанных кадров во время исследования. Также сохраняется файл DICOMDIR — это специализированный DICOM файл, который служит для каталогизации информации, и содержит служебную информацию по всем DICOM файлам, находящимся на этом устройстве хранения.

В таком виде программный модуль требует участия врача, проводящего исследования, для получения сопутствующей информации. Например, выбор типа электронного оборудования или частоты излучения. Поэтому дальнейшем развитием данного алгоритма, будет является автоматизация и трёхмерная визуализация процесса исследования, что позволит избежать ошибок во время исследования и наглядно продемонстрировать результаты.

Библиографический список

1. Илясов Л.В. Биомедицинская измерительная техника/ Л.В. Илясов. – 2007. С. 342.
2. Лисачкина А. Б. Программный модуль для сохранения изображений, полученных при помощи ядерного магнитно-резонансного томографа в медицинском формате DICOM / А. Б. Лисачкина. – 2012. С. 70–73.
3. Пронин И.Н. Программное обеспечение для работы с данными в формате DICOM на IBM PC / И.Н. Пронин, П.В. Родионов, Л.М.Фадеева и др. // НИИ нейрохирургии им. акад. Н.Н. БурденкоРАМН – М.: №2, 2002, С.138–142.
4. Digital Imaging and Communications in Medicine (DICOM). – http://medical.nema.org/dicom

УДК 519.816:004.414.2
Д.А. Немков, студент 6-го курса
Научный руководитель: А.В. Колесников, доцент
Севастопольский государственный университет

СИСТЕМА ПОДДЕРЖКИ ПРИНЯТИЯ РЕШЕНИЙ В ПРОЦЕССЕ ПРОВЕДЕНИЯ ИСТОРИКО-ПОИСКОВЫХ РАСКОПОК

Аннотация: В данной статье рассмотрены этапы проведения поисковых работ по увековечиванию памяти погибших при защите Отечества в период Великой Отечественной Войны 1941-1945 годов. Проанализировав особенности этапов поисковых работ автором выделяются задачи и функции системы поддержки принятия решений для поисковой деятельности, проводится попытка описания подобной системы.
Ключевые слова: увековечивание памяти; поисковая работа; СППР
Annotation: This article describes the stages of prospecting work to perpetuate the memory of those killed in the defense of the Motherland during the Great Patriotic War of 1941-1945. After analyzing the features of the stages of prospecting works the author allocated tasks and decision support systems for the functions of search operations, carried out an attempt to describe such a system.
Key words: the perpetuation of the memory; research work; decision support system

Исследовательские возможности, предлагаемые информационными технологиями, оказались востребованными в процессе проведения историко-поисковых раскопок и реализуются средствами различных геоинформационных систем и систем поддержки принятия решений, с помощью которых реализуются потребности картографирования данных в исторических исследованиях, предлагаются решения на основе выявленных закономерностей и позволяют использовать такие свойства информационных систем как наглядность, интерактивность и доступность.

Обычно проведение поисковых работ включает в себя следующие этапы:

- архивно-документальная предварительная поисковая работа;
- полевая разведывательно-поисковая работа;
- полевая эксгумационно-поисковая работа;
- архивно-документальная завершающая поисковая работа.

Архивно-документальная поисковая работа в архивах первую очередь заключается в изучении первоисточников архивных документов с целью получения информации для организации проведения полевых работ, выясняется, в какой период времени на данной территории происходили боевые действия, какие части или соединения участвовали в этих событиях. Установление характера боевых действий начинается с изучения журналов боевых действий частей, соединений, армий и фронтов. Далее изучаются – боевые донесения, боевые приказы, приказания, распоряжения, оперативные сводки и оперативные карты. По завершении архивно-документальной предварительной поисковой работы принимается решение о целесообразности проведения на данном участке полевой разведывательно-поисковой работы.

Поисковая работа в местах, где велись боевые действия, проводится при наличии (полученного в установленном порядке) разрешения от постоянно действующей аттестационной комиссией Военно-мемориального центра Вооруженных Сил Российской Федерации (ВМЦ ВС РФ). Полевая разведывательно-поисковая работа проводится с целью определения непосредственных границ участков с наличием незахороненных останков и неизвестными захоронениями. По завершении работ определяются объем, сроки и предполагаемые трудовые затраты на полевую эксгумационно-поисковую работу.

Полевая эксгумационно-поисковая работа проводится с целью поиска и эксгумации незахороненных останков на исследуемой территории, а также останков из неизвестных захоронений. Обнаруженные в ходе полевой поисковой работы останки погибших воинов временно захораниваются поисковыми объединениями с целью дальнейшего перезахоронения в установленных исполнительными органами местного самоуправления местах с отданием воинских почестей в торжественной обстановке.

Архивно-документальная завершающая поисковая работа основывается на материалах архивно-документальной, разведывательно- и эксгумационно-поисковой работы и завершается установлением имен погибших или пропавших без вести при защите Отечества или его интересов в войнах и военных конфликтах, останки которых были обнаружены в ходе поисковой работы, и оповещением их родственников либо принятием решения о невозможности идентификации обнаруженных останков.

Таким образом, на основе приведенных выше этапов поисковой работы, можно выделить следующие необходимые задачи для системы поддержки принятия решений в процессе проведения историко-поисковых раскопок: обеспечение возможности объединения современных полевых, архивных и картографических материалов для совместного анализа, а также принятия решений по планированию и проведению поисковых работ, наглядное представление текущих и окончательных результатов этих работ.

Одной из задач проектирования информационной системы является определение ее функций, таких как:

1. Редактирование данных – возможность информационной системы принимать команды на редактирование данных со стороны пользователя.
2. Поисковые функции – поиск информации в информационном хранилище и представление полученных данных для пользователя.
3. Функции безопасности.

– защита доступности данных;
– защита целостности данных;
– защита конфиденциальности информации.

4. Расчетные функции – обработка информации, которая находится в системе (по определенным алгоритмам).
5. Аналитические функции – проведение операций над данными, результатом которых является прогностическая информация.
6. Точность данных – является основой для географической привязки данных.
7. Визуализация различных слоев данных.

Для достижения поставленных целей и успешного решения задач, определенных выше, система может иметь следующую структуру (рис.1):

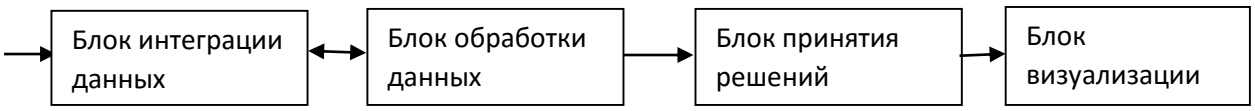


Рис.1. Примерная структура СППР.

В качестве формального описания системы было использовано обобщенное описание:

$$S=<X,Y,Z,H,D,\{m\}>, \quad (1)$$

где X – входные данные;
Y – выходные данные;
Z – состояния системы;
H – оператор переходов;
D – критерии к данным;
$\{m\}$ – элементы системы;
Множество входных данных:

$$X=\{A,B\}, \quad (2)$$

где $A=[(x_1,y_1),\ldots,(x_n,y_n)]$ – массив координат мест, где обнаружены останки;

$B=[\vec{B}1,\ldots,\vec{B}n]$ – векторы, полученные из привязанных к местности исторических карт.

Множество выходных данных:

$Y = \{S\}$ – некоторая площадь поверхности.

Множество состояний системы:

$$Z=\{z_0,z_1,z_2,z_3\}, \quad (3)$$

где z_0 – начальное состояние системы;

z_1 – ввод (чтение) исходных данных;

z_2 – обработка и анализ данных;

z_3 – вывод результатов, построение графиков.

Переход из одного состояния в другое осуществляется последовательно по мере обработки данных.

Таким образом, новое состояние будет определятся с помощью оператора перехода, начальный вид которого:

$$Z(t+1)=H(Z(t_0),X), \quad (4)$$

где $Z(t_0)$ – начальное состояние системы.

Множество критериев к данным имеет вид:

$$D=\{F_1,\ldots,F_n(Z,X)\}, \quad (5)$$

где $F_1,\ldots,F_n$ – различные функции выбора и обработки данных.

Множество элементов системы имеет вид:

$$m = \{m_1,m_2\}, \quad (6)$$

где m_1 – модуль распознавания данных с исторической карты (с заданными требованиями);

m_2 – модуль анализа и визуализации результатов.

Таким образом, была предпринята попытка найти такой вариант системы, который бы по максимуму удовлетворял заданным функциям и требованиям к системе.

Библиографический список

1. Методические указания «О порядке проведения поисковых работ» изданы Военно-мемориальным центром Вооруженных Сил РФ г. Москва, 2006 г.
2. Сайт «Союза поисковых отрядов России» http://sporf.ru/ (03.03.2016)
3. Постановление главы администрации Краснодарского края «О Положении о порядке организации и проведения поисковых работ» от 11.07.2000 г. №519 г. Краснодар. (В редакции Постановления Главы администрации (губернатора) Краснодарский край от 18.08.2010 г. №706)

УДК 004.415
Д.Д. Шевченко, студент 4-го курса
Научный руководитель: Е.Н.Мащенко, к.т.н., доцент
Севастопольский государственный университет

ИССЛЕДОВАНИЕ АКТУАЛЬНЫХ МЕТОДОВ ОБНАРУЖЕНИЯ ПЛАГИАТА В ТЕКСТОВЫХ ДОКУМЕНТАХ

Аннотация: проведен сравнительный анализ методов обнаружения плагиата в текстовых документах. В качестве основных методов были рассмотрены: метод «шинглов», метод «I-Match», метод «опорных» слов. Определены основные показатели эффективности методов.
Ключевые слова: плагиат, метод шинглов, I-Match, текстовый анализатор.
Annotation: A comparative analysis of the methods of detection of plagiarism in text documents. The main methods were considered: the method of "shingles", the use of a thesaurus, the method of «I-Match», the method of most commonly used word. The main indicators of the effectiveness of methods.
Keywords : plagiarism, shingles method , I- Match, text analyzer .

Введение

На современном этапе развития информационных технологий все большое значение приобрели проблемы защиты авторских прав в связи с монетарными и репутационными рисками, а так же ростом избыточности информации, проявляющемся в распространении плагиата авторских текстов.

Согласно определению в [1], плагиат − это умышленное присвоение авторства чужого произведения искусства или науки, чужих идей или изобретений. Плагиат в сфере научной литературы проявляется в полном или частичном копировании авторского текста, переноса авторской информации с перефразированием без указания источника заимствования. Плагиат может быть нарушением авторско-правового и патентного законодательства, и в связи с этим, может повлечь за собой юридическую ответственность. Проблематика обнаружения плагиата в текстовых документах складывается из следующих аспектов:

– Большие объемы информации делают задачу обнаружения плагиата алгоритмически и технически сложной.

– Существование большого количество неаутентичных видов плагиата, в свою очередь, так же усложняет задачу их обнаружения.

– Большая часть существующих алгоритмов обнаружения плагиата являются строго ориентированными на конкретный его вид, и неэффективны при комбинировании нескольких видов плагиата.

В связи с этим актуальной является задача разработки компьютерных систем выявления плагиата в текстовых документах, использующих в процессе анализа текстовой информации комплекс разноплановых алгоритмов и ориентированных на диалоговое человеко-машинное взаимодействие с целью повышения качества анализа текстов при минимизации времени обработки.

1. Обзор основных видов плагиата в текстовых документах

В соответствии с [2], плагиат в текстовой информации можно условно разделить на следующие виды:

– Дословное копирование – полный перенос авторской информации без указания источника копирования.

– "Мозаичный" плагиат или компиляция − соединение фрагментов из разных источников без ссылки на них и без оформления цитат в тексте.

– Перефразирование (рерайтинг) текстов чужих работ с помощью небольшой замены или простой перестановки слов. Может быть выполнен с помощью программных систем.

– Включение в текст оформленной цитаты (в кавычках) без ссылки на источник.

– Использование материалов, полученных в результате совместной деятельности (групповая работа, "мозговой штурм" и др.), без указания на источник происхождения и ссылок на членов авторского коллектива.

– Перевод и использование материалов иностранных авторов без указания первоисточника.

Поисковые алгоритмы, как правило, эффективно выявляют первые два вида плагиата. При исследовании остальных видов плагиата применим термин «нечеткий дубликат». Согласно [3] «нечеткий дубликат» − это документ, в котором основной смысл и стилистика первоисточника сохранена и при этом нет полного дублирования. Нечеткие дубликаты позволяют предположить, являются ли два объекта частично одинаковыми или нет [4].

Использование комбинации поисковых алгоритмов позволяет адаптировать работу компьютерных систем поиска плагиата в текстовых документах, при этом необходимо определить два целевых аспекта: 1) критерий определения похожести текстов (по форме или по содержанию); 2) определение оценки степени похожести документов и ее порогового значения, до которого тексты не считаются дубликатами.

2. Анализ методов обнаружения плагиата

В процедурах проверки текстовых документов на плагиат можно выделить три основных этапа:

1. Эвристический поиск – из каждого документа из списка документов для сравнения (корпуса документов) отбираются отдельные тексты, которые содержат фрагменты проверяемого документа, на основе полученных значений о наличии соответствий и их объеме производится предварительная оценка информационной ценности анализируемого текста.
2. Подробный анализ – для каждого документа последовательно анализируется степень сходства с проверяемым документом, и на основе полученных значений делается вывод об аутентичности проверяемого документа.
3. Пост-обработка – анализ совпадений, полученных при подробном анализе, проводимый для исключения ложных совпадений.

В работе [5] для формального определения дубликатов и нечетких дубликатов введена мера близости − $\rho(x_i, x_j)$, значения которой изменяются в интервале от нуля до единицы. Значения x_i, x_j – публикации, представленные в виде векторов, элементы векторов – термины, использованные в тексте документа. Мера близости должна равняться единице в случае, если документы x_i и x_j – дубликаты, и стремиться к нулю, в противном случае. Тогда: два документа x_i, x_j считаются полными дубликатами, если мера близости $\rho(x_i, x_j)=1$; два документа x_i, x_j считаются нечёткими дубликатами, если мера близости $\rho(x_i, x_j)$ превосходит экспериментально установленный порог Θ, $\rho(x_i, x_j)\geq\theta$ Пороговое значение Θ выбирается экспертным путем. В зависимости от применяемых мер сравнения текста и методов поиска плагиата для дальнейшего исследования выбраны [5-8] следующие методы:

1) Локальные методы. К ним относятся методы, использующие хеш-функцию (*MD5, SHA-2, CRC32*); метод шинглов и его модификации; методы, использующие семантические сети; частотные методы.

2) Глобальные методы: сигнатурные методы; методы «опорных слов».

Основная идея локальных методов сводится к синтаксическому анализу текстового документа. В результате анализа документу ставится в соответствие некоторое количество сигнатур. Примером реализации метода являются алгоритмы, вычисляющие хеш-функцию (MD5, *SHA-2, CRC32*) от конкатенации двух самых длинных предложений в документе. Вычисленная хеш-функция является сигнатурой документа. Недостатки данного метода: при корректировке пользователем самых длинных предложений документа метод перестает корректно работать.

Более эффективным способом нахождения нечетких дубликатов является метод, основанный на вычислении отношения числа вхождения некоторого слова к общему количеству слов документа (*TF — term frequency* — частота слова). По частоте оценивается значимость слова в пределах отдельного документа. Для каждого слова в документе

вычисляется его вес, равный отношению числа вхождения этого слова к общему количеству слов документа. Далее сцепляются *n* упорядоченных слов с наибольшими весовыми коэффициентами и вычисляется хеш-функция.

Метод шинглов (подпоследовательностей) разработан для поиска копий и дубликатов рассматриваемого текста в веб-документе. Алгоритм метода состоит из следующих основных этапов [6]:

– канонизация текста (приведение текста к единой нормальной форме);

– разбиение на шинглы (выделенные из статьи подпоследовательности слов, зачастую используются шинглы 3-10 слов);

– вычисление хэш-функций шинглов;

– случайная выборка 84 значений контрольных сумм;

– сравнение, определение результата (заключается в сравнении случайной выборки контрольных сумм шинглов (подпоследовательностей) двух текстов между собой).

Основным недостатком алгоритма является экспоненциальный рост числа сравнений при росте числа шинглов, что, в свою очередь, порождает необходимость увеличивать длину каждого из них. В свою очередь, увеличение длины шингла снижает эффективность алгоритма.

Методы анализа с использованием семантической сети. Задача определения факта заимствования сводится к сравнению моделей, отражающих смысловую нагрузку текстов. Анализ ведется с использованием алгоритмов на графах, модифицированных и оптимизированных для применения в рамках данной задачи. Использование схем анализа данных в этом методе может позволить выявлять факт заимствования, даже если оригинал был определенным образом модифицирован (выполнен перевод, слова были заменены на синонимы, текст был изложен с использованием другой лексики и т.д.).

Глобальные сигнатурные методы. Дальнейшим развитием метода, использующего меру *TF*, стал алгоритм, анализирующий все документы выборки. В нем используются мера *TF-IDF*. *IDF* (*Inverse Document Frequency* — обратная частота документа) — инверсия частоты, с которой некоторое слово встречается в документах выборки.

Метод *I-Match*. Концепция данного метода состоит в вычислении дактилограммы (уникального идентификатора содержания текста документа) для представления содержания документов. С этой целью сначала на основе проверяемых документов строится словарь, включающий слова со средними значениями IDF. Такие слова, как правило, обеспечивают более точные результаты при обнаружении нечетких дубликатов. Слова с большими и маленькими значениями IDF отбрасываются, как не значащие [6]. Затем для каждого документа формируется множество слов, входящих в него, и определяется пересечение множества слов для документа и словаря. Если размер этого пересечения больше некоторого минимального порога (определяемого экспериментально), то список слов, входящих в пересечение упорядочивается и для него вычисляется *I-Match* сигнатура (хеш-функция). Два документа считаются похожими, если у них совпадают *I-Match* сигнатуры (имеет место коллизия хеш-кодов). Алгоритм имеет высокую вычислительную эффективность при сравнении небольших по размеру документов. Основной недостаток — неустойчивость к небольшим изменениям содержания документа.

Метод «опорных» слов. Изначально из документа выбирается множество из N слов, называемых «опорными». Значение N определяется экспериментально. В дальнейшем, каждый текстовый документ выглядит как N-мерный двоичный вектор, в котором i-я координата равна 1, если i-е «опорное» слово имеет в документе относительную частоту выше определенного порога (устанавливаемого отдельно для каждого «опорного» слова) и равна 0 в противном случае. Этот двоичный вектор и является сигнатурой документа. Соответственно, два документа считаются идентичными при совпадении сигнатур. При построении множества «опорных» слов используются следующие правила:

1) Множество слов должно охватывать максимально возможное число документов.

2) Число слов в наборе должно быть минимальным.
3) «Качество» слова должно быть максимально высоким.

Если предположить, что «внутридокументная частота» — это нормированная частота слова в документе TF, которая находится в диапазоне 0..1, тогда единица в данном случае будет соответствовать наиболее частому слову в документе. Распределение документов по данной «внутридокументной частоте» строится для каждого слова однократно. Далее алгоритм разбивается на несколько этапов, каждый из которых состоит из двух фаз. В первой фазе увеличивается покрытие документов в индексе при фиксированной (ограниченной снизу) точности. Во второй фазе уже увеличивается точность при фиксированном покрытии. В данном случае точность будет максимально высокой, если повторение слова в дельта-окрестности значения относительной частоты минимально. Частота, которая имеет наибольшую точность, получила название пороговой. Поэтапно сортируются самые неподходящие слова, а когда наступает последняя стадия алгоритма, то остаются только группы слов, которых достаточно для обеспечения качественного покрытия. Алгоритм позволяет существенно отфильтровать исходный набор слов.

Заключение

В рамках дальнейших исследований предполагается разработать компьютерную систему анализа текстовых документов на плагиат с учетом специфики научных документов (структурированность документов, наличие терминов и сокращений, наличие слов-клише, устойчивых словосочетаний). С целью повышения эффективности работы системы планируется использование комплекса алгоритмов с возможностью адаптивной настройки оценочных метрик документов-прототипов.

Библиографический список

1. Плагиат [Электронный ресурс] // Википедия: свободная энцикл. – Электрон. дан. – [Б.м.], 2016 –URL: ru.wikipedia.org/wiki/ Плагиат - (дата обращения: 05.03.2016)
2. Плагиат [Электронный ресурс]// Научно-техническая библиотека Томского Политехнического университета – Электрон. дан. – URL: http://www.lib.tpu.ru/plagiatus.html - (дата обращения: 05.03.2016)
3. Дубликат [Электронный ресурс] — Режим доступа: http:// ru.wikipedia.org/wiki/Дубликат — 16.03.2013 г. — Загл. с экрана. - (дата обращения: 05.03.2016).
4. Квашина Ю. А. Методы поиска дубликатов скомпонованных текстов научной стилистики // ТАРП . 2013. №1 (11). URL: http://cyberleninka.ru/article/n/metody-poiska-dublikatov-skomponovannyh-tekstov-nauchnoy-stilistiki (дата обращения: 01.03.2016).
5. Дербенёв Н. В., Толчеев В. О. Что можно улучшить в наукометрическом анализе — учет наличия дубликатов и заимствований в научных публикациях // УБС . 2013. №44. URL: http://cyberleninka.ru/article/n/chto-mozhno-uluchshit-v-naukometricheskom-analize-uchet-nalichiya-dublikatov-i-zaimstvovaniy-v-nauchnyh-publikatsiyah (дата обращения: 21.03.2016).
6. Зеленков Ю.Г., Сегалович И.В. Сравнительный анализ методов определения нечетких дубликатов для WEB-документов // Труды 9-ой Всероссийской научной конференции «Электронные библиотеки: перспективные методы и технологии, электронные коллекции» RCDL'2007: Сб. работ участников конкурса, том 1. Переславль-Залесский, Россия: «Университет города Переславля», 2007. С. 166-174
7. Шиков А. Н., Сорокин Д. С.Применение метода токенизации для поиска схожих фрагментов кода в студенческих работах // Образовательные технологии и общество. 2014. №2. URL: http://cyberleninka.ru/article/n/primenenie-metoda-tokenizatsii-dlya-poiska-shozhih-fragmentov-koda-v-studencheskih-rabotah (дата обращения: 06.03.2016).
8. Зиберт А. О., Хрусталев В. И. Разработка системы определения наличия заимствований в работах студентов высших учебных заведений. Методы предварительной обработки текста // Universum: технические науки . 2014. №4 (5). URL: http://cyberleninka.ru/article/n/razrabotka-sistemy-opredeleniya-nalichiya-zaimstvovaniy-v-rabotah-studentov-vysshih-uchebnyh-zavedeniy-metody-predvaritelnoy (дата обращения: 06.03.2016)

УДК 004.62:[004.89:519.876.5]
Ю.Е. Шишкин, магистрант
Научный руководитель: А.В. Скатков, д-р техн. наук, проф.
Севастопольский государственныйуниверситет

ИССЛЕДОВАНИЕ МЕТОДОВ АНАЛИЗА СЛАБОСТРУКТУРИРОВАННЫХ ДАННЫХ ДЛЯ МАШИННОГО ОБУЧЕНИЯ

Аннотация: Интенсивный рост требований к оперативности систем поддержки принятия решений приводит к дефициту актуальных структурированных статистических данных для принятия решений.Традиционные подходы становятся малоэффективными.Перспективным направлением повышения эффективности обучения систем принятия решений является извлечение знаний из слабоструктурированных данных. Показано преимущество метода машинного обучения с помощью больших данных.
Ключевыеслова: слабоструктурированные данные, большие данные, динамическое программирование, интеллектуальный анализ данных, машинное обучение
Annotation: Rapid growth of requirements to decision-making support systems efficiency leads to a shortage of relevant structured statistical data for decision making.
The rapid growth of requirements to efficiency of decision-making support systems leads to a shortage of relevant structured statistical data for decision making. Traditional approaches are ineffective. Perspective way to increase the efficiency of decision-making training systemsis the extraction of knowledge from semi-structured data.The report shows advantage of using Big Datamethod for machine learning.
Keywords: semi-structured data, big data, dynamic programming, data mining, machine learning

Информационные технологиии методологическая база, существующие на сегодняшний день, позволяют хранить, анализировать и использовать большие объемыинтенсивно меняющихся и сложно организованных данных. Достаточно подробно описаны в литературе методы структуризации, кластеризации и визуализации неструктурированных и слабоструктурированных данных. Использование методов анализа данных позволяет оперативно решать различные прикладные задачи, такие как оптимизация процесса мониторинга распределенных вычислительных сетей [1] и классификация информационных ситуаций в облачных средах [2]. В этой связи представляет интерес развитие новыхметодов автоматизированного анализа данных с целью извлечения из них новых, неизвестных, нетривиальных и практически полезных знаний(DataMining).Применение методов DataMining подразумевает работу с большими объемами данных, в контексте данной статьи уместно использование понятия «Большие Данные» (BigData) для обозначения собственно данных и методов работы с ними. Исходя из особенностей организации работы cBigData возникает необходимость использования различных подходов к сбору достаточного объема репрезентативных выборок данных [3] и создание единой концептуальной модели взаимодействия компонентов такихсистем, особенно это актуально дляорганизации крупномасштабных научных исследований [4].

Трудность, перед которой стоит исследователь, зачастую заключается в физической природе функционирования исследуемой системы. Практическая полезность полученных результатов, в таком случае, заключается в поиске закономерностей среди проявлений изучаемой системы, на основании которых строится адекватно отражающая связи внутри системы модель, что позволяет производить прогнозирование динамики изменения изучаемых систем и строить для них системы поддержки принятия решений.

В существующих подходахпроектирования программных систем поддержки принятия решений предполагается формирование управляющих решений в соответствии с паттернами поведения, выявленными у эксперта[5]. Существуют прикладные области демонстрирующие исключительную эффективность данной стратегии [6].Остается не выясненным вопрос о том, как будет вести себя система при недостаточном объеме статистических данных обеспечивающих наилучшее решение критических ситуаций. Кроме того, предикативная модель строится только для одной итерации, из чего возникает предположение о необходимости построениямодели предсказывающей всю цепь состояний управляемого объекта.

Следует признать, что вопрос автоматизированного обучения автономных систем на основе слабоструктурированных данных относится к наименее изученным задачам

DataMining[7].Классическая модель обучения реализуется в виде конечного набора состояний и переходов между ними, процесс перехода сопровождается вознаграждением, которое, в свою очередь, может быть максимизировано выбором оптимальной последовательностью переходов.

Ставится задача проектирования системы позволяющей, на основе большого объема слабоструктурированных данных, максимально автоматизировано строить управляющие матрицы поведения автономного роботизированного устройства (агента). Произвести оценку сходимости показателей эффективности алгоритма агента к области аттрактора на основе результирующей поведенческой матрицы в зависимости от объемов обучающей выборки, сравнить эффективность применения стандартных методов прогнозирования (динамического программирования, Монте-Карло).

На схеме взаимодействия агента и окружающей среды (рис.1) показано, что состояние агента в каждый следующий момент времени однозначно определяется состоянием среды и вознаграждением, полученным в результате действий агента на предыдущей итерации.

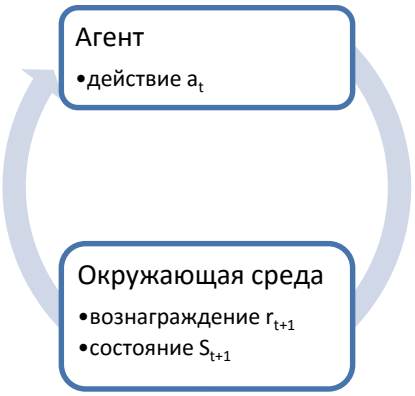


Рис.1. Схема взаимодействия агента и окружающей среды.

Математическая модель приведенной схемы взаимодействия широко представлена в [8], и в дискретном виде описывается уравнением Белмана:

$$V^{\pi}(s)=\sum_{a}\pi(s,a)\sum_{s'}P^{a}_{ss'}\left[R^{a}_{ss'}+\gamma V^{\pi}(s')\right],\ \text{где}$$

s — текущеесостояние принадлежащее к множеству S;

a — действие принадлежащее к множеству A(s), допустимых в состоянии s;

$\pi(s,a)$ — стратегия выбора aв состоянии s;

$P^{a}_{ss'}$ — вероятность перехода из состояния sв состояние s'в случае реализации сценария a.

$R^{a}_{ss'}$ — вознаграждение при переходе из состояния sв состояние s'в случае реализации сценария a;

Типовым подходом реализации системы подсчета вознаграждений для цепи итераций является введение функции кумулятивного вознаграждения:

$$V^{\pi}(s)=\sum_{a}\pi(s,a)Q^{\pi}(s,a),\quad Q^{\pi}(s,a)=\sum_{s'}P^{a}_{ss'}\left[R^{a}_{ss'}+\gamma V^{\pi}(s')\right],$$

которая при объединении с функцией оценки образует уравнение Белмана в виде:

$$Q^{\pi}(s,a)=\sum_{s'}P^{a}_{ss'}\left[R^{a}_{ss'}+\gamma\sum_{a'}\pi(s',a')Q^{\pi}(s',a')\right],$$

решение которого позволяет получить оптимальную стратегию принятия решений агентом.

Для иллюстрации процесса обучения построим имитационную модель, описывающую состояние внешней среды и обеспечивающую возможность взаимодействиясреды с агентом.

Применение таких моделей имеет практическую значимость вследствие того, что позволяет получить достаточно большой объем репрезентабельной выборки, необходимый для корректного обучения агента. При рассмотрении модели в динамике можно произвести учет ограничений аппаратных ресурсов агента, в частности на графике (рис. 2а) показаны результаты моделирования системы агент-среда свидетельствующие о значительном увеличении времени реакции и средней длины очереди необработанных воздействий по мере увеличения интенсивности поступления внешних воздействий при неизменной производительности аппаратных ресурсов агента. На графике зависимости функции штрафа и времени выполнения поведенческого алгоритма от размера обучающей выборки (рис. 2б) хорошо видно, что по мере увеличения объема выборки функция штрафа монотонно снижается, а время выполнения алгоритма стремится к точке аттрактора.

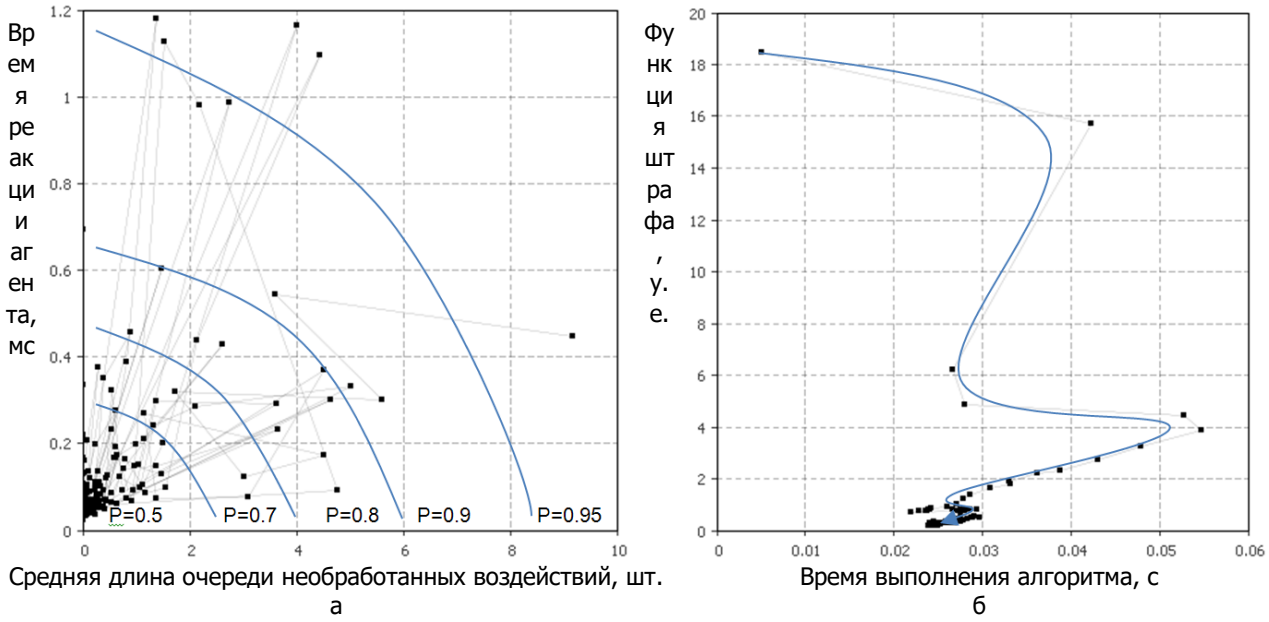


Рис.2. График эффективности процесса обучения: а — зависимость времени реакции агента на изменение состояния внешней среды и средней длины очереди необработанных воздействий в зависимости от загруженности системы P, находящегося в диапазоне (0,1);б — зависимость функции штрафа и времени выполнения поведенческого алгоритма от размера обучающей выборки, возрастающего вдоль стрелки от 10^2 до 10^4 записей.

Применение метода динамического программирования становится возможным лишь в том случае, когда модель среды полностью определена, для этого должны быть известны или получены методами анализа BigData матрицы состояний окружающей среды и вознаграждений. Для формирования такой матрицы была использована имитационная модель описанная выше, использование данного подхода позволяет значительно снизить требования к аппаратному обеспечению агента делегируя собственно процесс обучения программно аппаратному комплексу включающемуимитационную модель и алгоритму обучения сервера который может быть представлен в виде нейронный сетей, генетических алгоритмов, обучения с подкреплением и других. Таким образом, становится возможным создание управляющих поведенческих матриц реализующих конечный автомат, управляющих действиями агента и обеспечивающих адекватную реакцию на изменение состояний внешней среды и выполнение поставленной агенту конечной цели.

В случае применения метода Монте-Карло априорная информация о среде может быть неполной или полностью отсутствовать, в таком случае формирование матрицы состояний и реакции среды происходит в режиме реального времени по данным собственных датчиков агента или путем работы с имитационной моделью, отражающей поведение реальной окружающей среды. Технически методы Монте-Карло достаточно просты в применении и могут быть использованы для формирования поведенческих матриц агента в процессе его

фактической эксплуатации. Основным недостатком методов Монте-Карло является высокая вычислительная сложность, делающая проблематичным использования данного метода в реальном времени на оборудовании агента, которое зачастую имеет низкую производительность, также перед исследователем остается проблема решения коллизий в случае несоответствия прогнозируемой модели и фактическим состоянием среды.

Наиболее сложным вопросом представляется обеспечение корреляцииданных полученных на имитационной модели, представляющей с некоторой степени достоверности агента и окружающую среду, с фактическими значениями, полученными на реальном объекте. При построении модели имеют место быть некоторые допущения и погрешности, точная оценка которых позволит оценить степень достоверности полученных результатов, однако это положение вряд ли может быть строго доказано ввиду отсутствия аналитических решений таких систем для произвольных распределений вероятности наступления воздействий в системе агент-среда. Отсюда можно сделать вывод о том, что физическая реализация агента и его совместная работа с поведенческими матрицами, полученными на имитационной модели, позволит наглядно оценить степень достоверности принимаемых управленческих решений.

Библиографический список

1. Шишкин Ю.Е. Имитационная модель кластера для оптимизации процесса мониторинга с целью ликвидации последствий вирусных атак / Ю.Е. Шишкин, С.А. Чорномыз // Оптимизация произв. процессов: сб.науч. тр. – Севастополь: изд-во СевНТУ, 2013. – Вып. 14. – С.174 – 178.

2. Скатков А.В. Модель классификации информационных ситуаций для задач мониторинга многомерных объектов, выполняемых в облачных средах / А.В. Скатков, Е.Н. Мащенко, В.И. Шевченко // Академический форум корпорации ЕМС 2015. Материалы международной учебно-методической конференции 12-17 октября 2015 г.Таганрог. Научное издание. - Таганрог: Изд-во ЮФУ, 2015. - 116.с., С. 61 – 71.

3. Шишкин Ю.Е. Исследование систем управления высокоточными измерениями / Ю.Е. Шишкин, Н.А. Греков // Материалы международной научно-технической конференции молодых ученых, аспирантов и студентов «Интеллектуальные системы, управление и мехатроника - 2015». Севастопольский государственный университет.; науч. ред. Барабанов А.Т. Севастополь, 2015. С. 221 – 225.

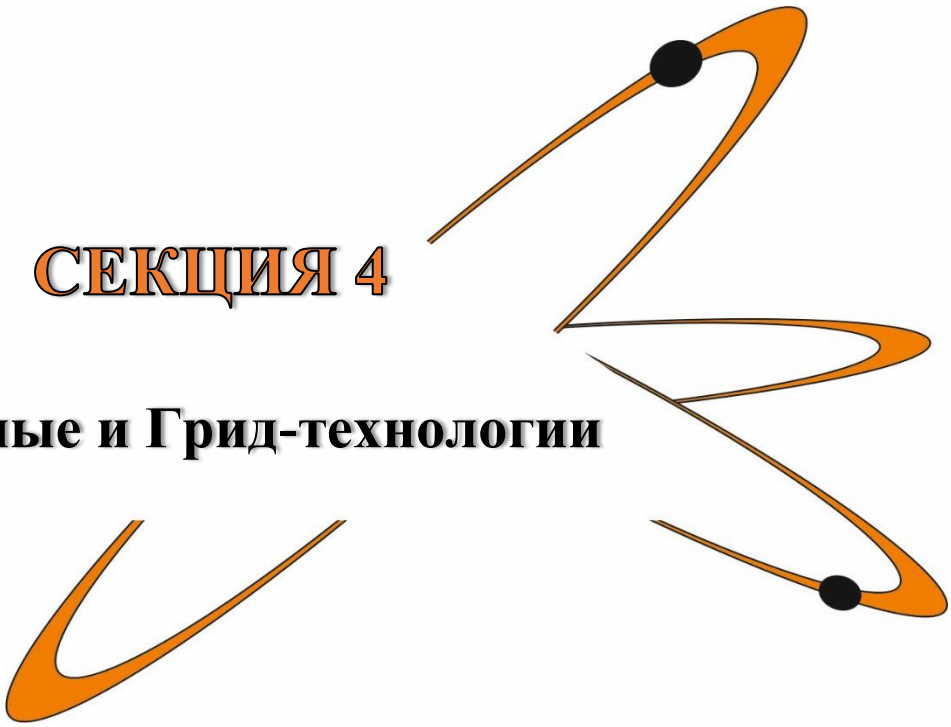
4. Скатков А.В. Концептуальная модель взаимодействия агентов облачного сервиса при проведении крупномасштабных научных исследований / А.В. Скатков, Ю.П. Николаева, Д.В. Сигаев, В.И. Шевченко // В сб. трудов Международной научно-практической конференции «Информационные технологии и информационная безопасность в науке, технике и образовании «ИНФОТЕХ-2015»», Севастополь, 7-11 сентября. 2015 – Севастополь, 2015. г./ Министерство образования и науки Российской Федерации, Севастоп. гос. ун-т; науч. ред. А.В. Скатков – Севастополь: СевГУ, 2015. – 109 с. ISBN 978-5-9906915-6-8, С. 45 – 46.

5. Шишкин Ю.Е. Решение задачи составления расписаний большой размерности с применением технологии больших данных / Ю.Е. Шишкин, А.В. Скатков // «Информационные технологии и информационная безопасность в науке, технике и образовании «ИНФОТЕХ – 2015»» Материалы международной научно-практической конференции. М-во образования и науки Российской Федерации, Севастоп. гос. ун-т; науч. ред. А.В.Скатков. Севастополь, 2015. С. 103 – 105.

6. R.S. Sutton, A.G. Barto. Reinforcement Learning.An Introduction. MIT Press, Cambridge, London, 1998.

7. D. Goldberg; D. Nichols, B. M. Oki, D. Terry (1992). Using collaborative filtering to weave an information tapestry. Communicationsofthe ACM 35 (12): 61—70, 1992.

8. H. Ahonen, O. Heinonen, M. Klemettinen, and A. Verkamo.Applying data mining techniques for descriptive phrase extraction in digital document collections.In Advances in Digital Libraries (ADL'98), Santa Barbara Calfornia, USA April 1998, 1998.



СЕКЦИЯ 4

Облачные и Грид-технологии

УДК 004.94:75
А.В. Гримута, студент 3-го курса
И.В. Смирнов, студент 3-го курса
Научный руководитель: В.И.Шевченко, канд. техн.наук, доцент
Севастопольский государственный университет

БАЗОВАЯ МОДЕЛЬ ВЗАИМОДЕЙСТВИЯ АКТОРОВ ОБЛАЧНОЙ ВЫЧИСЛИТЕЛЬНОЙ СРЕДЫ

Аннотация: Приведен обзор свойств и особенностей основных акторов модели облачной вычислительной среды. Проанализированы особенности различных подходов имитационного моделирования. На основе проведенного анализа построена базовая имитационная модель взаимодействия акторов облачной вычислительной среды с использованием дискретно-событийного подхода.
Ключевые слова: имитационное моделирование, актор, облачная вычислительная среда, дискретно-событийное моделирование.
Annotation: Review the properties and a characteristic of the main actors in the cloud computing environment model was given. Distinctive features of simulation approaches were analyzed. Basic simulation model of the interaction of a cloud computing environment actor was built on the basis of the analysis. The model was constructed using discrete-event approach.
Key words: imitation modeling, actor, cloud computing environment, discrete event modeling.

В научных кругах в последнее время всё большую популярность набирают облачные технологии, которые используются не только для проведения распределенных вычислений в рамках экспериментов, но и как хранилища научных статей, материалов, баз знаний, экспериментальных данных и т.д. Будучи новым направлением, облачные технологии порождают целый ряд научных и технических проблем, связанных с процессами организации взаимодействия всех участников процесса, способами сжатия и хранения данных, обеспечением безопасности и гарантоспособности реализуемой системы, предоставлением требуемого уровня IT-сервисов. Разрешение этих вопросов возможно путем построения реальной системы с последующими её модификациями для определения опытным путём наиболее подходящей под требования пользователей структуры, что ведет к весьма существенным временным и финансовым затратам, а, зачастую, вообще не представляется возможным. Одним из путей разрешения сложившейся ситуации является применение методов математического и имитационного моделирования. В данной работе авторы при исследовании взаимодействий между элементами облачной среды опирались на эталонную архитектуру NIST [1, 2]. В качестве аппарата был выбран метод имитационного моделирования в системе AnyLogic на основе дискретно-событийной модели взаимодействия.

В основе эталонной архитектуры NIST (National Institute of Standards and Technology), лежит пять основополагающих, главных действующих субъектов – акторов (actors), каждый из которых выступает в своей роли (role) и выполняет строго определенные действия (activities) и функции (functions). Полный перечень акторов в концепции NIST приведен в таблице 1. В рамках проводимого исследования на данном этапе было решено рассмотреть взаимодействие только трёх акторов между собой: Облачного Потребителя, Облачного Провайдера и Облачного Брокера, схема взаимодействия которых приведена на рис. 1.

Как уже отмечалось ранее, в качества аппарата моделирования была выбрана отечественная среда AnyLogic от компании «The AnyLogic Company» [3], которая, в отличие от большинства узконаправленных инструментов имитационного моделирования, относится к инструментам общего назначения, позволяет описывать модели любых систем и объединяет в себе три основных подхода к моделированию: системную динамику, агентное и дискретно-событийное моделирование. Коротко перечислим отличительные особенности каждого из подходов.

1. Агентное моделирование

Примером использования агентного моделирования является потребительский рынок, где в условиях сложной, динамичной и конкурентной среды выбор покупателя зависит от многих факторов: сети контактов, индивидуальных особенностей, врожденной активности, внешних влияний и т.д. Такие взаимодействия лучше всего описывает именно агентное

моделирование [4]. Тем не менее, данный подход применим не только для решения задач коммуникативного характера, но и для моделирования ситуаций, связанных с цепями поставок, логистикой, бизнес-процессами, производством.

Таблица 1. Список акторов в эталонной архитектуре NIST

Актор	Определение
Облачный Потребитель (Потребитель) Cloud Consumer	Лицо или организация, поддерживающая бизнес-отношения и использующая услуги *Облачных Провайдеров.*
Облачный Провайдер (Провайдер) Cloud Provider	Лицо, организация или сущность, отвечающая за доступность облачной услуги для *Облачных Потребителей*.
Облачный Аудитор (Аудитор) CloudAuditor	Участник, который может выполняет независимую оценку (assessment) облачных услуг, обслуживания информационных систем, производительности и безопасности реализации облака.
Облачный Брокер (Брокер) Cloud Broker	Сущность, управляющая использованием, производительностью и предоставлением облачных услуг, а также устанавливающая отношения между *Облачными Провайдерами* и *Облачными Потребителями*.
Облачный Оператор Связи (Оператор) Cloud Carrier	Посредник, предоставляющий услуги подключения и транспорт (услуги связи) <доставки> облачных услуг от *Облачных Провайдеров* к *Облачным Потребителям*.

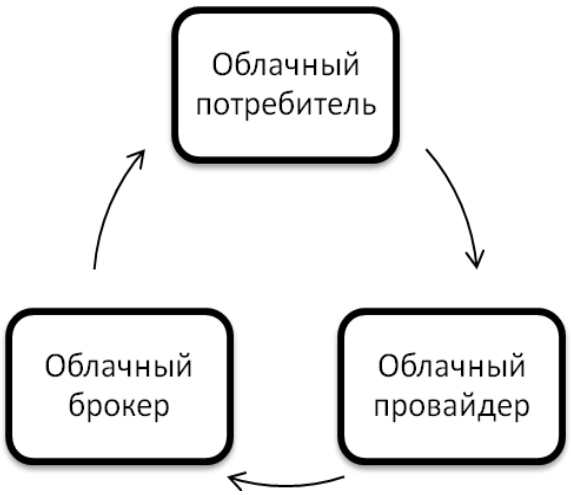


Рис. 1. Схема взаимодействие Облачного Потребителя, Облачного Провайдера и Облачного Брокера

2. Системная динамика

Данный подход имитационного моделирования позволяет посредством своих методов и инструментов понять динамику и структуру сложных систем. В частности, эта концепция применима для создания точных компьютерных моделей сложных систем, которые используются для проектирования более эффективной организации и политики взаимоотношений с данной системой, например, посредством сжатия и замедления пространства и времени для получения возможности изучения принимаемых решений с последующим проектированием тактики и стратегии управления, способствующих достижению наибольшего успеха [5].

3. Дискретно-событийное моделировании

Большинство процессов, происходящих в реальном мире, являются непрерывными во времени. Однако, для их изучения и анализа, бывает удобно отказаться от их непрерывной

природы и рассматривать только некоторые «важные моменты» (или «события») в жизни моделируемой системы, которые важны для исследователя. Такой подход получил название дискретно-событийного моделирования (discrete event modeling), хотя этот термин зачастую используется в более узком смысле для обозначения процессного моделирования, при котором динамика системы представляется операций (прибытие, задержка, захват ресурса, разделение и т.д.) над агентами (клиенты, документы, звонки, пакеты данных, транспортные средства и т.п.) [6].

Поскольку процессы обработки заявок от Пользователя Брокером, выделение Брокеру облачного времени Провайдером, получение Потребителем времени от Провайдера через Брокера и подобные события дискретны во времени, все эти взаимодействия решено было рассматривать в точки зрения концепции дискретно-событийного моделирования.

Были использованы следующие параметры модели:

– time_usis – время, которое нужно пользователю для работы с сервером;

– shtraf – переменная, которая накапливает штраф, за превышение времени пребывания заявки в системе.

Общие характеристики модели приведены в таблице 2.

Таблица 2. Характеристики модели

Структурные характеристики модели	**Значения характеристик**
Число локальных узлов, участвующих в процессе обмена информацией	3
Число одновременно инициируемых запросов	5
Запрашиваемое время на загрузку сервера	От 1 до 9 с
Время задержки у облачного брокера(queue)	От 3 до 4 с

Схема базовой модели взаимодействия акторов облачной вычислительной среды представлена на рисунке 2.

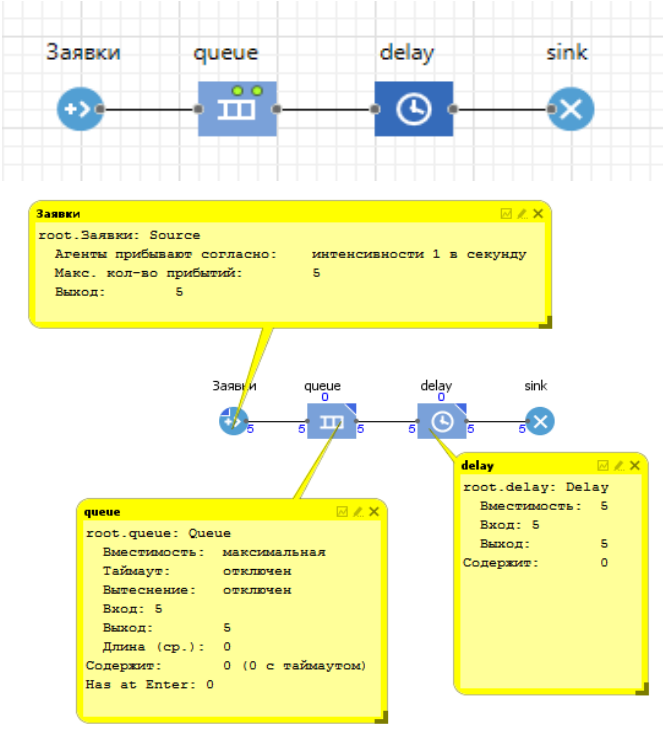


Рис. 2. Разработанная имитационная модели и результаты её работы

Для реализации процесса генерации блока из пяти одновременно поступающих запросов от Пользователя в имитационной модели используется один блок *sourse.* Блок *queue* имитирует задержку на стороне Брокера при распределении запросов Пользователя к ресурсам Провайдера. Для моделирования выполнения работы Провайдера использован блок *delay*. Результаты работы Провайдера фиксируются в системах мониторинга Брокера, этот процесс имитируется блоком *sink.*

Рассмотренная модель позволяет наглядно наблюдать имитацию взаимодействия между акторами: Пользователь – Провайдер. По результатам исследования для рассмотренного примера можно сделать вывод, что данная модель взаимодействия удобна для иллюстрации сценариев работы с Провайдером. Модель может быть дополнена набором целевых функций, отображающих финансовые потоки Пользователей, Провайдеров и Брокеров.

В перспективе дальнейших исследований предусматривается разработка моделей, отражающих сценарии взаимодействия между другими акторами облачной среды. Также планируется разработки моделей на основе системно-динамического подхода. Комплекс имитационных моделей планируется интегрировать в единую систему поддержки принятия решений. Такая система применима, в частности, для исследования процессов работы научных сотрудников с Облачным Хранилищем Данных, позволяющим объединять данные разных вузов и коллективов в единую сеть для ускорения научных исследований и формирования общего информационного поля.

Библиографический список

1. Liu F., Tong J., Mao J. NIST Cloud Computing Reference Architecture. Recommendations of the National Institute of Standards and Technology. Cloud Computing Program Information Technology Laboratory National Institute of Standards and Technology Gaithersburg, MD 20899-8930 September 2011 – Режим доступа: (http://www.nist.gov/customcf/get_pdf.cfm?pub_id=909505) (дата обращения: 17.11.2015).
2. Hogan M., Liu F., Sokol A. NIST Cloud Computing Standards Roadmap. Computer Security Division Information Technology Laboratory National Institute of Standards and Technology Gaithersburg, MD 20899-8930 July 2011– Режим доступа: (http://www.nist.gov/customcf/get_pdf.cfm?pub_id=909024) (дата обращения: 17.11.2015).
3. Anylogic.ru [Электронный ресурс]: Инструмент многоподходного имитационного моделирования / AnyLogic - единственный многоподходный инструмент имитационного моделирования: агентное, дискретно-событийное моделирование и системная динамика. – 2012. – 21 марта. – Режим доступа: http://www.anylogic.ru/ (дата обращения: 14.02.2016).
4. Anylogic.ru [Электронный ресурс]: Агентное моделирование. Инструмент имитационного моделирования AnyLogic/ AnyLogic - единственный многоподходный инструмент имитационного моделирования: агентное, дискретно-событийное моделирование и системная динамика. – Электрон.дан. – 2012. – 21 марта. – Режим доступа: http://www.anylogic.ru/agent-based-modeling (дата обращения: 06.03.2016).
5. Anylogic.ru [Электронный ресурс]: Системная динамика. Инструмент имитационного моделирования AnyLogic / AnyLogic - единственный многоподходный инструмент имитационного моделирования: агентное, дискретно-событийное моделирование и системная динамика. – 2012. – 21 марта. – Режим доступа: http://www.anylogic.ru/system-dynamics (дата обращения: 06.03.2016).
6. Кондратьев М.А. Построение распределенных дискретно-событийных моделей в среде AnyLogic / М.А. Кондратьев//Параллельные вычислительные технологии. – 2009. – №1. – С.542-548.

УДК 681.3
Е.О. Каргаполова, магистрант
Научный руководитель: Ю. Г. Шаталова, доцент, канд. техн. наук
Севастопольский государственный университет

ИССЛЕДОВАНИЕ МОДЕЛИ ВЗАИМОДЕЙСТВИЯ КЛИЕНТОВ И УДАЛЕННОГО СЕРВЕРА БАЗ ДАННЫХ

Аннотация: Анализируются модели организации взаимодействия клиентов и распределенной базы данных. Обосновывается выбор трехуровневой модели. Строится модель выбранной архитектуры в среде моделирования AnyLogic, проводится анализ результатов эксперимента.
Ключевые слова: распределенная база данных, сервер приложений, моделирование.
Annotation: Analyzes the model of interaction of clients and distributed database. The choice of a three-tier model. The model of the chosen architecture in an environment AnyLogic simulation experiment results analysis is carried out.
Key Words: distributed database, application server, modeling.

Архитектура клиент - сервер предполагает сервер баз данных, представляющий собой приложение, осуществляющее комплекс действий по управлению данными - выполнение запросов, хранение и резервное копирование данных, отслеживание ссылочной целостности, проверку прав и привилегий пользователей, ведение журнала транзакций.

Почти все модели организации взаимодействия пользователя с базой данных построены на основе модели «клиент-сервер». То есть предполагается, что каждое такое приложение отличается способом распределения функций ранее приведенных групп обработки данных между, как минимум, двумя частями:

Сервер – это программа, реализующая функции собственно СУБД: определение данных, запись – чтение данных, поддержка схем внешнего, концептуального и внутреннего уровней, диспетчеризация и оптимизация выполнения запросов, защита данных.

Клиент – это различные программы, написанные как пользователями, так и поставщиками СУБД, внешние или «встроенные» по отношению к СУБД. Программа-клиент организована в виде приложения, работающего «поверх» СУБД, и обращающегося для выполнения операций над данными и компонентами СУБД через интерфейс внешнего уровня. [1]

Архитектура «клиент-сервер» определяет общие принципы организации взаимодействия в сети, где имеются серверы, узлы-поставщики некоторых специфичных функций (сервисов) и клиенты, потребители этих функций.

Практические реализации такой архитектуры называются клиент-серверными технологиями. Каждая технология определяет собственные или использует имеющиеся правила взаимодействия между клиентом и сервером, которые называются протоколом обмена (протоколом взаимодействия).

В любой сети, построенной на современных сетевых технологиях, присутствуют элементы клиент-серверного взаимодействия, чаще всего на основе двухзвенной архитектуры.

Двухзвенной эта архитектура называется из-за необходимости распределения трех базовых компонентов между двумя узлами (клиентом и сервером).

Двухзвенная архитектура используется в клиент-серверных системах, где сервер отвечает на клиентские запросы напрямую и в полном объеме, при этом используя только собственные ресурсы. Т.е. сервер не вызывает сторонние сетевые приложения и не обращается к сторонним ресурсам для выполнения какой-либо части запроса.

Еще одна тенденция в клиент-серверных технологиях связана с все большим использованием распределенных вычислений. Они реализуются на основе модели сервера приложений, где сетевое приложение разделено на две и более частей, каждая из которых может выполняться на отдельном компьютере. Выделенные части приложения взаимодействуют друг с другом, обмениваясь сообщениями в заранее согласованном формате. В этом случае двухзвенная клиент-серверная архитектура становится трехзвенной (three-tier, 3-tier).

Как правило, третьим звеном в трехзвенной архитектуре становится сервер приложений, т.е. компоненты распределяются следующим образом (рисунок 1):

1. Представление данных — на стороне клиента.
2. Прикладной компонент — на выделенном сервере приложений (как вариант, выполняющем функции промежуточного ПО).
3. Управление ресурсами — на сервере БД, который и представляет запрашиваемые данные.

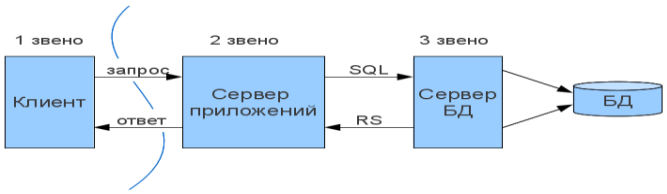


Рис.1. Трехзвенная клиент-серверная архитектура

Трехзвенная архитектура может быть расширена до многозвенной (N-tier, Multi-tier) путем выделения дополнительных серверов, каждый из которых будет представлять собственные сервисы и пользоваться услугами прочих серверов разного уровня. [2]

Система клиент-сервер является наиболее перспективной, так как поддерживает большое число пользователей и сложные приложения, кроме этого она обладает высоким уровнем защиты информации, за счет среды программирования SQL Server и все данные и прикладные средства хранятся централизованно, то есть, сосредоточены в одном месте. [3]

Применительно к системам баз данных архитектура "клиент-сервер" интересна и актуальна главным образом потому, что обеспечивает простое и относительно дешевое решение проблемы коллективного доступа к базам данных в локальной сети. В некотором роде системы баз данных, основанные на архитектуре "клиент-сервер", являются приближением к распределенным системам баз данных, конечно, существенно упрощенным приближением, но зато не требующим решения основного набора проблем действительно распределенных баз данных.[4]

Для исследования была выбрана многоуровневая клиент - серверная архитектура с облачным сервером, представленная на рисунке 2.

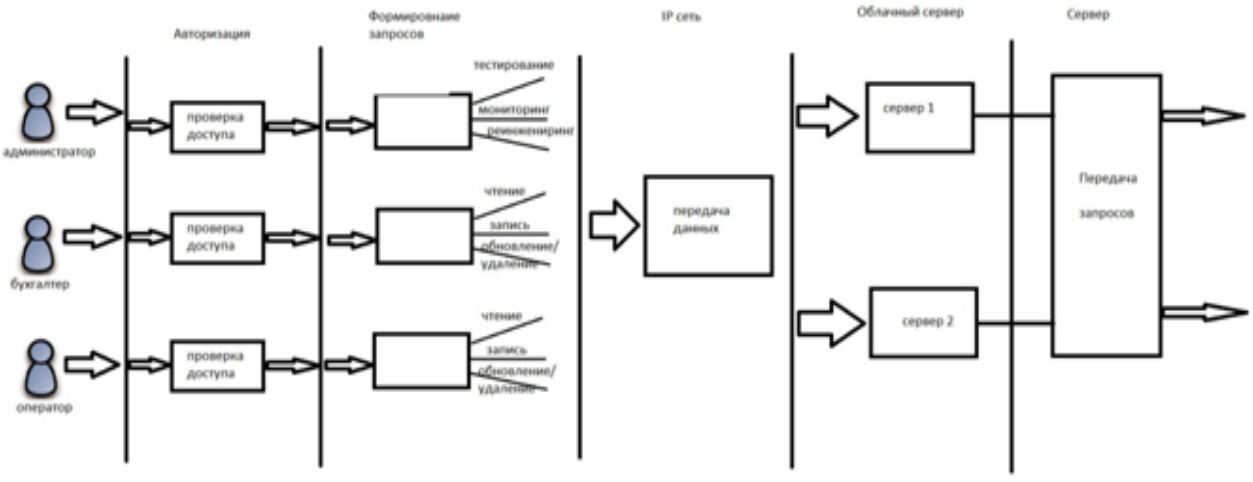


Рис. 2. Клиент – серверная архитектура с взаимодействием облачных серверов

Пусть в системе находятся три типа пользователей (клиентов): администратор, бухгалтер, оператор, посылающие запросы на сервер.

Для начала пользователю необходимо пройти авторизацию, в зависимости от его уровня доступа, затем пользователю будут выданы права для работы в системе.

Передача данных происходит по IP-сети, после чего данные изменяются на сервере, затем происходит выдача результата пользователю.

Построим модель исходной задачи в системе имитационного моделирования AnyLogic. Для этого введем следующие функциональные элементы:

source – генератор заявок;

delay– элемент задержки (обработки).

При помощи среды AnyLogic была разработана модель исследуемой архитектуры, которая приведена на рисунке 3.

.

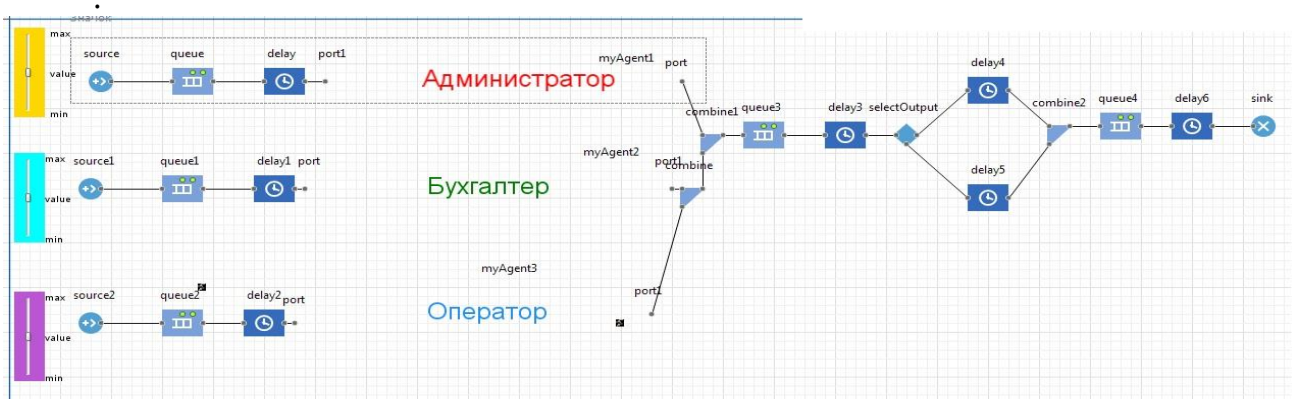


Рис.3. Модель клиент-серверной архитектуры

В модели “sourse”- 3 группы клиентов: администратор, бухгалтер, оператор. Элементы queue queue1, queue2 и delay, delay1, delay2 составляют модель блока авторизации клиентов.

Формирование запросов для клиентов представлено агентами: администратор, бухгалтер, оператор.

Для каждого пользователя назначаются различные приоритеты, если на вход поступает заявка с приоритетом выше заданного, то клиенту выдаются права на чтение или изменение данных.

Элементы combine и combine1 собирают заявки и передают их по IP-сети (блоки queue3 и delay3) на облачные сервера (delay4 delay5). После того, как заявки обработались на сервере, они поступают в FC SAM, после чего происходит ответ пользователю.

Время моделирования 21 600 секунд

В таблице 1 представлены исходные данные для разработанной модели.

Таблица 1 – Исходные данные

Элемент	Действия при выходе
Клиент (source) (source1) (source2)	agent.parameter=(int)(6*Math.random()); agent.parameter=(int)(9*Math.random()); agent.parameter=(int)(8*Math.random());
	Время задержки заявок
delay	exponential(0.06)
delay1	exponential(0.04)
delay2	exponential(0.03)
delay3	exponential(0.01)
delay4	exponential(0.05)
delay5	exponential(0.03)
delay6	exponential(0.02)
	Для агента Администратор:
delay	exponential(0.002)
delay1	exponential(0.03)

delay2	exponential(0.04)
selectOutput	Проверяем условие приоритета входящей заявки: gent.parameter==5
	Для агента Бухгалтер
Delay	exponential(0.002)
Delay1	exponential(0.06)
Delay2	exponential(0.03)
selectOutput	Проверяем условие приоритета входящей заявки: agent.parameter==6
	Для агента Оператор:
Delay	exponential(0.005)
Delay1	exponential(0.03)
Delay2	exponential(0.02)
selectOutput	Проверяем условие приоритета входящей заявки: gent.parameter==5

Бегунками задан массив возможных значений для генерации заявок:
Для администратора: минимальное значение: 0, максимальное: 50
Для бухгалтера: минимальное значение: 1, максимальное: 100
Для оператора: минимальное значение: 0, максимальное: 100

Результаты моделирования представлены на рисунках 4 – 7.

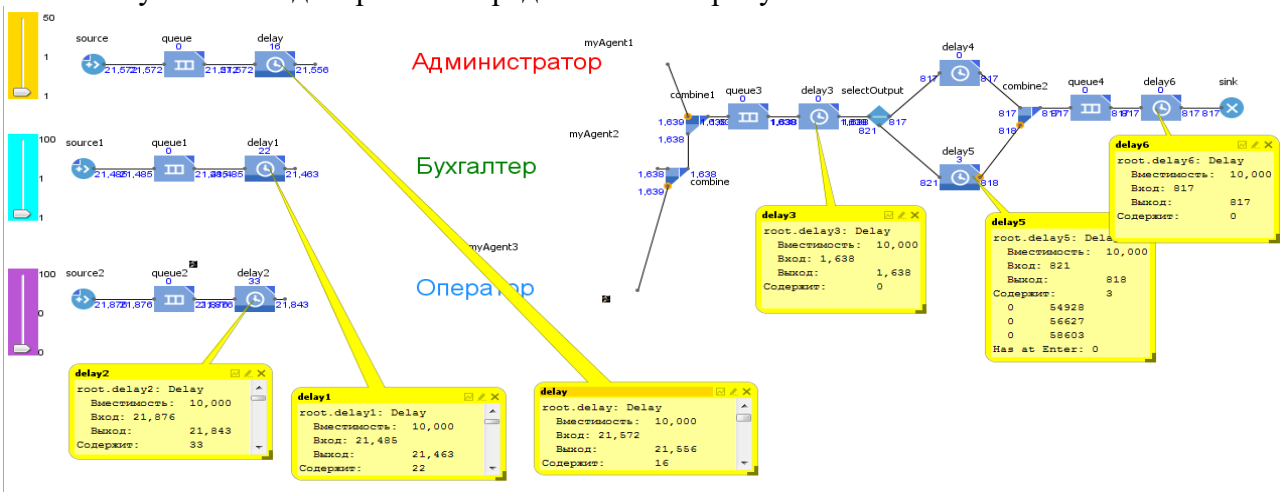


Рис.4. Результат имитационного моделирования

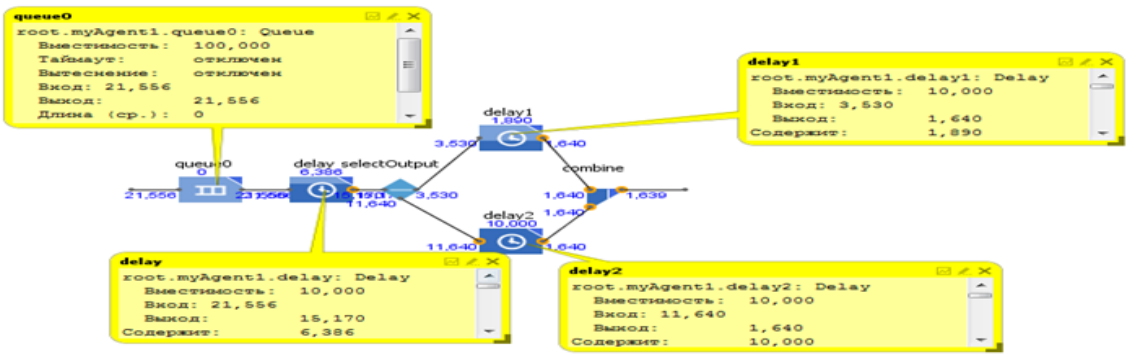


Рис. 5. Результаты моделирования агента- Администратор

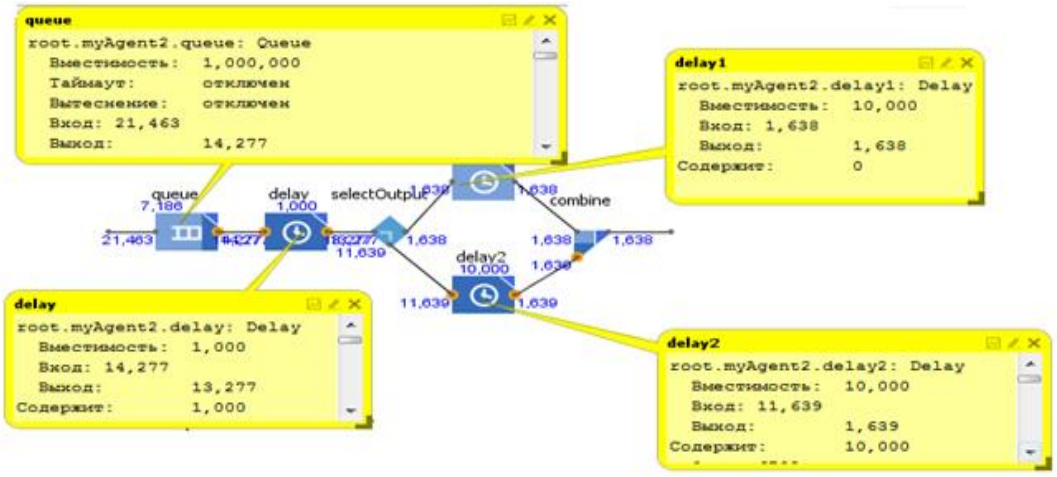


Рис. 6. Результаты моделирования агента Бухгалтер

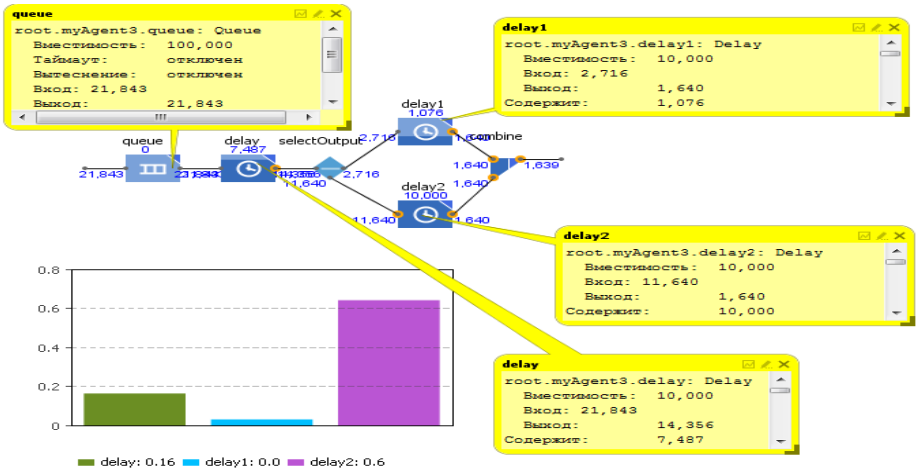


Рис. 7. Результаты моделирования агента Оператор

По результатам работы агентов мы видим, что не все заявки обрабатываются, так как проверяется приоритетность заявки. Если приоритет меньше заданного, то данная заявка не обрабатывается или назначаются ограниченные права клиенту.

На данном этапе научно-исследовательской работы были проанализированы архитектуры взаимодействия клиентов и удаленного облачного сервера.

Библиографический список

1. Алексеев А.С. Информационные ресурсы и технологии начала XXIвека//Связь времен: Сборник.–М.: МГВП КОКС, 2004.–С. 780–794.
2. А.Г. Анатольев. Сетевые технологии/Конспект лекций/ Компоненты сетевого приложения. Клиент-серверное взаимодействие и роли серверов, http://www.4stud.info/networking/lecture5.html (дата обращения 3.03.2016).
3. Попов И.Г., Мамонов С.Г. Информационные системы. М.: Инфра, 2007. - 528 с.
4. С.Д. Кузнецов. Основы современных баз данных/ Информационно-аналитические материалы /Центр Информационных технологий МГУ, http://citforum.ru/database/osbd/contents.htm (дата обращения 27.02.2016).

УДК 004.052
М.С. Лисов, магистрант
Научный руководитель: Д. Ю. Воронин, канд. техн. наук, доц.
Севастопольский государственный университет

АНАЛИЗ РИСКОВ НАРУШЕНИЯ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ ОБЛАЧНОЙ ВЫЧИСЛИТЕЛЬНОЙ СРЕДЫ

Аннотация: В настоящее время стремительно увеличивается потребность в хранении и обработке больших объёмов данных, что провоцирует интенсивное развитие вычислительных сред, использующих облачную парадигму. Однако, использование парадигмы облачных вычислений может иметь как положительные, так и отрицательные эффекты, проявляющиеся, например, в наличии рисков нарушения информационной безопасности облачной вычислительной среды. По результатам анализа этих рисков предложена фасетная классификация.
Ключевые слова: управление рисками, облачная среда, фасетная система классификации, облачный провайдер.
Annotation: At the present time, the need for storing and processing large amounts of data is rapidly growing, that provokes an intensive development of computing environments, which use cloud paradigm. However, the usage of cloud computing can have both positive and negative effects, that appear, for example, in the risks of violating the information security of the cloud computing environment. According to the analysis of these risks the facet classification is proposed.
Key words: risks management, cloud environment, faceted classification model, cloud provider.

В настоящее время «облачные» вычисления — это технология, которая имеет огромную популярность. Использование в облачных системах IaaS и PaaS моделей обслуживания, позволяют ИТ-компаниям получить преимущества в области повышения эффективности, экономии затрат и масштабирования [1]. Однако этим революционным возможностям сопутствует целый ряд новых проблем, благодаря которым традиционные способы обеспечения безопасности стали неэффективны. Парадокс кроется в самой основе новой концепции вычислений. С одной стороны, «облако» — это упрощенная вычислительная среда с оплатой за фактически потребляемые ресурсы. С другой стороны, возникают многочисленные новые заботы и дополнительные риски по обеспечению информационной безопасности.

В общем случае все риски информационной безопасности при использовании облачных вычислений, можно условно разделить на следующие категории:

- Организационные.
- Технические.
- Потеря доступности.

Организационные риски

А) Привязка к поставщику [4]. Серьезным риском является привязка к определенному поставщику облачных услуг. Поскольку расходы по миграции из локальной среды в облако значительны, то, в случае, если поставщик перестанет удовлетворять потребности компании по каким-либо критериям (увеличится плата за использование, на рынке появится более хороший и дешевый сервис и т.д.), то сменить его будет достаточно проблематично. Денежные и временные затраты могут быть колоссальными. Поэтому необходимо ответственно отнестись к выбору поставщика облачных услуг.

Б) Банкротство или поглощение провайдера [4]. Если по каким-либо причинам провайдер обанкротился или был поглощён более успешной компанией, то есть шанс, что будет остановлено предоставление сервиса или изменение предоставления услуг. Возможные контрмеры – тщательный подход к выбору провайдера, работа с несколькими провайдерами, наличие плана действия на смену провайдера.

В) Вопрос анонимности [1]. Вся информация, которая необходима для получения доступа, ограничивается лишь номером кредитной карты и незначительными введенными данными. Сведения о пользователе, получившему доступ, в таком случае сводятся к адресу электронной почты и IP-адресу. В этих условиях велика вероятность использования злоумышленником платформы облачных вычислений в корыстных целях. Возможными контрмерами, чтобы уменьшить этот риск является введение более строгих правил начальной регистрации, а также контроль со стороны поставщика за действиями пользователей.

Г) Вредоносные инсайдеры компании [2]. Сотрудникам организаций, использующих облачные вычисления, не обязательно иметь глубокие знания технических деталей того, как оказываются услуги, для того, чтобы использовать данные пользователей в корыстных целях (изменение и/или кража). Требуется плотный контроль со стороны организации за своими корпоративными сотрудниками.

Технические риски.

Технические риски и способы их предотвращения классифицируем и представим с помощью таблицы 1.

Таблица 1 – Технические риски информационной безопасности

1. Основные точки возникновения технических рисков	1.1 Трудности при перемещении обычных серверов в вычислительное облако
	1.2 Динамичность виртуальных машин
	1.3 Уязвимости внутри виртуальной среды
	1.4 Защита периметра и разграничение сети
2. Решения по защите от угроз безопасности от компании CSA [3]	2.1 Сохранность данных. Шифрование
	2.2 Защита данных при передаче
	2.3 Аутентификация
	2.4 Изоляция пользователей

Потеря доступности.

Последняя угроза, которая характерна только для облаков — это экономический эффект от DDoS-атаки. Она является обратной стороной преимущества облачных вычислений – плати только за фактическое потребление [1]. При реализации данной атаки колоссально возрастает объем исходящего интернет-трафика из-за увеличения количества запросов к серверу клиента, и пользователь теряет доступ к серверу. В итоге клиент будет обязан полностью оплатить траффик.

Удобнее и нагляднее, представить результаты анализа в виде фасетной классификации, приведенной на рисунке 1. Фасетной называется система классификации, основой которой является деление объектов или понятий на независимые группы – фасеты, каждый из которых характеризует объекты или понятия в одном аспекте, по одному признаку [5].

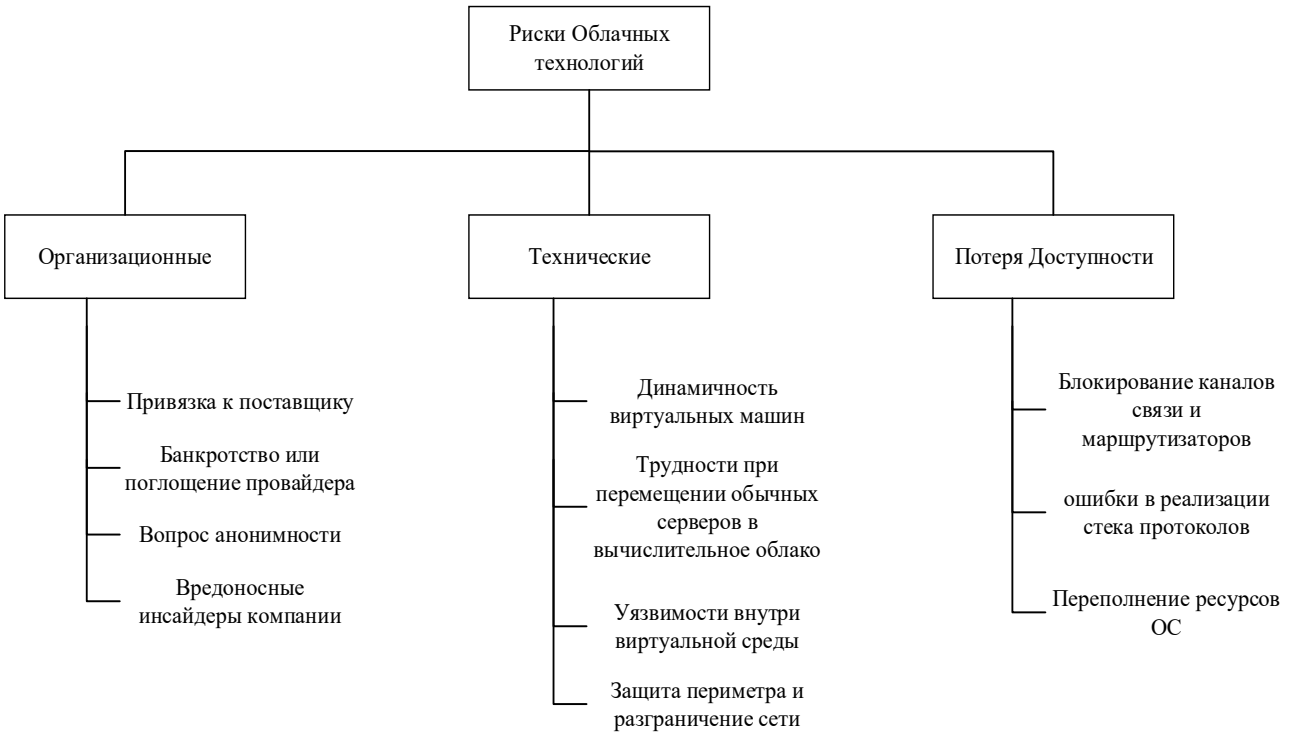


Рис. 1. Фасетная классификация рисков нарушения информационной безопасности облачной вычислительной среды

Технология облачных вычислений обладает множеством преимуществ, поэтому интерес к данной концепции постоянно увеличивается. Существенно повысить безопасность и сократить потери данных могут следующие контрмеры: обязательное применение шифрования к перемещаемым и хранимым данным, централизованное управление идентификационной информацией, а также контроль данных в виртуальных хранилищах. В результате была построена фасетная классификация рисков облачных технологий. Она отличается от иерархической более наглядным визуальным представлением данных, а также является весьма гибкой и многоцелевой.

Библиографический список

1. Малюк А.А., Перспективы развития «облачных» технологий. Информационная безопасность и защита персональных данных в «облачной» среде / А.А. Малюк // Вестник Национального исследовательского ядерного университета «МИФИ», 2013. №С. 120-124.

2. Cloud Security Alliance (CSA) [Электронный ресурс]. – Электрон. текстовые дан. – Режим доступа: https://cloudsecurityalliance.org/, свободный.

3. Cloud Computing Security Risk Assessment [Электронный ресурс]: – Электрон. текстовые дан. – European Network and Information Security Agency, Nov. 2009. Режим доступа: http://www.enisa.europa.eu/activities/risk-management/files, свободный.

4. The NIST Definition of Cloud Computing [Электронный ресурс]: многопредмет. науч. журн. / – USA Gaithersburg National Institute of Standards and Technology, 2011. – режим доступа к журн.: http://csrc.nist.gov/nistpubs

5. Фасетная система классификации [Электронный ресурс]. – Электрон. текстовые дан. – Режим доступа: http://www.in-nov.ru/do/inf_kult/gl252.html, свободный.

УДК 004.942
В.Д. Полетаева, студентка 3-го курса
Научный руководитель: Д.Ю. Воронин, канд. техн. наук, доц.
Севастопольский государственный университет

МОДЕЛИ АНАЛИЗА РИСКОВ ОБЛАЧНОЙ ВЫЧИСЛИТЕЛЬНОЙ СРЕДЫ

Аннотация: в данной статье рассматриваются различные риски характерные для облачных сред. Анализируется особенности реализации облачных технологий, приводятся модели оценки рисков в них.
Ключевые слова: облачные технологии, облачная среда, риск, имитационное моделирование
Annotation: The paper deals with different risks types which are the most typical for cloud environment. They structure of cloud technologies is analyzed, and model for examine risks in the cloud computing environment
Key words: cloud technologies, cloud environment, risk, simulation

Технологии облачных вычислений является своеобразным ответом на современные тенденции развития IT-области, которые характеризуются следующими особенностями: увеличением объемов и интенсивностей информационных потоков, неравномерное распределение вычислительных мощностей, несбалансированность размещения вычислительных ресурсов, необходимость создания развитых IT-инфраструктур даже для корпоративных клиентов среднего масштаба. Несмотря на существенные преимущества облачных вычислительных сред (ОВС), появляется острая необходимость в разработке новых подходов к решению задач эффективного анализа и управления рисками. Большинство из них связано с нарушением функциональных и нефункциональных требований к вычислительным процессам, происходящим в ОВС. В частности, большинство исследований [1, 2] посвящены обсуждению вопросов, связанных с управлением рисками базовых типов (технический, локационный, сегрегационный, ресурсный и социальный). Их описание приведено в табл. 1.

Таблица 1 – Типы облачных рисков

Тип риска	**Причина возникновения**	**Описание**	**Последствие**
Технический	Высокая виртуализация среды скрывает от пользователей реальные физические устройства.	Отказ аппаратуры, ошибка в физической составляющей облака	Потеря данных без возможности восстановления
Локационный	Облачные технологии создаются путем объединения в одно целых различных физических ресурсов	Невозможность отследить физическое расположение данных в облаке	Конфиденциальная информация может попасть в руки конкурирующих фирм
Сегрегационный		Все данные в облаке шифруются, и данные двух разных пользователей могут располагаться рядом	Неверная интерпретация данных, нарушение безопасности нескольких пользователей, потеря данных
Ресурсный	Принцип работы «вычислительная мощность по требованию» или «память по требованию»	Невозможность удовлетворить требования клиента по ресурсам при резком возрастании нагрузки на облако	Потеря данных, нарушения директивного срока задачи
Социальный	Преднамеренный перехват данных третьими лицами	Невозможность обеспечить целостность и конфиденциальность данных клиента.	Конфиденциальная информация может попасть в руки конкурирующих фирм, данные могут быть запорчены, результаты вычислений неверны.

Данные риски могут привести к неизбежной потере безопасности и конфиденциальности данных пользователя, но и к отрицательной характеристикам поставщика облачных серверов. В связи с этим, разработка механизма предсказывания,

предупреждения появления рисков и минимизации полученного ущерба – важная задача как для пользователя облачных услуг, так и для брокера

Целью синтеза разрабатываемых моделей анализа рисков является оценка вероятности появления неблагоприятных событий и предупреждение их негативных последствий. Таким образом, необходимо обеспечить выполнение следующих функций:

1. Преобразование эмпирической информации во внутренние структуры модели.
2. Прогнозирование неблагоприятных событий на основе информации о работе пользователя в облачной среде.
3. Оценка ущерба, основанная на данных о важности поставленных задач, их конфиденциальности и сложности.
4. Реализация эргономичного, интуитивно понятного интерфейса для взаимодействия широким кругом лиц и для предоставления актуальной информации, необходимой для принятия взвешенных аргументированных решений по противодействию рискам.

Для реализации необходимых функций предлагаются модели M1 и M2, их структура в виде черного ящика представлена на рисунках 1 и 2.

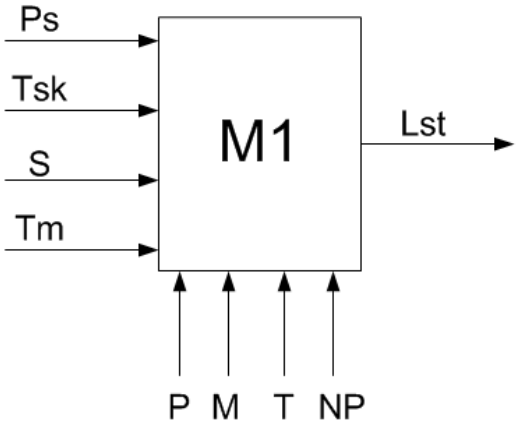


Рис. 1.Ммодель M1.

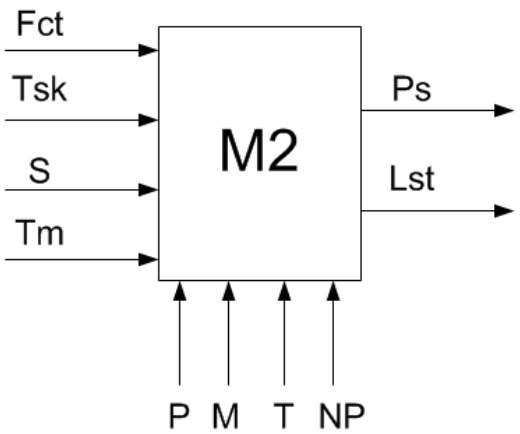


Рис.2. Модель M2

Используются следующие обозначения:

Ps – вероятность появления неблагоприятного сибытия;

множество TSK – параметрические характеристики задачи (количество подзадач, требуемая мощность, требуемая память, технические особенности);

Tm – директивный срок задачи;

S – сумма контракта между пользователем и брокером;

Fct – множество факторов, которые способствуют появлению рисков (сложность задачи, последовательность выполнения подзадач и т.д.);

Tr – множество, описывающее требования пользователя по расположению данных при решении задач или их хранении;

P – максимальная предоставляемая мощность всех физических вычислительных ресурсов брокера;

M – максимальная предоставляемая память всех вычислительных ресурсов брокера;

T – предположительное время работы над подзадачей;

NP – множество, описывающее характеристики провайдеров;

Lst – предполагаемый ущерб для пользователя.

Предлагаемые модели могут быть использованы в ОВС, обобщенная структура которой представлена на рисунке 3. Брокер покупает вычислительные ресурсы провайдеров. При поступлении запроса на решение задачи брокер формирует виртуальную машину (VM), которая полностью удовлетворяет предъявленным требованиям клиента. Задача клиента делится на множество подзадач, которые с помощью специального ПО распределяются по выделенным ресурсам и обрабатываются. Вычислительные ресурсы наращиваются по

требованию, т.е. динамически. Результаты решения задачи отправляются обратно клиенту. Потребитель облачных технологий платит за использование облако по установленному договором тарифу. Также данный договор четко регламентирует права и обязанности брокера, а также качество сервиса и условия конфиденциальности.

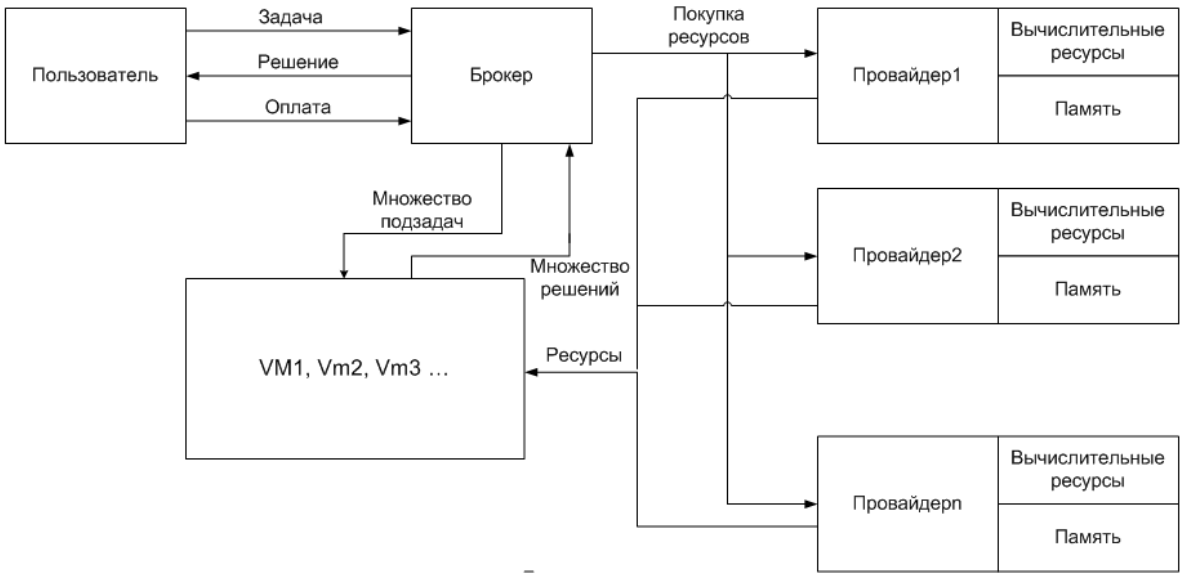


Рис.3. Общая модель работы облака

Сценарий использования разработанного комплекса состоит в следующем:

1. Потребитель предоставляет сведения о минимальном и максимальном требовании памяти, вычислительной мощности для группы задач. По выбору, можно определить сложности задач, возможность к декомпозиции задач на более простые. Обязательно задается директивный срок выполнения задачи.

2. Потребитель должен указать характеристики облачной системы: её вычислительную мощность, память, среднее время работы над подзадачей.

3. Модели функционируют по следующему принципу: генерируется некоторое число задач с заданными потребителем параметрами (минимальные и максимальные значения). Каждая сгенерированная заявка имитирует свой жизненный цикл внутри облака: попадает в блок, который моделирует провайдера, после этого задача разбивается на указанное число подзадач, попадает в блоки, которые моделируют виртуальные машины, и после удачного выполнения выходит из облака. В каждом из блоков задача может подвергнуться воздействию одной из вышеописанных неблагоприятных ситуаций. Если при выходе из блока корректность задачи была потеряна, то указываются потери, которые были повлечены этим событием.

4. Результаты работы модели – расчёт предположительных потерь клиента (модель М1) и определение вероятностей возникновения неблагоприятных событий (М2).

Предложенные имитационные модели могут быть использованы потребителями для минимизации ущерба от возникновения неблагоприятных ситуаций, а также при принятии решении о допустимости развертывания проекта в облачной системе с точки зрения риска.

Библиографический список

1. Nick Antonopoulos, • Lee Gillam. Cloud Computing Principles, Systems and Applications// Springer-Verlag London Limited 2010. – 380 с.

2. Sara Angeles.8 Reasons to Fear Cloud Computing [Электронный ресурс] // URL: http://www.businessnewsdaily.com/5215-dangers-cloud-computing.html (дата обращения: 20.03.2016)

УДК 004.05
И.И. Сапрыкин, студент 4-го курса
Научный руководитель: Ю. Л. Явкун, старший преподователь
Севастопольский государственный университет

СРАВНИТЕЛЬНЫЙ АНАЛИЗ PAAS ДЛЯ РАЗРАБОТКИ WEB ПРИЛОЖЕНИЙ

Аннотация: С развитием облачных технологий, повышением доступности и простоты их использования, все большее количество компаний предлагают свои PaaS(Platform as a service)-проекты для разработчиков. Так как все предлагаемые PaaS имеют различные характеристики, возникает вопрос выбора конкретной платформы. В данной работе проводится сравнительный анализ наиболее известных на данных момент предложений PaaS для разработки web-ориентированных Java приложений, выясняются их преимущества и недостатки.
Ключевые слова: PaaS, облачные вычисления, web-разработка, Google App Engine, OpenShift Online, Heroku Cloud Application Platform, AppFog, Windows Azure Cloud Services.
Annotation: With the development of cloud computing, increasing availability and ease of use, more and more companies are offering their PaaS (Platform as a service) -solutions for developers. Since all of the proposed PaaS have different characteristics, there is a question of choosing a particular platform. In this study has been done a comparative analysis of the most well-known PaaS offerings for the development of Java web applications, were found out their advantages and disadvantages.
Key words: PaaS, cloud computing, web development, Google App Engine, OpenShift Online, Heroku Cloud Application Platform, AppFog, Windows Azure Cloud Services.

Для начала определим, что такое Platform as a Service (PaaS, «платформа как услуга»). PaaS – это сервисные платформы услуг облачных вычислений, которые позволяют разработчикам запускать и управлять своими веб-приложениями без необходимости испытывать сложности создания и поддержания инфраструктуры. Инфраструктура PaaS включает в себя: операционные системы, системы управления базами данных(СУБД), связующее программное обеспечение, средства разработки и тестирования, размещённые у облачного провайдера. Бурное развитие PaaS началось с 2008 года и продолжается до сих пор. За первую половину 2015 года компании, предоставляющие PaaS услуги заработали более 1,2 миллиарда долларов [1], рис.1

Рис.1. Доход PaaS компаний за первую половину 2015 года

Это показывает, что рынок предоставления услуг облачных вычислений для разработчиков обширен, на нем существует множество «игроков», конкуренция на нем велика, и каждая компания старается привлечь разработчиков перейти на использование именно ее платформы.

Был проведен сравнительный анализ пяти наиболее известных и популярных PaaS для разработки web-ориентированных приложений [2]. Исследовались предложения таких

компаний как: Microsoft (Azure Web Sites), Salesforce (Heroku), CenturyLink (AppFog), Google (Google App Engine), Red Hat (OpenShift). Последовательно опишем каждое из них.

Windows Azure Cloud Services предлагаую не только PaaS, но и инфраструктуру как услугу (IaaS). При этом исследовательская и консалтинговая компания Gartner, специализирующаяся на рынках информационных технологий, в качестве лидера отрасли как для IaaS и PaaS выделяет именно Azure. Это мощное сочетание управляемых и неуправляемых напрямую сервисов позволяет создавать, развертывать и управлять приложениями так, для достижения непревзойденной производительности. Azure поддерживает разные операционные системы, языкы, инструменты разработки и фреймворки.

Особенности:

- Предоставление виртуальных машин с операционной системой Windows или Linux и приложений в течение нескольких минут.
- Создание и развертывание широкого спектра современных приложений для Android, iOS, и Windows, которые получачат все преиммущества облачных технологий.
- Автоматическое масштабирование вверх и вниз, для удовлетворения любых потребности.
- Azure предоставляет SQL и NoSQL базы данных. Позволяет использовать HDInsight для построения кластеров Hadoop для анализа данных.

Heroku предоставляет абстрактные вычислительные среды, называемые dynos. Эти среды существуют в двух вариантах: веб-dynos (которые отвечают на HTTP-запросы) и рабочие-dynos (для решения вычислительных задач). Heroku лучше всего работает с приложениями, созданными с использованием методологии двенадцати факторного приложения [2]. Сторонние приложения также доступны как услуги в рамках платформы Heroku.

Особенности:

- После отправки исходного кода приложений, Heroku подготавливает его к исполнению путем извлечения необходимых зависимостей, характерных для языков и фреймворков, которые вы используете в своем приложении.
- По умолчанию, Heroku поддерживает такие язвки, как: Ruby, Node.js, Python, Java и PHP, но можно работать и с другим языком, с помощью настраиваемого пользователем, так называемого buildpack-а.
- Heroku позволяет конфигурировать инфраструктуру отдельно от исходного кода для большей безопасности и переносимости.

AppFog является многоязычной, мульти-функциональной PaaS, что делает его хорошим вариантом для создания нескольких частных облаков. Он поддерживает Java, Ruby, PHP, Python, Node, Scala и Erlang и предлагает MySQL, PostgreSQL, Redis и RabbitMQ наряду со сторонними дополнениями. AppFog базируется на открытой платформе Cloud Foundry и поддерживает Git, SVN и Mercurial для управления кодом.

Особенности:

- Использование HTTP ускорителя Varnish и кэширование кодов операций запуска приложений уменьшает нагрузку на сервер и дает повышенную скорость работы.
- Конфигурация серверов, файерволов, безопасности и фреймворков происходит в автоматическом режиме.
- Совместимость с системами управления версиями: Git, SVN и Mercurial.
- Работает на различных центрах обработки данных в разных регионах.
- Технологии веб-приложений включают в себя PHP, Node, Ruby, Python и Java.
- Простое добавлениеразличных сервисов для приложений, включая MySQL, PostgreSQL, Redis и RabbitMQ.

Google App Engine – это PaaS, которая позволяет создавать и запускать приложения на инфраструктуре Google. Приложения App Engine легко написать, они просты в обслуживании и легко масштабируемы по мере изменения потребностей трафика и хранения данных. App Engine не предоставляет возможности настройки серверов - достаточно просто загрузить приложение, и оно готово к работе.

Особенности:

- Возможность создать и запустить приложение в облаке моментально.
- Поддержка языков Python, Java, PHP, и Go. Делается упор на использование популярных фреймворков для данных языков.
- Приложения запускаются в безопасной среде (sandbox), позволяя App Engine распределять запросы между несколькими серверами для удовлетворения потребностей трафика.
- Каждое приложение работает в пределах своей собственной безопасной, надежной среды, которая не зависит от аппаратного обеспечения, операционной системы или физического расположения сервера.

Red Hat OpenShift является наиболее настраиваемой PaaS и предлагает широкий спектр языков, баз данных и компонентов. Предлагается в трех формах:

1. OpenShift Online (облачный хостинг сервис).
2. OpenShift Enterprise (частная PaaS, которая работает в центре обработки данных заказчика).
3. OpenShift Origin (хостинг платформа для приложений с открытым исходным кодом).

OpenShift автоматизирует задачи системного администрирования, такие как виртуальное выделение ресурсов сервера, конфигурирование и масштабирование, поддерживает Git репозиториеи для управления кодом.

Особенности:

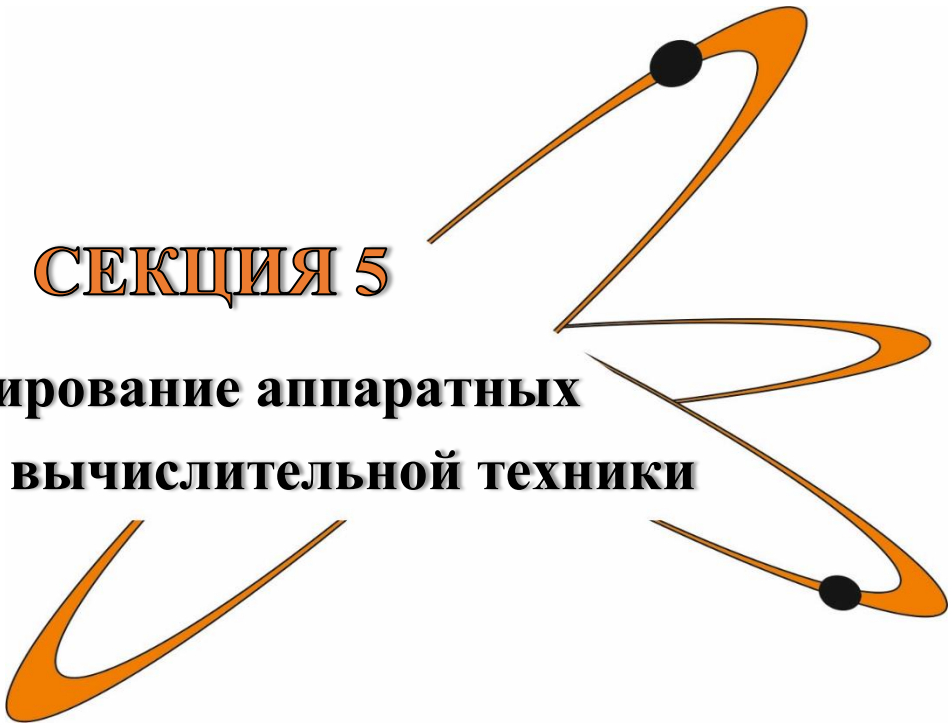
OpenShift Online

- Сокращает время, необходимое для создания и развертывания приложений, позволяя вам сосредоточиться на коде, а не предоставлении инфраструктуры и управления.
- Включает поддержку широкого спектра языковых сред выполнения и СУБД, включая Java EE6, Ruby, PHP, Python, Perl, MongoDB, MySQL и PostgreSQL.
- Использует платформу с открытым исходным кодом на основе стандартизированных компонентов для обеспечения переносимости приложений.
- Интегрированные средства разработки и интуитивно понятный интерфейс позволяют быстро приступить к работе.

Основываясь на вышеописанных характеристиках можно сделать вывод, что незьзя однозначно определить какая из описанных PaaS лучше, а какая хуже. Однако, можно отметить, что некоторые платформы лучше подходят для быстрой разработки приложений, когда не требуется тонкая настройка инфраструктуры (Google App Engine, Heroku, OpenShift Online), некоторые (Windows Azure Cloud Services, AppFog, OpenShift Online) позволяют добиться лучшей производительности и соответствия жестким требованиям после детального изучения возможностей платформы.

Библиографический список

1. Platform-as-a-Service revenue in first half of 2015 (in million U.S. dollars), by vendor. Режим доступа:http://www.statista.com/statistics/478128/paas-vendor-revenue-ranking-worldwide/
2. 12factor.net. Режим доступа: http://12factor.net.



СЕКЦИЯ 5

Проектирование аппаратных средств вычислительной техники

УДК 681.3
А.И. Кононенко, магистрант
Научный руководитель: Н.Е. Сапожников, д.т.н., профессор
Севастопольский государственный университет

ИССЛЕДОВАНИЕ МЕТОДОВ ПОВЫШЕНИЯ ТОЧНОСТИ И БЫСТРОДЕЙСТВИЯ ВЕРОЯТНОСТНОГО МНОЖИТЕЛЬНОГО УСТРОЙСТВА

Аннотация:данная работа посвящена исследованию методов повышения точности и быстродействия вероятностного множительного устройства
Ключевые слова:вероятностная форма представления данных, быстродействие, вероятностное множительное устройство, точность
Annotation:given work is devoted to the study of methods of increasing the accuracy and performance of probabilistic multiplier device
Keywords:probabilistic form of data presentation, performance, probabilistic multiplier device, increasing

Введение

В общем виде суть стохастического или вероятностного преобразования заключается в том, что любому значению преобразуемой величины можно привести в соответствие некоторую вероятность[1].

В настоящее время вероятностную форму представления данных применяют лишь в узконаправленной специализированной области, но она также допустимаи к другим областям человеческой деятельности, так как имеет ряд преимуществ над обычными цифровыми устройствами. Данные преимущества заключаются в малом аппаратном объёме, в способности функционировать в масштабе реального времени, а также – в повышенной помехозащищённости [2].

В связи с необходимостьюрасширить область применения вероятностных устройств, нужно достичь максимального повышения точности и быстродействияэтих устройств.

Для достижения данной цели нужно рассмотреть основные вычислительные узлы вероятностных систем, а также улучшить показатели точности и быстродействия.

Постановка задачи

Целью данной работы является исследование методов точности и быстродействия вероятностного множительного устройства. Данное исследование приведёт к созданию усовершенствованного устройства, которое возможно улучшит спектр применения данного устройства.

Решение задачи

Для того чтобы исследовать методы повышения точности и быстродействия вероятностного множительного устройства, рассмотрим вероятностную форму представления данных.

Вероятностная форма представления данных является двоичной непозиционной формой представления данных, которая говорит о том, что значение числа определяется не положением единиц, а их количеством. То есть, чем больше единиц в числе, тем больше его вес.

В виду того что нужно перейти от «числа» к «вероятности», т.е. выполнить преобразование в соответствии с правилом:

$$\begin{cases} 1, \text{ при } x > R_i \\ 0, \text{ при } x \le R_i \end{cases}, \qquad (1)$$

где x – преобразуемое число; R_i – i-тое значение вспомогательного случайного сигнала; $i = \overline{1, N}$ – количество статических испытаний; y_i – i-тое значение вероятностного отображения величины x.

Если вспомогательный случайный сигнал подчиняется равномерному закону распределения, то оценка математического ожидания вычисляется по формуле (2) и в единичном диапазоне входных данных является оценкой входной величины:

$$M^*[y] = \frac{\sum y_i}{N}. \qquad (2)$$

Для произвольного диапазона входных величин обратное преобразование будет выполняться по формуле (3) [1]:

$$X^* = \frac{\sum y_i}{N} \cdot X_{\max}. \qquad (3)$$

Исследование методов повышения точности и быстродействия вероятностного множительного устройства

Повышение точности вероятностного множительного устройства можно достичь с помощью таких методов, как: 1) увеличение количества испытаний данного устройства; 2) использование поразрядного умножения.

Согласно источнику [2] получим, что количество испытанийбудет зависеть от заданной погрешности преобразования, а такжепри равномерном распределении R_i меньше зависит от значения преобразуемого параметра χ.

При втором методе повышения точности будем использовать поразрядное умножение. Данная схема приведена на рисунке 1.

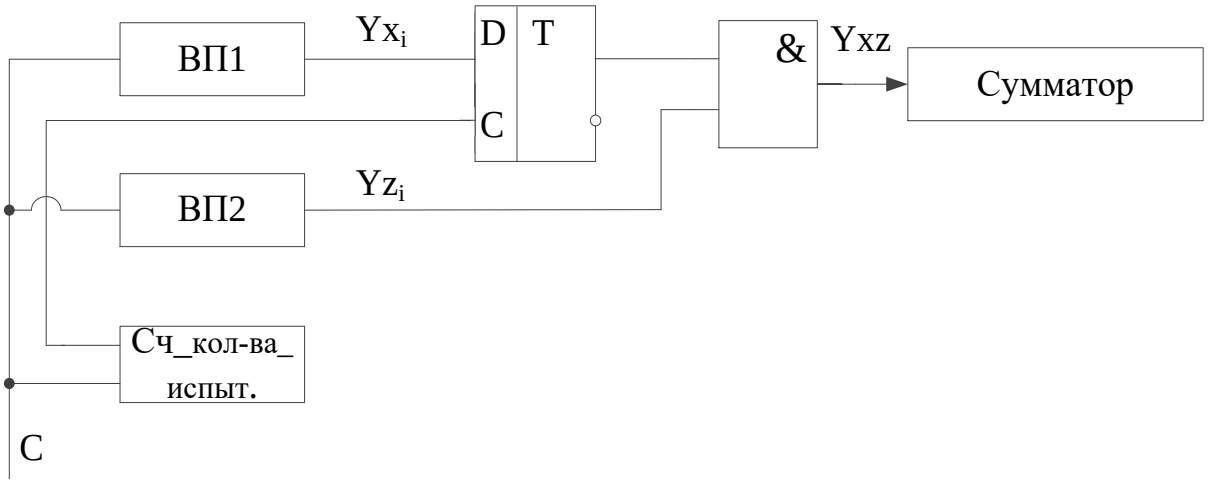


Рис.1. Структурно-функциональная схема поразрядного умножения

Для данной схемы были использованы такие элементы, как:

1. Вероятностный преобразователь – ВП1, ВП2;
2. Счётчик количества испытаний – Сч_кол-ва_испыт.;
3. D-триггер;
4. Логический элемент «И» (логическое умножение);
5. Сумматор.

Повышение быстродействия вероятностного множительного устройства достигается с помощью применения параллельных вычислений над вероятностно представленными данными, а также использования псевдовероятностного преобразователя.

В данном случае в роли параллельного вычислительного устройства выступает множительное устройство, которое увеличивает быстродействие. Множительное устройство представлено на рисунке 2.

Для того, что построить данную схему потребуется:

1. Вероятностный преобразователь – ВП1, ВП2;
2. Логический элемент «И» (логическое умножение);
3. Сумматор;
4. Сумматор результата – Сумматор_рез.

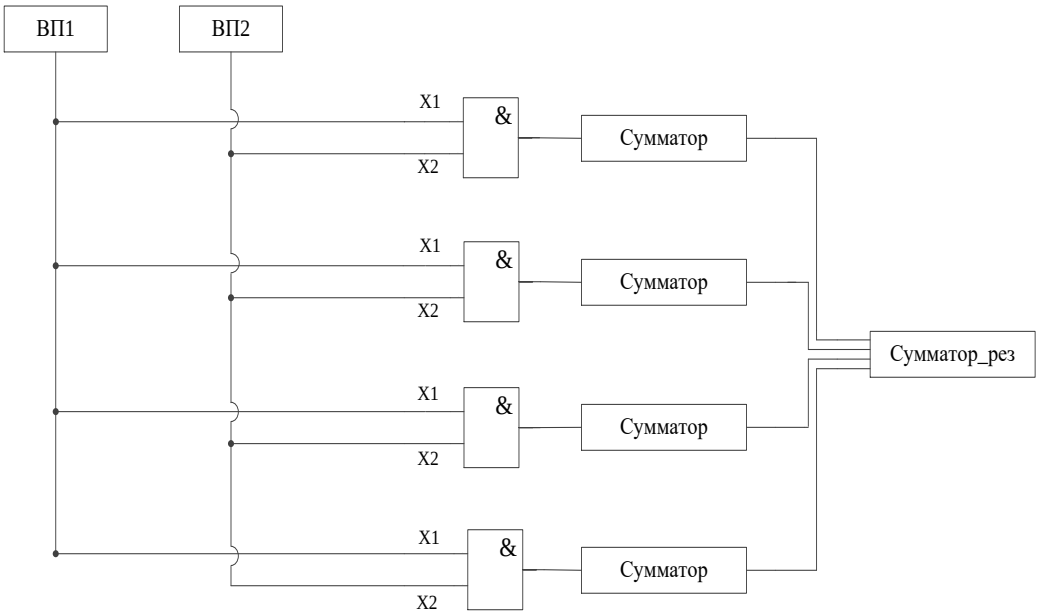


Рис.2. Структурно-функциональная схема параллельного вычислительного устройства

В результате получили, что быстродействие данного устройства увеличилось в 4 раза.

Таким образом, рассмотренные методы позволяют перейти к моделированию усовершенствованного вероятностного множительного устройства.

Сначала приведём обычную структурно-функциональную схему вероятностного множительного устройства, которая представлена на рис. 3.

Для того чтобы построить вероятностный умножитель для данных в формате с плавающей точкой, понадобятся следующие элементы:

1. Входные регистры – Рг;
2. Блок «Умножитель»;
3. Регистр результата – $Рг_р$.

Операция умножения производится по тому же принципу что и в десятичной системе счисления.Поэтому в Рг1 заносятся мантисса и порядок первого множителя, а в Рг2, аналогично, – второго. Далее данные поступают по шинам на блок умножения. После этого производится умножениеи формируется результат.

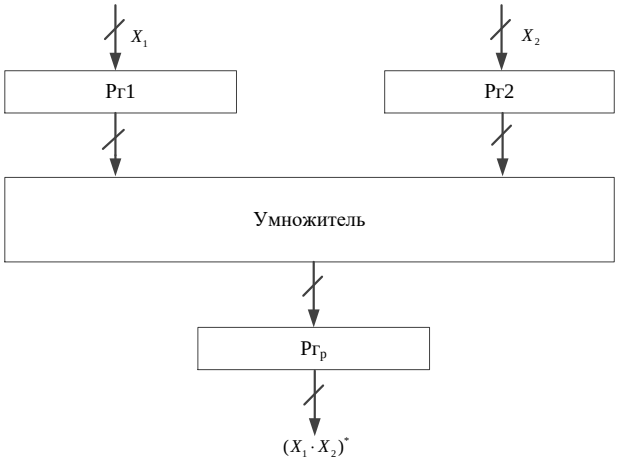


Рис. 3.Обобщённая структурно-функциональная схема вероятностного множительного устройства

В свою очередь блок «Умножитель» состоит из преобразователя «число-вероятность»; логического элемента «И» (логическое умножение);счётчика количества испытаний и

счётчика результата. Множители X_1 и X_2 поступают по шине на преобразователи «число-вероятность», где преобразуются в вероятностную форму. Вероятностные отображения множителей с преобразователей последовательно поступают на логический элемент «И», в котором мантиссы перемножаются, а результат накапливается в счетчике результата. Синхросигнал С, управляющий схемой, увеличивает значение на счётчике испытаний. Через логический элемент «И» переполнение счетчика испытаний формирует управляющий сигнал, разрешающий выдачу результата из счетчика результата на выход схемы. В итоге получаем перемноженные множители за определенное количество испытаний.

Исходя из рассмотренной обобщённой схемы вероятностного множительного устройства, составим структурно-функциональную схему усовершенствованного устройства (рис. 4).

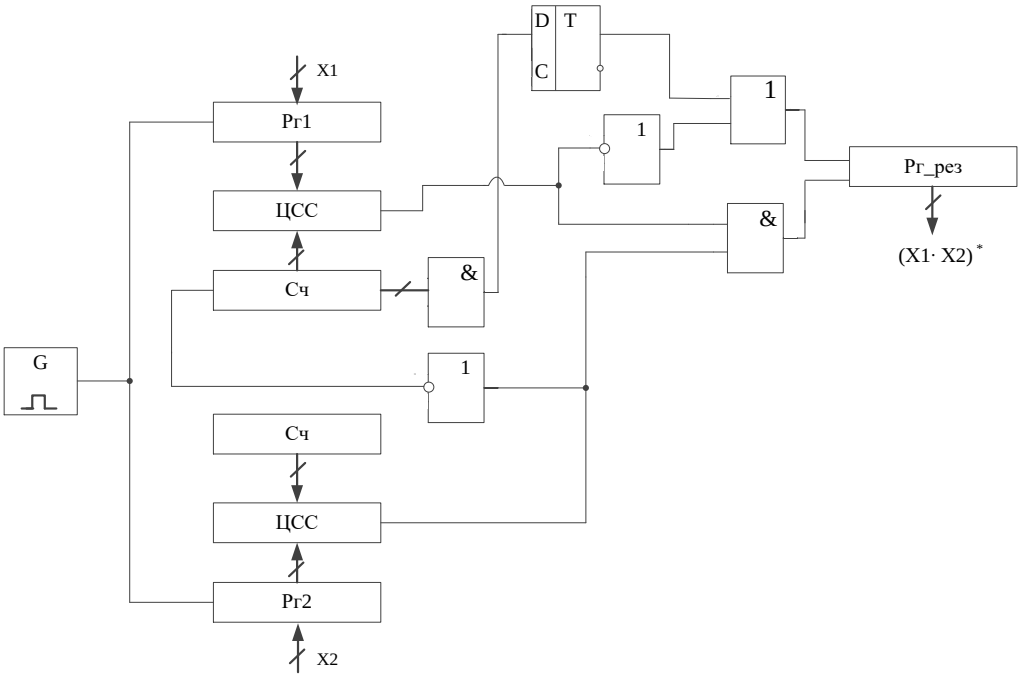


Рис.4. Усовершенствованная схема вероятностного множительного устройства

Различие между схемами «Обобщенная схема вероятностного множительного устройства» и «Усовершенствованная схема вероятностного множительного устройства» заключается в преобразователях «число-вероятность».

Выводы

Рассмотренные методы повышения точности и быстродействия вероятностного множительного устройства дают возможность на усовершенствование устройства, а также расширить область его применения.

Библиографический список

1. Моисеев, Д.В. Теория и структура генерации псевдовероятностных отображений измеряемых сигналов / Д.В. Моисеев, Н.Е. Сапожников // Зб. наук.пр. СНУЯЕтаП. – Севастополь: СНУЯЕтаП, 2014. – Вип.2 (50). – с. 221- 228.

2. Сапожников Н.Е. Оценка точности и быстродействия при вероятностной форме представления информации/Д.В. Моисеев, П.С. Бейнер, Н.В. Бейнер.// ВісникНТУ «ХПІ». – Харьков: НТУ «ХПІ», 2013, – №38(1011). – С. 34 – 39

УДК 681.3

О.В.Штолин, магистрант, студент 6-го курса
Научный руководитель: Д.В.Моисеев, канд. техн. наук, доцент
Севастопольский государственный университет

ИССЛЕДОВАНИЕ МЕТОДОВ УСКОРЕНИЯ ОПЕРАЦИИ УМНОЖЕНИЯ

Аннотация: В данной статье рассмотрены методы ускорения выполнения операции умножения и проанализирована необходимость их применения в ЭВМ. На основе проведенного исследования автором предлагается использование данных методов в вероятностной форме представления данных.

Ключевые слова: Вероятностная форма представления данных, методы ускорения логических операций.

Annotation: This article describes methods for accelerating the implementation of the operations of multiplication and analyzed the need for their application in the computer. On the basis of the research the author proposes the use of these methods in probabilistic form of data presentation.

Key words: Probabilistic form of data representation, methods of acceleration logic operations.

1. Введение

С тех пор как человек в своей повседневной деятельности перестал довольствоваться информацией, доставляемой ему лишь его органами чувств, и привлек им в помощь инструментальные средства, проблема создания и совершенствования этих средств стала одним из важнейших направлений, определяющих прогресс естественных наук и промышленного производства. Созданием и использованием средств измерений в тех или иных приложениях занимаются многие научные и производственные коллективы. К радикальным изменениям характера эксперимента привело использование вычислительной техники, сделавшей возможной оперативную обработку больших массивов информации.

Важнейшими составляющими эффективности ИИС, являются показатели живучести, надежности, точности воспроизведения исходной информации, способности функционировать в реальном масштабе времени, затрат оборудования, стоимости разработки, производства и эксплуатации.

2. Методы ускорения операции умножения

Методы ускорения умножения можно условно разделить на логические и аппаратные. Логические методы позволяют сократить время вычисления за счет более эффективных алгоритмов умножения, позволяющих уменьшить количество сложений в ходе умножения. Аппаратные методы направлены на схемное сокращение времени вычисления и суммирования частичных произведений. Логические подходы к ускорению умножения можно подразделить на две группы [1]:

– применение избыточных систем счисления;

– применение систем счисления с основанием большим двух.

Рассмотрим первую группу логических методов – умножение с использованием избыточных систем счисления. Наиболее известным и распространенным представителем этой группы является алгоритм Бута. В основе алгоритма Бута лежит следующее соотношение, характерное для последовательностей двоичных цифр:

$$2^m + 2^{m-1} + \cdots + 2^k = 2^{k+1} - 2^k,$$

где m и k – номера крайних разрядов в группе из последовательных единиц. Например, $011110 = 2^5 - 2^1$. Это означает, что при наличии во множителе групп из нескольких единиц (комбинаций вида 011110), последовательное добавление к СЧП множимого с нарастающим весом (от 2^k до 2^m) можно заменить вычитанием из СЧП множимого с весом 2^k и прибавлением к СЧП множимого с весом 2^{m+1}.

Алгоритм Бута сводится к перекодированию множителя из системы {0, 1} в избыточную систему $\{\bar{1}, 0, 1\}$. Такое перекодирование часто называют перекодированием Бута (Booth recoding) [1]. В записи множителя в новой системе 1 означает добавление множимого к сумме частичных произведений, $\bar{1}$ – вычитание множимого и 0 не предполагает никаких действий. Во всех случаях после очередной итерации производится сдвиг множимого влево или суммы частичных произведений вправо. Реализация алгоритма предполагает последовательный в

направлении справа налево анализ пар разрядов множителя – текущего b_i и предшествующего b_{i-1}. Для младшего разряда множителя (i = 0) считается, что предшествующий разряд равен 0, т. е. имеет место пара $b_0 0$. На каждом шаге i (i = 0, 1,…, n – 1) анализируется текущая комбинация $b_i b_{i-1}$.

Алгоритм предполагает три операции: сдвиг, сложение и вычитание. Комбинация 10 означает начало цепочки последовательных единиц, и в этом случае производится вычитание множимого из СЧП. Комбинация 01 соответствует завершению цепочки единиц, и в этом случае множимое прибавляется к СЧП. Комбинация 00 свидетельствует об отсутствии цепочки единиц, а 11 – о нахождении внутри такой цепочки. В обоих случаях никакие арифметические операции не производятся. По завершении описанных действий осуществляется сдвиг множимого влево либо суммы частичных произведений вправо, и цикл повторяется для следующей пары разрядов множителя. Если последняя пара разрядов множителя $b_{n-1}\, b_{n-2}$ равна комбинациям 10 или 11, должна быть выполнена коррекция результата, которая заключается в добавлении множимого к последней сумме частичных произведений.

В приведенных примерах используется вариант умножения со сдвигом множимого влево. Операция вычитания, как это принято в реальных умножителях, выполняется путем сложения с множителем, взятым с противоположным знаком и представленным в дополнительном коде.

Пример 1. Умножить два целых числа без знака по алгоритму Бута:

A = 6, B = 7 (n = 4).

Решение. $A = 00000110$, $[-A]_{\text{доп}} = 11111010$, $B = 0111$.

После перекодирования Мт 0111 приобретает вид $100\bar{1}$.

Шаг	МТ	МН и СЧП	Действие
0	0111	00000000	$\text{СЧП}_0 = 0$
1	011[10] => -1	11111010 11111010 00001100	Начало цепочки единиц (-Мн) СЧП_1 Сдвиг Мн влево
2	01[11] => 0	---------- 11111010 00011000	Внутри цепочки единиц СЧП_2 Сдвиг Мн влево
3	0[11]0 => 0	---------- 11111010 00110000	Внутри цепочки единиц СЧП_3 Сдвиг Мн влево
4	[01]11 => 1	00110000 00101010 1-отбрасывается	Конец цепочки единиц (+Мн) $P = \text{СЧП}_4 = 6 * 7 = 42$

На практике большее распространение получил модифицированный алгоритм Бута, в котором производится перекодировка множителя из стандартной двоичной системы {0, 1} в избыточную систему $\{\bar{2},\bar{1},0,1,2\}$[2]. В этой системе каждое число обозначает коэффициент, на который умножается множимое перед добавлением к СЧП:

$\bar{2}$ – вычитание удвоенного множимого из СЧП,

$\bar{1}$ – вычитание множимого,

2 – прибавление удвоенного множимого,

1 – прибавление множимого,

0 не предполагает никаких действий.

В процессе умножения одновременно анализируются три разряда множителя $b_{i+1}b_ib_{i-1}$ – два текущих и старший разряд из предыдущей тройки. В зависимости от комбинации 0 и 1 в этих разрядах выполняется прибавление или вычитание множимого, прибавление или вычитание удвоенного множимого, либо никакие действия не производятся в соответствии с таблицей 1.

Такая модификация позволяет еще больше сократить количество операций сложения, выполняемых в процессе умножения. Например, как видно из таблицы 1, при комбинации 010 в соответствии с модифицированным алгоритмом необходимо выполнить только одну операцию сложения – прибавить к СЧП множимое. В то же время при такой комбинации стандартный алгоритм Бута предполагает выполнение двух операций: вычесть из СЧП множимое и после сдвига множимого или СЧП добавить множимое.

Таблица 1- таблица модифицированного алгоритма Бута

b_{i+1}	b_i	b_{i-1}	Код	Действия
0	0	0	0	Не выполнять никаких действий
0	0	1	1	Прибавить к СЧП множимое
0	1	0	1	Прибавить к СЧП множимое
0	1	1	2	Прибавить к СЧП удвоенное множимое
1	0	0	-2	Вычесть из СЧП удвоенное множимое
1	0	1	-1	Вычесть из СЧП множимое
1	1	0	-1	Вычесть из СЧП множимое
1	1	1	0	Не выполнять никаких действий

Алгоритм Лемана. Еще большее сокращение количества сложений может дать модификация алгоритма Бута, предложенная Леманом. Суть модификации заключается в следующем[3]:

– если две группы нулей разделены единицей, стоящей в k-й позиции, то вместо вычитания в k-й позиции и сложения в (k+1)-й позиции достаточно выполнить только сложение в k-й позиции;

– если две группы единиц разделены нулем, стоящим в k-й позиции, то вместо сложения в k-й позиции и вычитания в (k+1)-й позиции достаточно выполнить только вычитание в k-й позиции.

Обработка двух разрядов множителя за шаг. Из второй группы логических методов приведем метод умножения с обработкой за шаг двух разрядов множителя[4]. В принципе это более эффективный вариант алгоритма Бута. Анализ множителя начинается с младших разрядов. В зависимости от входящей двухразрядной комбинации предусматриваются следующие действия:

00 – сдвиг на два разряда вправо суммы частичных произведений;

01 – к СЧП прибавляется множимое, после чего СЧП сдвигается на 2 разряда вправо;

10 – к СЧП прибавляется удвоенное множимое, СЧП сдвигается на 2 разряда вправо;

11 – из СЧП вычитается множимое, СЧП сдвигается на 2 разряда вправо. Полученный результат должен быть скорректирован на следующем шаге, что фиксируется специальным признаком коррекции.

Так как в случае пары 11 из СЧП вычитается одинарное множимое вместо прибавления утроенного, для корректировки результата к СЧП перед выполнением сдвига надо было бы прибавить учетверенное множимое. Но после сдвига на два разряда вправо СЧП уменьшается в четыре раза, так что на следующем шаге достаточно добавить одинарное множимое. Это учитывается при обработке следующей пары разрядов множителя, путем обработки пары 00 как 01, 01 – как 10, 10 – как 11, а 11 – как 00. В последних двух случаях фиксируется признак коррекции. Правила обработки пар разрядов множителя с учетом признака коррекции приведены в таблице 2.

После окончания процесса умножения требуется корректировка результата, если старшая пара разрядов множителя равна 11 или 10, и состояние признака коррекции единичное. В этом случае к полученному произведению должно быть добавлено множимое. Данный метод применим и для варианта умножения со сдвигом множимого влево (при этом множимое сдвигается на 2 разряда).

Таблица 2- Правила обработки пар разрядов

Пара разрядов множителя	Признак коррекции из предыдущей пары	Признак коррекции в следующую пару	Действия
00	0	0	-
01	0	0	+Мн
10	0	0	+2Мн
11	0	1	-Мн
00	1	0	+Мн
01	1	0	+2Мн
10	1	1	-Мн
11	1	1	-

Аппаратные методы ускорения умножения. Аппаратные методы ускорения умножения сводятся:

– к уменьшению времени распространения переносов при суммировании частичных произведений;

– к параллельному вычислению частичных произведений.

Методы первой группы используют более эффективные способы суммирования ЧП, исключающие затраты времени на распространение переносов. Достигается это за счет представления ЧП в избыточной форме, благодаря чему суммирование двух чисел не связано с распространением переноса вдоль всех разрядов числа. Наиболее употребительной формой такого избыточного кодирования является так называемая форма с сохранением переноса. В ней каждый разряд числа представляется двумя битами cs – перенос (c) и сумма (s). При суммировании двух чисел в форме с сохранением переноса перенос распространяется не далее чем на один разряд. Это делает процесс суммирования значительно более быстрым, чем в случае сложения с распространением переноса вдоль всех разрядов числа.

Вторая возможность ускорения операции умножения заключается в параллельном вычислении всех частичных произведений. Если рассмотреть общую схему умножения, то

нетрудно заметить, что отдельные разряды ЧП представляют собой произведения вида $a_i b_j$, т. е. произведение определенного бита множимого на определенный бит множителя. Это позволяет вычислить все биты частичных произведений одновременно с помощью n^2схем И.

3. Вывод

Методы ускорения логических операций позволяют значительно увеличить скорость обработки данных, что, в свою очередь, открывает возможность создания новых, более функциональных систем.

Библиографический список

1. М. В. Качинский, В. Б. Клюс, А. Б. Давыдов. Арифметические основы электронных вычислительных средств Минск : БГУИР, 2014. – 620 с.

2. Каган Б.М “Электронные вычислительные машины и системы“ Москва: Энергоатомиздат, 1991 Учебное пособие для вузов. – 270 с.

3. Цилькер Б.Я. Организация ЭВМ и систем : Учебник для вузов / Б.Я. Цилькер, С.А. Орлов. – 2-е изд. – СПб.: Питер, 2011. -355 с.

4. Курсовая работа “Блок ускоренного выполнения операции умножения целых чисел” Министерство образования и науки Российской Федерации ФГАОУ ВПО «Уральский федеральный университет имени первого Президента России Б. Н. Ельцина» 2014г.

УДК 681.3
В.В. Щербатей магистрант
Научный руководитель: Д.В. Моисеев к.т.н., доцент
Севастопольский государственный университет

ИССЛЕДОВАНИЕ МЕТОДОВ ПРЯМОГО И ОБРАТНОГО ВЕРОЯТНОСТНОГО ПРЕОБРАЗОВАНИЯ ДАННЫХ ПРИ НЕРАВНОМЕРНЫХ ЗАКОНАХ РАСПРЕДЕЛЕНИЯ ВСПОМОГАТЕЛЬНОГО СЛУЧАЙНОГО СИГНАЛА

Аннотация: данная работа посвящена исследованию методов прямого и обратного вероятностного преобразования данных при неравномерных законах распределения вспомогательного случайного сигнала
Ключевые слова: вероятностная форма представления данных, нормальный закон распределения, закон распределения
Annotation: this work is devoted to research methods of direct and inverse probability data conversion when the law of uneven distribution of the auxiliary random signal
Key words: probabilistic form of data presentation, normal distribution, distribution law

Введение

Эффективность использования любой информации зависит от точности ее представления — свойства, отражающего близость результатов представленных в какой-либо форме к истинным значениям. Это же касается и данных, представленных в вероятностной форме, которые априори будут иметь некоторую погрешность. Точность представления данных может быть большей или меньшей, в зависимости от различных факторов, к примеру, таких как закон распределения генерации псевдослучайных чисел, количества генераций, используемых для представления вероятностного отображения и от самой преобразуемой величины.

При выполнении операции вероятностного преобразования сигнала в вероятностное отображение будет абсолютно точно соответствовать преобразуемому сигналу только при бесконечном количестве независимых статистических испытанийK (без учета методической погрешности). В реальных условиях значениеK приходится выбирать, исходя из заданного быстродействия, либо частотного диапазона преобразуемого сигнала при работе в реальном масштабе времени. Это приводит к появлению специфической погрешности, которую называют погрешностью вероятностного преобразования.

Постановка задачи

Целью данной работы является определение зависимости абсолютной погрешности представления данных в вероятностной форме на интервале [0;1] от закона распределения псевдослучайных чисели от количества генераций вспомогательного сигнала, а также от самого значения преобразуемой величины.

Решение задачи

В общем виде суть стохастического или вероятностного преобразования заключается в том, что любому значению параметра преобразуемой величины можно привести в соответствие некоторую вероятность.

Для реализации вероятностного кодирования, необходимо обратиться к основному принципу вероятностного представления информации, в соответствии с которым каждому значению входной величины можно поставить в соответствие некоторую вероятность. Данный принцип реализуется за счет выполнения пошагового процесса сравнения входной величины со значением случайного вспомогательного сигнала. Все выше сказанное укладывается в следующее выражение:

$$p_{ij} = \begin{cases} 1, \text{для всех } a_i \geq R(t) \\ 0, \text{для всех } a_i < R(t) \end{cases}, \quad (1)$$

гдеP_{ij}– j-е значение вероятностного отображения из множества:

$$p_i = \{p_{i1}, p_{i2}, p_{i3}, ..., p_{i1}, ..., p_{ik}\};$$

a_i – i-й преобразуемый символ;

j – цикл вероятностного преобразования;

R(t) – произвольное значение из ряда: $R\{r_1, r_2, r_3, ..., r_i, ..., r_k\}$.

Исходя из выражения (1), можно уверенно сказать, что данный код относится к весовым кодам, учитывая свойства вспомогательного сигнала *R(t)*, следует отметить, что в виду свойств данного сигнала, каждому a_i будет соответствовать не единственное значение p_i , а множество вероятностных отображений с одинаковым весом.

Данное свойство вытекает из выражения (1), его наглядно можно представить графически. На рисунке 1 показан процесс вероятностного преобразования входного значения *a,* равного 7, при помощи двух вспомогательных псевдослучайных равномерно распределенных случайных сигналов *R* и *R1* (значения уровней которых представлены на верхнем графике), при помощи которых соответственно получаются два вероятностных отображения *p* и *p1* (значения уровней представлены на нижнем графике).

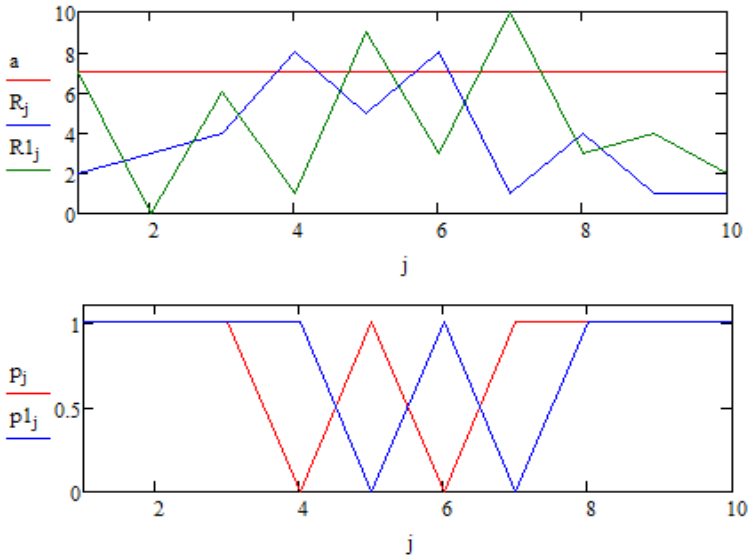


Рис.1. Графическое представление процесса вероятностного преобразования при равномерном законе распределения

Как видно из графиков, вероятностному отображению p будет соответствовать последовательность:

1110101111,

а p1:

1111010111.

На втором этапе моделирования рассмотрено влияние законов распределения на корректирующие свойства вероятностного преобразования. Для этого вместо равномерного закона распределения применимтакие законы распределения, как нормальный закон распределения и гамма распределение.

Пример их реализации можно увидеть на рисунках 2 - 5.

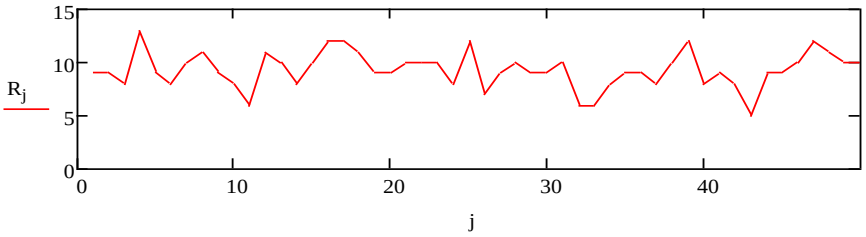


Рис.2. Графическое представление процесса вероятностного преобразования при нормальном законе распределения

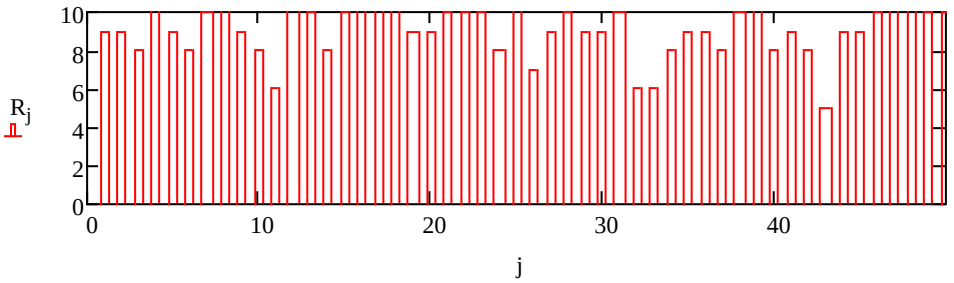


Рис.3. Графическое представление процесса вероятностного преобразования при нормальном законе распределения

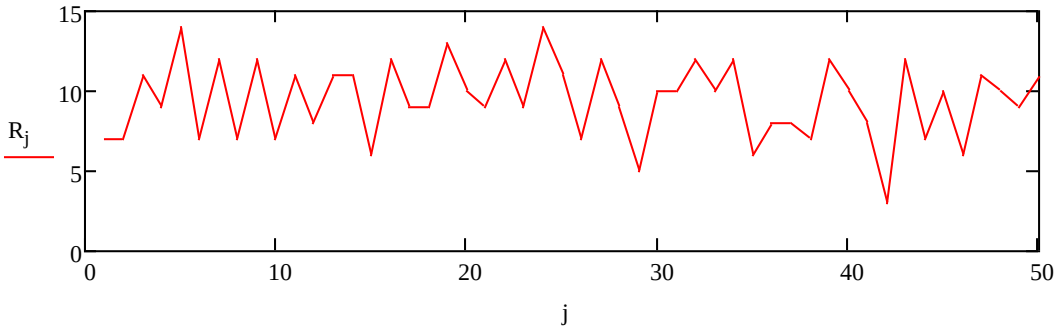


Рис.4. Графическое представление процесса вероятностного преобразования при гамма распределении

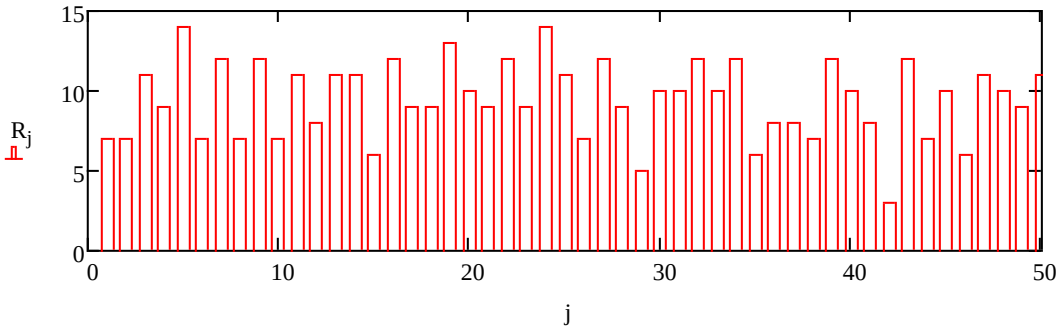


Рис.5. Графическое представление процесса вероятностного преобразования пригамма распределении

Выводы

В результате проанализированных литературных источников по вопросам вероятностного представления данных, был создан алгоритм, позволяющий представить числа в вероятностной форме, а также привести их к графической форме являющейся более удобной для проведения анализа результатов.

Библиографический список

1. Моисеев, Д.В. Теория и структура генерации псевдовероятностных отображений измеряемых сигналов / Д.В. Моисеев, Н.Е. Сапожников // Зб. наук.пр. СНУЯЕтаП. – Севастополь: СНУЯЕтаП, 2014. – Вип.2 (50). – с. 221- 228.

2. Сапожников Н.Е Вероятностные вычислительные модели / Н.Е. Сапожников, Д.В. Моисеев, А.Г. Шокин, Ю.А. Барановский // Зб. наук.пр. СНУЯЕтаП. – Севастополь: СНУЯЕтаП, 2013. – Вип.1 (45) С.– 210–215.

3. Сапожников Н.Е. Оценка точности и быстродействия при вероятностной форме представления информации / Н.Е. Сапожников, Д.В. Моисеев, Ю.Ю. Столярчук // Зб. наук.пр. СНУЯЕтаП. – Севастополь: СНУЯЭиП, 2011. – Вип. 3 (39). − С. 134 - 140.

СЕКЦИЯ 6

Компьютерные сети

УДК 004.722:004.725.2:004.771:004.451:004.05
А.Д. Бернацкий, студент 4-го курса
Научный руководитель: И.В. Дымченко, старший преподаватель
Севастопольский государственный университет

ПУТИ МОДЕРНИЗАЦИИ ИНФОКОММУНИКАЦИОННОЙ СИСТЕМЫ КАФЕДРЫ «ИНФОРМАЦИОННЫЕ СИСТЕМЫ»

Аннотация: в статье приводится описание структуры внедряемой инфокоммуникационной системы кафедры «Информационные системы» Севастопольского государственного университета, предназначенной для ведения учебно-проектной деятельности, а так же выбор и обоснование компонентов системы.
Ключевые слова: инфокоммуникациионная система и сеть, топология, сервер, виртуализация, гипервизор, информационная система
Annotation: This article provides a description of the structure of the department implemented infocomm systems "Informa-tion systems" Sevastopol National Technical University, intended to conduct training and project activities, as well as the selection and justification of the system components.
Keywords: infocommunication system and network topology, server virtualization, hypervisor, information system

Введение. Требования Федерального государственного образовательного стандарта [1] к профессиональным компетенциям выпускников направления «Информационные системы» в области проектной деятельности, определяют уровень оснащения учебно-материальной базы, как выпускающей кафедры, так и Университета в целом. Данная инфраструктура должна обеспечить возможность обучения, приближенную к реальной профессиональной деятельности и предоставить доступ к самым современным технологиям проектирования и разработки информационных систем и технологий.

Цель статьи. Исследовать состояние инфокоммуникационной среды выпускающей кафедры, описать решение, принятое по ее модернизации и дать представление о возможностях проектной деятельности, предоставляемых данным решением.

Основная часть. В настоящее время локальная вычислительная сеть кафедры, вследствие стихийного развития, не имеет четко выраженной топологии и как результат – обладает невысокой пропускной способностью.

Проведенная модернизация позволила привести сеть к топологии «звезда», где каждое помещение кафедры подведено к магистральному сетевому оборудованию университета, рис.1. Подключение будет осуществляться к магистральному оборудованию, которое в свою очередь соединено с основным сервером университета по оптическим линиям связи.

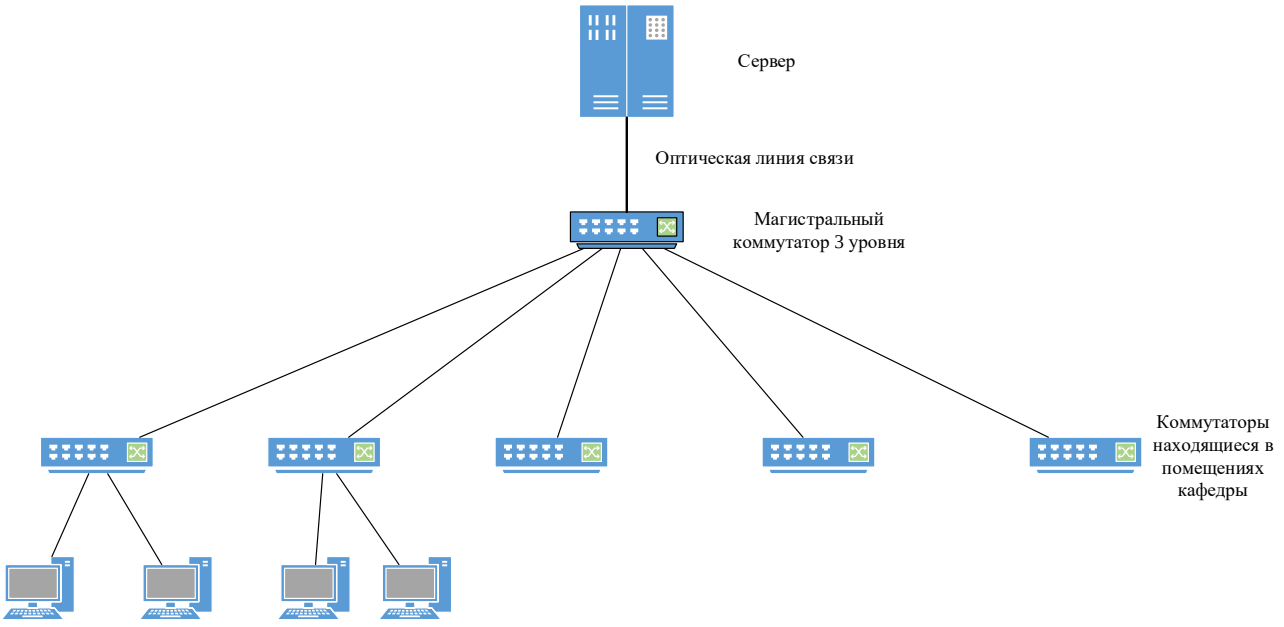


Рис.1. Топология «звезда» модернизированной сети.

Данное решение позволит обеспечить каждую лабораторию или учебный компьютерный класс доступом к необходимым вычислительным ресурсам, как при проведении учебных занятий, так и при проектных разработках.

В настоящий момент на кафедре запущены несколько учебных и пилотных проектов (Web-ГИС СевГУ, Web-ГИС «Археолог-Херсонес», информационная система «Агрофирма+» и др), которые предполагают разработку информационных систем (ИС) для различных сфер деятельности. Разрабатываемые системы используют различные технологические схемы построения, программное обеспечение (ПО), многопользовательский доступ к хранилищам данных. Следует так же отметить стремление к использованию кроссплатформенных решений и ПО с открытым исходным кодом (free software).

Все вышеперечисленное определяет использование следующих компонентов:

− высокопроизводительный сервер, который призван обеспечить контроль за рабочими станциями лабораторий и учебных классов и предоставить вычислительные мощности;

− сервер резервного копирования данных, обеспечивающий создания резервных копий систем виртуальных машин и (или) отдельных данных по определенному алгоритму.

Исходя из наличия одного физического сервера, а так же необходимостью выполнения сервером нескольких функций: принято решение о применении технологии виртуализации, т.е. предоставление набора вычислительных ресурсов или их логического объединения, абстрагированное от аппаратной реализации, и обеспечивающее при этом логическую изоляцию вычислительных процессов, выполняемых на одном физическом ресурсе, рис.2.

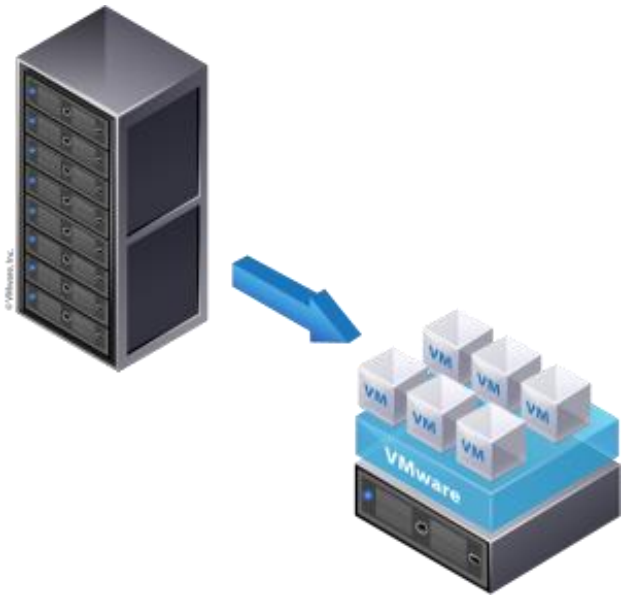


Рис.2. Виртуализация на основе VMware ESXI

Первоначальные настройки сервера включают создание Raid массива(ов), учитывая, что на момент написания статьи объем постоянной памяти неизвестен, рассматриваются несколько вариантов:

1. Raid 1+0 (виртуальные машины), Raid 1 (сервер резервного копирования);
2. Raid 5 (виртуальные машины), Raid 1 (сервер резервного копирования);
3. Raid 5 (виртуальные машины), отдельный жесткий диск для сервера резервного копирования.

Наилучшим вариантом является первый, в этом случае, на сервере будут созданы два Raid массива 1+0, что позволит обеспечить высокую отказоустойчивость, а также высокую

скорость считывания записи информации. Один Raid массив будет использоваться для основных виртуальных машин, второй для сервера резервного копирования.

Средним вариантом будет использование 5-ти жестких дисков (вариант 2), что обеспечит среднюю отказоустойчивость и среднюю скорость считывания, но при этом будет получена высокая отказоустойчивость жестких дисков с резервными копиями.

Наихудший вариант – 3-й, который предполагает работу при количестве жестких дисков равным 4. Использование Raid 5 обеспечит среднюю отказоустойчивость и среднюю скорость считывания, однако наличие отдельного жесткого диска для резервного копирования не гарантирует отказоустойчивость – это означает, что с выходом из строя данного жесткого диска будут утеряны все резервные копии.

В качестве инструмента виртуализации принято решение использовать гипервизор ESXI (vSphereHypervisor). ESXI – это аппаратный гипервизор, устанавливаемый непосредственно на физический сервер и разделяющий его на несколько виртуальных машин. Все виртуальные машины используют одни и те же физические ресурсы и могут работать одновременно. В отличие от других гипервизоров управление платформой vSphere осуществляется с помощью средств удаленного управления, рис.3.

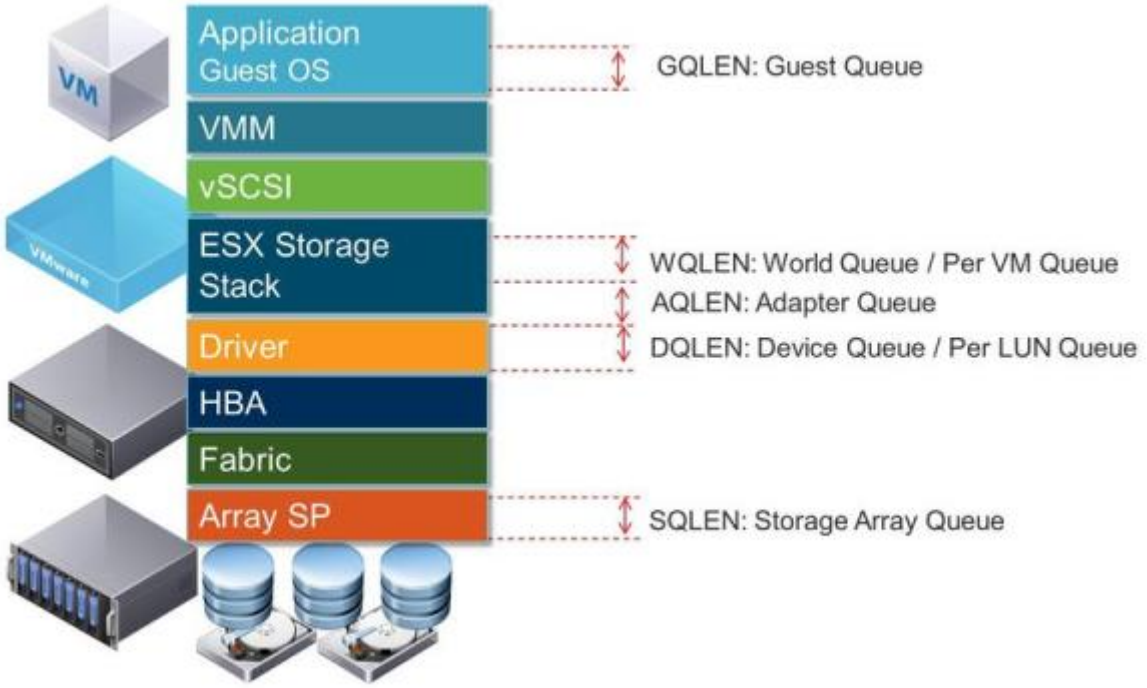


Рис.3. Иерархия компонентов виртуальной машины

Еще один момент, определивший выбор данного инструмента виртуализации, (помимо лидерства на рынке среди подобных продуктов) это наличие бесплатной лицензии. Бесплатная версия имеет ограничения в возможностях (создание\настройка кластера, объем оперативной памяти, объем ПЗУ виртуальной машины и т.д.), но при установке и настройке данного сервера эти ограничения не имеют значения.

В качестве основного серверного программного обеспечения выбор остановился на WindowsServer 2012 R2 StandardEdition, поскольку данное ПО является одним из лучших вариантов, обеспечивающих контроль рабочих станции, разграничение прав доступа к сетевым ресурсам, выдачу каждому пользователю определенных прав, как на клиентских компьютерах лабораторий и учебных классов, так и к операционной системе виртуальной машины сервера.

Технология виртуализации так же позволит иметь виртуальную машину с Unix\LinuxServer.

В результате планируется установить и настроить, как минимум, три виртуальные машины:

- WindowsServer 2012 R2 StandardEdition;
- сервер резервного копирования;
- Unix\Linux\WindowsServer (в зависимости от требуемой ОС).

Виртуальная машина с ОС Windows Server 2012 R2 Standard Edition является основной на данном сервере. После настройки служба ActiveDirectory (AD) он становится контроллером домена. Так же машина предназначена для развертывания DNS-сервера.

Установка и настройка службы AD обеспечит разграничение прав доступа пользователей к ресурсам сети, а DNS-сервер обеспечит распознавание имен учетных записей пользователей и соответствия им клиентских компьютеров.

Поскольку некоторые проекты предполагают разработку Web-ресурсов, виртуальная машина с ОС Windows Server 2012 R2 Standard Edition позволит их разместить, а DNS-сервер понадобится для распознавания символьных адресов сайтов и перенаправления на необходимый ip-адрес.

Выводы и дальнейшее направление исследований. Проведенная модернизация позволит существенно повысить пропускную способность сети. Дальнейшее развитие материально-технической базы кафедры предполагает использование больших вычислительных мощностей для реализации как учебной, так и проектной деятельности. Технологическим решением их обеспечения является создание кластера(ов). Предложенная структура сервера позволит быстро и без осложнений настроить кластер на основе текущего сервера.

Библиографический список

1. Федеральный государственный образовательный стандарт высшего образования по направлению подготовки 09.03.02 Информационные системы и технологии (уровень бакалавриата), утвержденный приказом Министерства образования и науки Российской Федерации от «12» марта 2015 г. №219.
2. Windows Server 2012 R2 [Электронный ресурс] URL: https://www.microsoft.com/ru-ru/server-cloud/products/windows-server-2012-r2/Comparison.aspx(Дата обращения: 07.03.16)
3. vSphere и vSphere with Operations Management [Электронный ресурс] URL: http://www.vmware.com/ru/products/vsphere/ (Дата обращения: 03.03.16)
4. vSphere и vSphere with Operations Management [Электронный ресурс] URL: http://www.vmware.com/ru/products/vsphere/compare.html (Дата обращения: 04.03.16)

УДК 004.7
В.Г. Лисецкий, магистрант 2-го курса
Научный руководитель: Ю.К. Апраксин, канд.техн.наук, проф.
Севастопольский государственный университет

ВЕРИФИКАЦИЯ АВТОМАТНЫХ СПЕЦИФИКАЦИЙ ПРОТОКОЛОВ ДЛЯ СЕТЕЙ ИНФОРМАЦИОННОГО ВЗАИМОДЕЙСТВИЯ

Аннотация: Рассматриваются особенности верификации автоматных спецификаций протоколов управления информационным обменом в распределенных технических системах. Исследуется возможность автоматической верификации спецификаций при помощи метода достижимых глобальных состояний.
Ключевые слова: сети информационного взаимодействия, распределенные технические системы, достижимые глобальные состояния, верификация протокольных спецификаций
Annotation: This article deals with verification features of protocols automatic specifications of information exchange in distributed computer networks. A possibility to verify specifications automatically using reachable global states method is researched.
Key words: information interaction networks, distributed technical systems, reachable global states, protocol specifications verification.

Неотъемлемой частью развития информационных технологий является увеличение сложности решаемых задач, а следовательно и технических систем. Такие системы обычно содержат большое количество территориально распределенных взаимосвязанных элементов, главной целью которых является обмен информацией. Уникальность распределенных технических систем (РТС), динамичность их структуры, нестационарность поведения, неформализуемость цели существования делают применение в отношении к ним традиционных методов управления мало эффективным [1].

При проектировании протоколов обмена информацией между техническими системами особенно важным является этап верификации, при котором определяется их логическая корректность, так как только логически корректные протоколы в состоянии выполнять свои управленческие функции.

Объектом исследования являются информационные системы на уровне взаимодействия путем посылки сообщений. Каждый объект может быть представлен в виде конечного автомата, изменяющего своё состояние в следствие приёма или посылки сообщения. Поскольку изменение состояния подразумевает под собой не переключение триггеров, а изменение ситуации в каком-либо процессе, такая модель может быть использована для управления в критических системах, так как критические ситуации могут быть определены заранее и можно задать действия автомата, направленные на избежание таких ситуаций.

В данной работе для верификации протоколов предлагается метод достижимых глобальных состояний, применимый к модели взаимодействующих конечных автоматов, объединенных каналами ограниченной емкости. Модель канала – очередь с дисциплиной обслуживания типа FIFO. Два протокольных автомата PA_i и PA_j обслуживаются двумя симплексными каналами $C_{i,j}$ и $C_{j,i}$ которые определяются последовательностями сообщений, находящихся в этих каналах. Каждый канал имеет максимальную емкость $(C_{i,j})_{max}$. Время передачи рассматривается как элементарное событие, реализуемое мгновенно. Канал $C_{i,j}$ принимает события от протокольного автомата PA_i и доставляет их без искажений и изменения последовательности протокольному автомату PA_j. Глобальное состояние системы n взаимодействующих автоматов представляется матрицей $S = || S_{i,j} ||_{nxn}$, где $S_{i,j} = S_i$, если i = j, иначе – $S_{i,j} = C_{i,j}$; S_i – текущее состояние PA_i, $C_{i,j}$ – последовательность сообщений, находящихся в канале.

Переход из одного глобального состояния в другое происходит после выполнения одиночного перехода в одном из протокольных автоматов. В результате такого перехода изменяется текущее состояние выбранного автомата PA_i и одного из каналов $C_{i,j}$ (в случае посылки сообщения) или $C_{j,i}$ (если инициатором перехода является прием сообщения из

канала). Состояния каналов остаются без изменения, если происходит внутреннее событие, не связанное с приемом или передачей сообщений.

В процессе генерации всех возможных глобальных состояний строится граф достижимости глобальных состояний, вершинами которого являются глобальные состояния (начальное, текущие, конечные), а дугами - события, инициирующие соответствующий переход. Анализ такого графа дает возможность проверить, какие свойства протокола нарушаются, т.е. определить и классифицировать ошибочные ситуации, возникающие в процессе функционирования протокола.

Ошибочная ситуация типа «переполнение канала» возникает при нарушении свойства ограниченности протокола. Под ограниченностью протокола подразумевается, что при функционировании протокола число сообщений в каждом канале между протокольными объектами не превышает емкости канала. Обнаружение этой ошибки может производиться путем подсчета сообщений в массиве, описывающем канал передачи данных с дисциплиной обслуживания FIFO и сравнением полученного числа сообщений с емкостью канала.

Ошибочная ситуация типа «невыполнимое взаимодействие» возникает при избыточности спецификации. Отсутствие избыточности спецификации – это отсутствие описания обработки непоступивших сообщений. Переопределенность протокола сама по себе не приводит к некорректности, однако затрудняет анализ и тестирование протоколов, поскольку увеличивает размерность модели. Распознавание данной ошибочной ситуации может производиться путем анализа таблиц приема и передач (все ли передаваемые и принимаемые сообщения, указанные в таблицах, задействованы при построении дерева достижимости?). Однако, обнаружение невыполнимых взаимодействий должно проводиться на этапе когда все возможные ошибки уже выявлены и откорректированы. Иначе невыполнимые взаимодействия будут лишь следствиями ошибочных ситуаций, препятствующих обмену сообщениями в протоколе.

«Неспецифицированный прием» - это тип ошибки, возникающий при нарушении полноты описания протокола. Протокол, не обладающий свойством полноты, считается недоопределенным. В спецификации такого протокола не предусмотрен прием всех возможных сообщений ($c_{i,j} = m.X$, а $\varphi_i, (s_i, +m)$ не определено). Для обнаружения ошибочной ситуации типа «неспецифицированный прием» рассматриваются таблицы приемов для PAi

($i = 1,2,...,n$). Если PAi находится в некотором состоянии s_i и поступает входное сообщение +m, а состояние перехода не определено, то такая ситуация соответствует ошибке «неспецифицированный прием».

Ошибка типа «статический тупик» фиксируется, если для всех $\varphi_i, (s_i, -m)$ не определено и для всех $c_{j,i} = \lambda$, т.е. тупиковой является такая ситуация, когда PAi ожидает поступления сообщения из канала, а каналы пусты, причем для всех PAi $\varphi_i, (s_i, -m)$ не определены. Статический тупик распознается при анализе таблиц приемов PAi и каналов $c_{j,i}$. Если PA; находятся в состояниях, позволяющих принимать сообщения из каналов, а соответствующие каналы пусты, то идентифицируется ошибка типа статический тупик. При этом, состояния, в которых находятся PAi, не являются конечными.

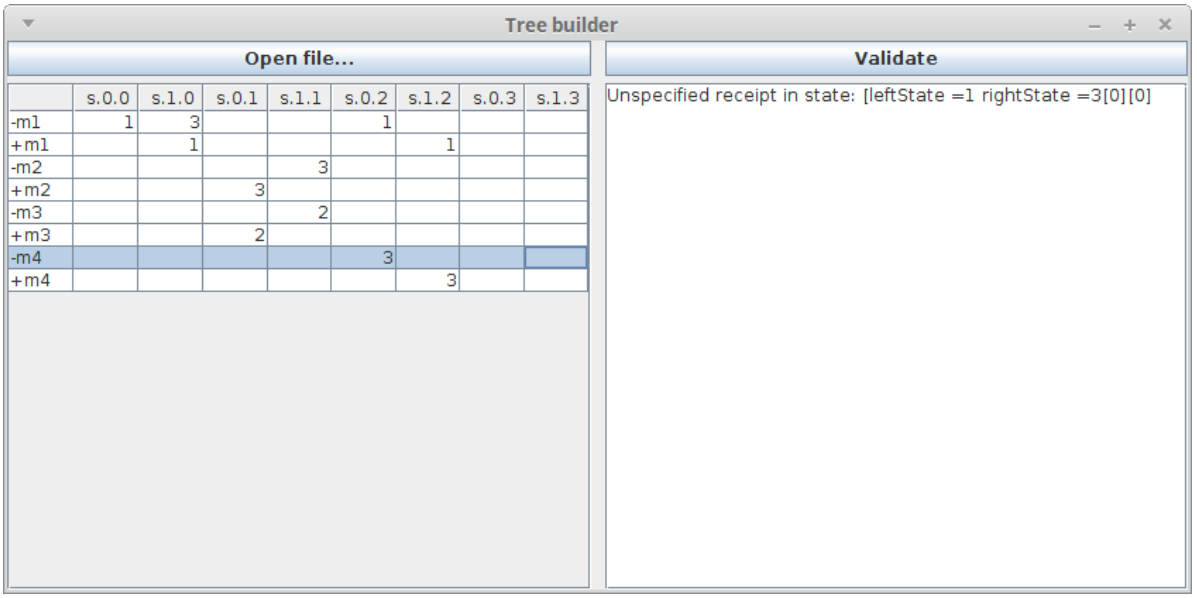
«Динамический тупик» - это идентификатор ошибочной ситуации, которую можно охарактеризовать как наличие непродуктивных циклов, в которых происходит бесполезный обмен одними и теми же сообщениями. В автоматных моделях непродуктивным называется любой цикл в графе глобальных состояний, не позволяющий достичь конечных состояний. Необходимым условием возникновения непродуктивного цикла является наличие хотя бы в одном из протокольных объектов циклически выполняемых процедур. Следовательно, для выявления динамического тупика необходимо найти цикл в графе глобальных состояний и выяснить, есть ли выход из этого цикла на конечные состояния. Если такого выхода нет, то рассматриваемый цикл является неэффективным.

«Неоднозначность глобальных состояний» характеризует ситуацию, когда для какой-либо пары устойчивых глобальных состояний все элементы за исключением одного попарно

равны, т.е. для некоторого i Si ≠ S`i, в то время как для любых j ≠ i Sj = S`j и Ci,j = Cj,i = λ . Ситуация ошибочна и свидетельствует о том, что конкретное состояние системы взаимодействия может быть представлено более, чем одним устойчивым состоянием.

«Невыполнение назначения протокола» тип ошибки, характеризующий при отсутствии у протокола свойства корректного выполнения. Корректное выполнение назначения протокола означает, что протокол обеспечивает предоставление конкретных услуг. Если при окончании построения дерева достижимых состояний множество конечных состояний не является полным, то свойство корректного выполнения назначения протокола не выполняется.

Перечисленные выше характеристики распознаются методом достижимых глобальных состояний. Для обеспечения исследования работоспособности метода и возможности его автоматизации разработана программная система (рисунок 1).



	s.0.0	s.1.0	s.0.1	s.1.1	s.0.2	s.1.2	s.0.3	s.1.3
-m1	1	3			1			
+m1		1				1		
-m2				3				
+m2			3					
-m3				2				
+m3			2					
-m4					3			
+m4						3		

Рис.1. Графический интерфейс программной системы для исследования метода достижимых глобальных состояний

В ходе исследования в программную систему вводились спецификации, содержащие одну или более ошибочных ситуаций. В результате выявлено, что метод достижимых глобальных состояний действительно позволяет определить все ошибочные ситуации перечисленные выше.

Метод глобальных достижимых состояний имеет существенный недостаток – это лавинообразный рост числа состояний. В дальнейшем планируется исследовать эффективность декомпозиции, позволяющего разбивать процесс на минимальные по сложности процессы, резолюция о корректности которых могла бы быть распространена и на процесс в целом.

Библиографический список

1. Апраксин Ю.К. Автоматизация управления информационным взаимодействием в распределенных системах на основе специфицированных протокольных автоматов / Ю.К. Апраксин. – Севастополь, 2004. – 328 с.
2. Апраксин Ю.К. К вопросу автоматизации проектирования протоколов вычислительных сетей на основе конечноавтоматной модели / Ю.К. Апраксин // Автоматика и вычислительная техника. — 1987.- No2. — С.12 – 19.
3. Протоколы информационно-вычислительных сетей: Справочник / Под ред. И.А. Мизина, А.П. Кулешова. – М.: Радио и связь, 1990. – 504с.

УДК 004.7
Л.Т.Мирхалыкова, магистрант
Научный руководитель: Ю.К. Апраксин, д.т.н., профессор
Севастопольский государственный университет

РАЗРАБОТКА И ИССЛЕДОВАНИЕ СИСТЕМЫ МОДЕЛИРОВАНИЯ СЕТЕВЫХ ОБЪЕКТОВ ПРОТОКОЛЬНОГО ВЗАИМОДЕЙСТВИЯ

Аннотация: в статье рассматривается созданная система моделирования сетевых объектов протокольного взаимодействия. Описывается язык спецификаций (таблицы событий) и возможные ситуации при моделировании. Исследуется реакция системы на эти ситуации.
Ключевые слова: таблица событий, спецификация, событийное моделирование.
Annotation:The article deals with the established system of modeling of network objects protocol interaction. Describes the language specifications (event table), and the possible situations in the simulation. We study the system response to these situations.
Key words: event table,specification,event simulation.

Особенностью развития современной техники и информационных технологий является взаимообусловленное интенсивное усложнение как технических задач, так и решаемых ими задач. Такие системы представляют собой в общем случае сложные искусственные объекты, которые содержат элементы, распределенные как во времени, так и территориально.Распределенные технические системы (РТС) все глубже проникают в жизнедеятельность абсолютного большинства организаций самого широкого профиля.

Использование РТС ставит перед проектировщиками подобных систем самые противоречивые задачи. В то же время РТС подвержены широкому спектру параметрических и внешних воздействий, изменяющих не только техническое состояние системы, но и ее структуру.

Перечисленные проблемы обосновывают необходимость создания эффективных методов и средств управления динамическими характеристиками и состоянием РТС различной сложности в условиях неполной определенности.

Одним из направлений решений данной задачи является разработка единого методологического подхода к усовершенствованию систем управления РТС на основе иерархической протокольной системы организации взаимодействия ее объектов. Необходимо создать универсальную формальную модель системы протокольного взаимодействия, обеспечивающую автоматизацию процедур анализа и синтеза протоколов управления РТС.

Целью исследования является разработка общей концепции использования протоколов управления информационным обменом для организации эффективного управления взаимодействием сетевых объектов. Объектом исследований являются протокольные системы управления информационным взаимодействием. Предмет исследования - автоматные модели, их приложение к различным задачам идентификации, анализа и синтеза протоколов информационного взаимодействия.

Методы исследования: в работе используется комплексный подход, включающий как теоретические, так и экспериментальные исследования. Теоретические исследования проводились с применением методов теории конечных автоматов, регулярных событий, графов и формальных языков. Экспериментальные исследования с вязаны с моделированием на ЭВМ алгоритмов анализа и синтеза протоколов управления информационным обменом.

1.Спецификация систем протокольного взаимодействия

Спецификация объектов моделирования (в данном случае протоколы информационного взаимодействия) представляется при помощи языка таблиц событий. Табличные модели[1] обладают свойством наглядности и доступности, тем самым дают возможность взглянуть шире на задачи проектирования систем управления и моделирования.

Таблицы событий представлена в системе в виде кортежа:

$$<E,A,C,Cond,Act>.$$

Здесь $E = \{E_i\}$ –множество условий, описывающих состояния протокола;

$A = \{A_i\}$ – множество действий;

$C = \{C_i\}$ – множество векторов условий;

Cond = $||c_{ij}||$, Act = $||a_{ij}||$ - матрицы взаимосвязей множеств векторов данных и векторов действий.

Развитие систем на основе таблиц[2] событий может дать мощный аппарат для автоматизации процессов в проектировании и моделировании. Благодаря модульности, система на основе таблиц событий может быть легко модифицирована и расширена для обеспечения более широкого круга функциональных возможностей.

2.Разработка протокольной системы моделирования

Системанаосноветаблицсобытийпредставляетсобойсовокупностьподсистем[3], обеспечивающихвозможностимоделированияиуправления. Протокольная система моделирования включает в себя следующие подсистемы: пользовательский интерфейс (подсистема ввода – вывода), библиотека спецификаций, подсистема проверки и коррекции таблиц событий, подсистема генерации событий (используя вероятность появления того или иного события либо нет), подсистема инициирования взаимодействия.

Структура системы изображена на рисунке 2.

Рис.2. Структура системы

Для хранения спецификаций используются реляционные СУБД. Это предполагает то, что все таблицы событий в системе должны иметь идентичную структуру, но хранить различные данные. Таким образом таблица событий представляет собой объект, имеющий определенную структуру и хранящий определенные данные.

Внешний вид программы представлен на рисунке 3.

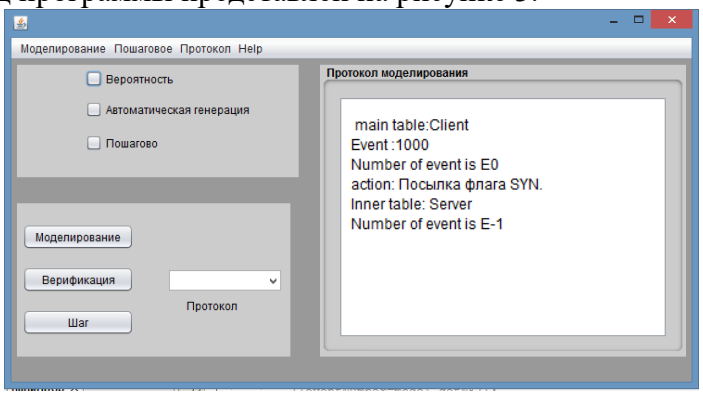


Рис.3. Пользовательский интерфейс

Подсистема проверки и коррекции таблиц событий представляют собой интерактивную систему, способствующую пользователю принимать решения об внесении изменений в моделируемую систему.

Подсистема генераций событий генерирует события для конкретной таблицы событий либо случайным образом, либо в соответствии с вероятностью возникновения той или иной ситуации. Благодаря такой системе у пользователя больше шансов обнаружить ошибки в моделируемой системе и исправить их на первых этапах проектирования.

3.Исследование протокольной системы моделирования

Исследование описанной выше системы моделирования заключается в определении реакции системы в различных ситуациях.

Условно ситуации можно разбить на 2 класса: общие и специальные.

Общие можно отнести ко всем системам, использующим таблицы состояний. Таблицы событий должны обладать свойством полноты и безизбыточности. Неполнота свидетельствует о том, что моделируемая система, представленная при помощи таблиц событий, не полностью описывает реальную систему, т.е. может возникнуть ситуация, реакция на которую не прописана. Избыточность говорит об обратном, т.е. на одну и ту же ситуацию указаны различные действия. Решение данных проблем может быть различное. К примеру, для ситуации с неполнотой, можно добавить непрописанное в таблице событие и соответствующие ему действия.

Система моделирования должна различать эти ситуации и предлагать возможные варианты решения. Например, была создана спецификация, описывающая протокол TCP, с умышленно введенными в описании ошибками и система моделирования корректно отреагировала на это и предложила возможные варианты решения (рис.4).

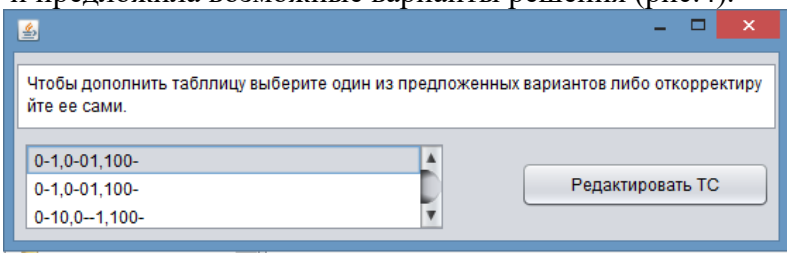


Рис.4.Реакция системы

К специальным ситуациям можно отнести ситуации, которые относятся к конкретным моделируемым системам. К примеру, реакция системы на событие, характеризующее отправку сообщения от сервера к клиенту. Возможны различные варианты реакции: сообщение не дошло до клиента (потерялось), сообщение дошло, но ответ об этом так и не был получен, сообщение дошло и мы получили ответ и т.д. Все это прописывается в протоколе моделирования (рис.3), прочитав который, можно определить на каком этапе и где именно произошел сбой или убедиться в корректности моделируемой системы.

В дальнейшем планируется выявить и другие возможные ситуации и определить реакцию системы на них. Это позволит получить практически универсальную систему для моделирования сетевых объектов информационного взаимодействия, используя спецификации, представленные при помощи языка таблиц событий, что уменьшит энтропию при разработке подобных систем.

Библиографический список

1. Хамби Э. Программирование таблиц решений/ Э. Хамби. – М.: Мир, 1976. – 86с.
2. Поспелов Д.А. Ситуационное управление: теории и практика/ Д.А. Поспелов. – М.: Наука, 1986. – 288 с.
3. Мирхалыкова Л.Т. Компьютерная система моделирования сетевых структур, представленных средствами языка таблиц событий/ Л.Т.Мирхалыкова//Материалы внутривузовской студен. науч.-техн. конф., – Севастополь: СевНТУ, 2014. – 32с.

СЕКЦИЯ 7

Web – разработка

УДК 004.4

Р.Ю. Измайлов, студент 4-го курса
Научный руководитель: ст. пр. Ю.Л. Явкун
Севастопольский государственный университет

ОБЗОР ТЕХНОЛОГИИ WEBVIEW ДЛЯ РАЗРАБОТКИ КРОСС-ПЛАТФОРМЕННЫХ ПРИЛОЖЕНИЙ НА ОСНОВЕ ВЕБ-ТЕХНОЛОГИЙ. ПРИМЕРЫ БИБЛИОТЕК

Аннотация: Широкое распространение веб-технологий, хорошая документация, готовые решения для веб-технологий, большое количество исходного кода привело к их сипользованию для широкого спектра задач. Появились специализированные библиотеки для разработки кросс-платформенных приложений.

Ключевые слова: webview, библиотека, кросс-платформенный, native-приложения, API, веб-приложения, веб-технологии.

Annotation: widespread web technologies , good documentation , turnkey solutions for web technologies , a large number of source code led to their programs included a wide range of applications. There are specialized libraries for developing cross-platform applications .

Keywords: webview, framework, cross-platform, native app, API, web app, web technologies.

В наши дни интеграция веб-приложений во все отрасли деятельности человека все растет. Благодаря чему широкую популярность получили такие веб-технологии как язык разметки HTML, каскадные таблицы стилей CSS, язык сценариев Javascript. Эти технологии просты в изучении и использовании. Популярность последних в свою очередь привела к тому, что специализация веб-разработчика стала одной из самых многочисленных специализаций в индустрии информационных технологий. Проблема заключается в том, что веб-разработчик ограничен API веб-обозревателя на стороне клиента. Веб-обозреватель выступает в роли прослойки между операционой системой и веб-приложением в целях универсальности. Но это достигается в ущерб функциональности веб-приложения в сравнении с программными продуктами, которые написаны для конкретной операционной системы и получают полный доступ к API операционной системы. Какое решение должен принять веб-разработчик для обеспечения большей функциональности? Возможны два решения:

1) Выбрать операционную систему, API которой он хочет использовать. Учить родной (native) язык с помощью которого пишутся программы для выбранной ОС (Java для Android, Swift для iOS, C# для Windows Phone и т.д.), а также изучать сопутствующие технологии;
2) Использовать библиотеку для разработки кросс-платформенных приложений и применить привычные технологии - HTML5, CSS3, JavaScript.

Недостатки первого решения очевидны: высокая стоимость, целевая аудитория ограничена пользователями выбранной ОС, высокая трудоемкость работ. Второе решени стало возможным благодаря двум составляющим:

1) Соблюдение филосовского принципа программирования: написано однажды - работает везде;
2) Технология WebView.

Первая составляющая позволяет использовать уже готовые решения, накопленный опыт, написанный код (на языках HTML5, CSS3, JavaScript), различную документацию повторно. Вторая составляющая является основой всех библиотек для разработки кросс-платформенных приложений.

WebView - технология отображения веб-контента в различных контекстах. Под различными контекстами подразумеваются программы, разработанные для различных ОС. Если говорить более доступно, то эта технология позволяет встраивать окно браузера куда

угодно. Термин WebView происходит от названия Java-класса в ОС Android, который выполняет функцию отображения веб-контента внутри Android-приложения. Так как различные ОС имеют разные инструменты (с разными наименованиями) для отображения веб-контента внутри приложений, то в качестве общего названия таких инструментов было принято наименование самого популярного - WebView.

Принцип работы библиотек для разработки кросс-платформенных приложений прост: исходные тексты программ на языках HTML5, CSS3, JavaScripts транслируются библиотекой в установочные пакеты для различных ОС (*.apt для Android, *.appx для Windows Phone и т.д). Далее приложение отображается как окно браузера в полноэкранном режиме. Приложение, полученное таким образом называется гибридным за то, что оно не является полностью native-приложением (т.к. интерфейс отображается с помощью WebView вместо native-библиотек для отображения интерфейса), но и не является полностью веб-приложением (т.к. представляют собой установочные пакеты и имеет доступ родному API устройства). Упрощенный принцип работы библиотек приведен на рисунке 1. На рисунке 2 приведена архитектура гибридного приложения.

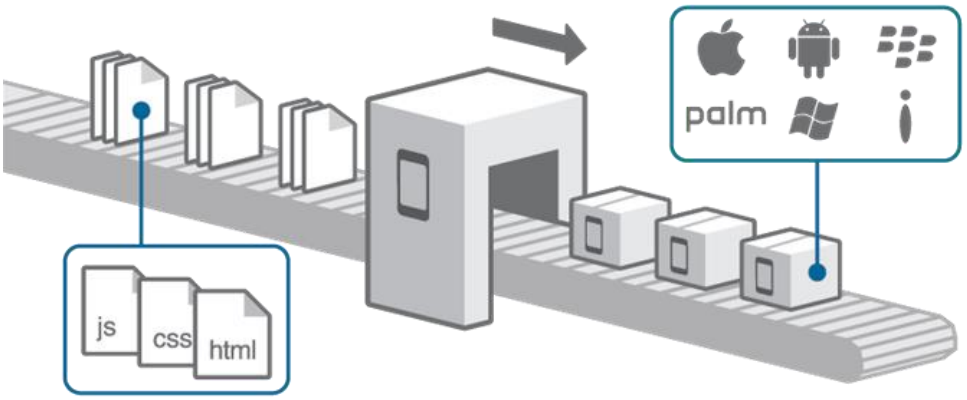


Рис.1. Упрощенный принцип работы библиотек для разработки кросс-платформенных приложений

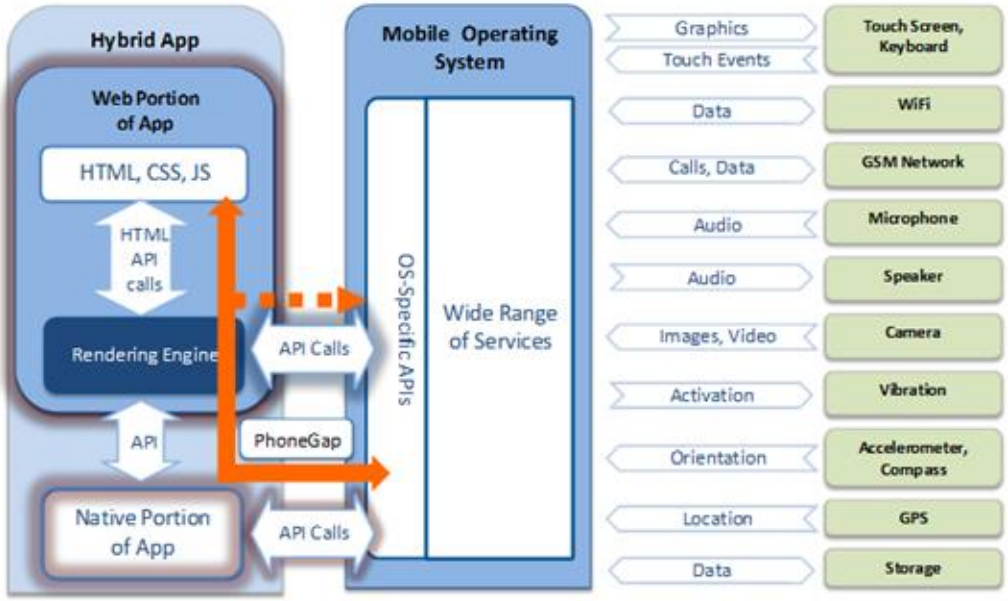


Рис.2. Архитектура гибридного приложения

Apache Cordova - одна из самых распространенных библиотек для разработки мобильных приложений при помощи HTML5, CSS3, JavaScript. Для организации отображения используется HTML5 и CSS3. Для организации логики приложения используется JavaScript. С помощью Apache Cordova можно писать мобильные приложения для большого числа мобильных ОС: Apple iOS, Bada, BlackBerry, Firefox OS, Google Android, LG webOS, Microsoft Windows Phone, Nokia Symbian OS, Tizen (SDK 2.x) и Ubuntu Touch.

Electron - библиотека разработки кросс-платформенных приложений для настольных ОС. Эта библиотека изначально разрабатывалась для написания кросс-платформенного текстового редактора Atom. Затем была выпущена компанией GitHub под открытой лицензией. Библиотека позволяет разрабатывать приложения для настольных ОС: Linux, MS Windows, Mac OS.

Так же есть множество других библиотек со схожими возможностями: Appcelerator Titanium, Kony Platform, IBM Worklight, Telerik Platform, Verivo Akula, Xamarin.

На основе открытых библиотек разрабатываются целые экосистемы для быстрой и удобной разработки. Примером такой экосистемы является Adobe PhoneGap, основанная на Apache Cordova. PhoneGap предоставляет:

1) Платную поддержку;
2) Онлайн обучение;
3) Облачные сервисы, позволяющие собрать конечное приложение без необходимости инсталляции на локальном компьютере разработчика.

Недостатком такого способа разработки приложений является слабая производительность гибридных приложений в сравнении с native-приложениями.

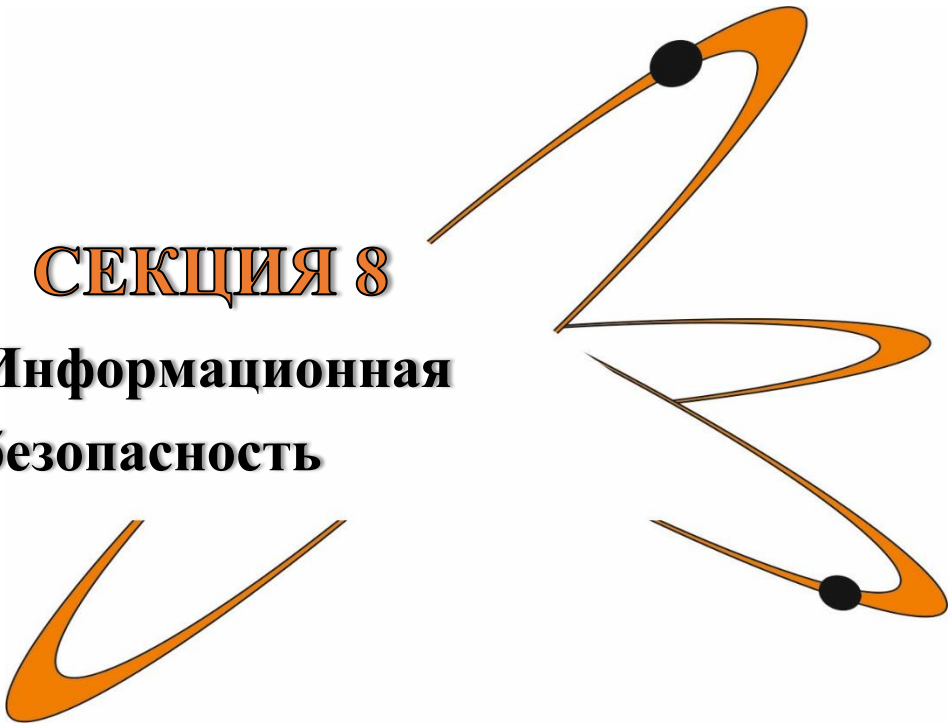
Заключение

Библиотеки для разработки кросс-платформенных приложений - наиболее удобное решение, если в приоритете стоит цена разработки, время и трудоемкость, но не рекомендуемое, если производительность критична. Дальнейшее развитие этих библиотек, возможно, пойдет по одному из двух сценариев:

1) Дальнейшее совершенствование существующих библиотек и разработка новых. Рост еще более мощных экосистем вокруг открытых библиотек;
2) Разработка и поддержка таких приложений прекратится как только веб-обозреватели смогут предоставлять более функциональный API с сильной интеграцией к аппаратными средствами устройства.

Библиографический список

1. http://electron.atom.io/
2. https://myshadesofgray.wordpress.com/2014/04/15/hybrid-applications-and-android-native-browser/
3. http://developer.telerik.com/featured/what-is-a-webview/
4. https://ru.wikipedia.org/wiki/PhoneGap



СЕКЦИЯ 8

Информационная безопасность

УДК 004.056.55
К.А.Афонин, студент 4-го курса
Научный руководитель: М.А.Лебедева, ст. преподаватель
Севастопольский государственный университет

КРИПТОГРАФИЧЕСКАЯ СИСТЕМА НА БАЗЕ НЕЙРОННОЙ СЕТИ RBF

Аннотация: рассмотрен подход к проектированию криптографической системы, основанной на использовании радиальных базисных функциях нейронной сети. Элементом системы является криптографически стойкий генератор псевдослучайных чисел. Приведено краткое описание системы и особенности ее функционирования. Изложены достоинства использования нейронных сетей в криптографии.
Ключевые слова: шифрование, нейронные сети, радиальные базисные функции, псевдослучайная последовательность, криптостойкость.
Annotation: an approach to designing cryptographic systems based on the use of Deposit radial basis functions neural network. Element of the system is a cryptographically secure pseudorandom number generator. Brief description of the system and peculiarities of its functioning. Set out the advantages of using neural networks in cryptography.
Key words: encryption, neural networks, radial basis functions, pseudo-random sequence output, the cryptographic strength.

Одним из важнейших принципов информационной безопасности является обеспечение конфиденциальности информации. Для защиты информации от несанкционированного доступа чаще всего используют симметричные и асимметричные криптографические системы. В основе симметричных систем лежат криптографические алгоритмы, в которых для шифрования и дешифрования информации используется один и тот же ключ. В современной криптографии действует принцип открытости алгоритмов, вся секретность криптографической системы основывается на секретности ключа. Криптографическая система включает процедуры шифрования E и дешифрования D, которые на основе секретного ключа выполняют преобразования над открытым текстом M и зашифрованным текстом C:

$$C = E_k(M),\ M = D_k(C) = D_k(E_k(M)) \quad (1)$$

В традиционных криптографических системах криптостойкость зависит от длины ключа k, поскольку для вскрытия методом грубой силы необходимо просмотреть $2^{|k|}$ комбинаций. Если длина ключа $|k|$=64 бит, то задача вскрытия ключа решается за несколько недель. Поэтому непрерывно идет поиск новых подходов и алгоритмов, которые обеспечивали бы надежную защиту информации. В качестве одного из таких подходов предлагается использование искусственных нейронных сетей, которые могут быть обучены задачам из области криптографии.

Искусственные нейронные сети базируются на законах работы человеческого мозга. Они представляют собой сеть элементов – искусственных нейронов, связанных между собой синоптическими соединениями. Сеть обрабатывает входную информацию и в процессе изменения своего состояния во времени формирует совокупность выходных сигналов. Сложность, возникающая благодаря нелинейности и многочисленным взаимодействиям динамик нейронных сетей можно использовать для создания криптографической системы.

Разрабатываемая криптографическая система основана на радиальных базисных функциях (RBF) нейронных сетей. Выбор основан на том, что ошибка обучения сети RBF равна нулю, поэтому восстановление происходит предельно точно.

Структура радиальной сети включает входной слой, на который подаются сигналы, скрытый слой с нейронами радиального типа и выходной слой, состоящий, как правило, из одного или нескольких линейных нейронов. Скрытых нейроны соединяются связями с весами с выходными линейными нейронами, Функция выходного нейрона сводится к взвешенному суммированию сигналов, генерируемых скрытыми нейронами [1].

Каждый элемент j скрытого слоя использует в качестве активационной функции радиальную базисную функцию типа гауссовой:

$$\varphi_j(x, c_j) = e^{-\frac{-\sum_{i=1}^{n}(x_i - c_j)^2}{2\cdot\sigma_j^2}} \quad (2)$$

где x=(x_1, x_2, …, x_n) – входной вектор.

Радиальная базисная функция (функция ядра) центрируется в точке c_j, которая определяется весовым вектором, связанным с нейроном, σ_j - параметр, от значения которого зависит ширина размаха функции.

Нейроны скрытого слоя соединены по полносвязной схеме с нейронами выходного слоя, которые осуществляют взвешенное суммирование:

$$y_k = \sum_{j=1}^{m} \omega_{j,k} \cdot \varphi(x, c_j), \qquad k = 1..m \tag{3}$$

где $\omega_{j,k}$ - элемент весовой матрицы.

Для построения криптографической системы на основе нейронной сети необходимо задать ее характеристики: количество нейронов в слое, весовые коэффициенты, функции активации, пороговые значения и т.д. Значения этих характеристик используются как составной секретный ключ и определяют алгоритмы шифрования и дешифрования. Эти значения, с одной стороны, зависят от нейронной сети, а, с другой, определяются представлением кода: его длиной, количеством одновременно кодируемых символов. Алгоритм дешифрования заключается в распознавании полученной информации. На вход построенной сети поступает зашифрованный текст, код которого обрабатывается сетью. Алгоритм шифрования основывается на поиске искаженного кода, который может распознать или восстановить используемая сеть [2].

Входной вектор RBF-сети (вектор x) представляет собой псевдослучайную последовательность, которая генерируется при помощи криптографически стойкого генератора псевдослучайных чисел. В качестве такого генератора выбран генератор Блюм-Блюм-Шуба (BBS) [3]. Параметры генератора: $N=p \cdot q$, где p,q – простые числа, сравнимые с 3 по mod 4, случайное стартовое целое число s ($1 \le s \le N-1$) так, что НОД(s,N)=1. Параметры N и s являются частью секретного ключа и должны передаваться по защищенному каналу связи. Генератор описывается формулой:

$$u_{i+1} = u_i^2 \ mod \ N \tag{4}$$

где $u_0 = s$.

Открытое сообщение представляется как целевой вектор RBF-сети (вектор y). При обучении сеть будет формировать нелинейную зависимость между входным и целевым векторами. Результатом обучения сети является зашифрованное сообщение (вектор c), которое может передаваться получателю по открытому каналу связи. В системе количество нейронов в скрытом и выходном слоях совпадают и равны m, поскольку при шифровании длина сообщения не меняется. Количество нейронов во входном слое n, в общем случае $n \neq m$. При большой длине исходного текста целесообразно разбивать его на блоки длиной m и использовать в качестве входного вектора n-элементные участки псевдослучайной последовательности. Получение криптограммы равнозначно решению системы m линейных уравнений (3) относительно φ и определения весовых значений (2).

Для дешифрации по криптограмме восстанавливается сеть RBF. Полученная сеть и секретный ключ используются для генерации входного вектора, который подается на вход сети RBF, вследствии чего происходит получение исходного сообщение на выходе.

Поскольку в рассмотренной системе секретной ключевой информацией являются как параметры нейронной сети, так и параметры генератора псевдослучайных чисел, то по сравнению с традиционной криптографией пространство ключей возрастает/ что приводит к увеличению криптостойкости системы.

Библиографический список

1. Хайкин С. Нейронные сети: полный курс, 2-е издание.: Пер. с англ. – М.: Издательский дом «Вильямс», 2006. – 1104 с.:ил.
2. Червяков Н. И. Применение искусственных нейронных сетей и системы остаточных классов в криптографии / Н. И. Червяков, А. И. Галушкин, А. А. Евдокименко, А. В. Лавриненко — Москва: Физматлит, 2012. — 280 с.: ил.
3. Шнайер Б. Прикладная криптография./ Б. Шнайер.—СПб.:Питер, 2003. — 368 с.

УДК 004.056
Б.Г. Бабенко, студент 4-го курса
Научный руководитель: Н. Е. Губенко, доцент кафедры КМД
Донецкий национальный технический университет (г. Донецк)

АНАЛИЗ УГРОЗ И УЯЗВИМОСТЕЙ ДЛЯ ОЦЕНКИ ЗАЩИЩЕННОСТИ АВТОМАТИЗИРОВАННОЙ СИСТЕМЫ ЭКОЛОГИЧЕСКОГО МОНИТОРИНГА «АКИАМ»

Аннотация: В данной работе проведенанализ угроз и уязвимостей для оценки защищенности автоматизированной системы экологического мониторинга «АКИАМ». Приведены действующие методы и приемы, позволяющие обеспечить безопасность автоматизированной системы.
Ключевые слова: Анализ угроз, уязвимости, оценка защищенности, автоматизированная система.
Annotation: In this paper, an analysis of threats and vulnerabilities to assess the security of the automated environmental monitoring system «AKIAM». Presents current methods and techniques to assure the safety of the automated system.
Keywords: An analysis of threats, vulnerabilities, security assessment, automated system.

Целью данной работы является оценка стойкости информационной защиты системы экологического мониторинга «АКИАМ».

«АКИАМ» -это автоматизированная система, предназначенная, для сбора и обработкиэкологической информации, касающейся окружающей среды и природных ресурсов города Донецка для повышения эффективности управления природоохранной деятельностью на городскомуровне.

Автоматизированная система позволяет:

• контролировать показатели загрязнения атмосферного воздуха и поверхностных вод по ингредиентам, для которых имеются серийно выпускаемые анализаторы и датчики со стандартным аналоговым или цифровым выходом;

•контролировать метеорологические параметры: направление и скорость ветра, температуру, влажность и давление атмосферного воздуха, интенсивность солнечной радиации и т.д.;

•вести базы данных параметров природных и техногенных объектов, а также показателей загрязнения сред и метеопараметров на городском уровне;

•представлять тематические карты в геоинформационной системе;

•повысить эффективность экологического контроля за счет проведения оперативных наблюдений и автоматизации процесса регистрации информации, ее передачи, обработки, анализа и хранения.

«АКИАМ» представляет собой сложную систему, состоящую из большого числа компонентов различной степени автономности, которые связаны между собой и обмениваются данными. Практически каждый компонент может подвергнуться внешнему воздействию или выйти из строя. Компоненты АС можно разбить на следующие группы:

• аппаратные средства — компьютеры и их составные части (процессоры, мониторы, терминалы, периферийные устройства — дисководы, принтеры, контроллеры, кабели, линии связи и т. д.);

• программное обеспечение — приобретенные программы, исходные, объектные, загрузочные модули; ОС и системные программы (компиляторы, компоновщики и др.), утилиты, диагностические программы и т. д.;

• данные — хранимые временно и постоянно, на магнитных носителях, печатные, архивы, системные журналы и т. д.;

• персонал — обслуживающий персонал и пользователи.

Одной из особенностей обеспечения информационной безопасности в подобных системах является то, что таким абстрактным понятиям, как информация, объекты и субъекты системы, соответствуют физические представления в компьютерной среде:

• для представления информации — машинные носители информации в виде внешних устройств компьютерных систем (терминалов, печатающих устройств, различных накопителей, линий и каналов связи), оперативной памяти, файлов, записей и т. д.;

• объектам системы — пассивные компоненты системы, хранящие, принимающие или передающие информацию. Доступ к объекту означает доступ к содержащейся в нем информации;

• субъектам системы — активные компоненты системы, которые могут стать причиной потока информации от объекта к субъекту или изменения состояния системы. В качестве субъектов выступают пользователи, активные программы и процессы.

«АКИАМ» может работать в двух режимах: в режиме центрального сервера и в режиме локального сервера. С точки зрения защиты информации для нас особенно важен режим центрального сервера, так как именно в этом режиме информационная система наиболее подвержена атакам. Доступ к центральному серверу осуществляется через сеть Интернет по адресу http://www.akiam.org.ua.

Структура обмена информации в информационной системе экологического мониторинга «АКИАМ» представлена на рисунке 1.

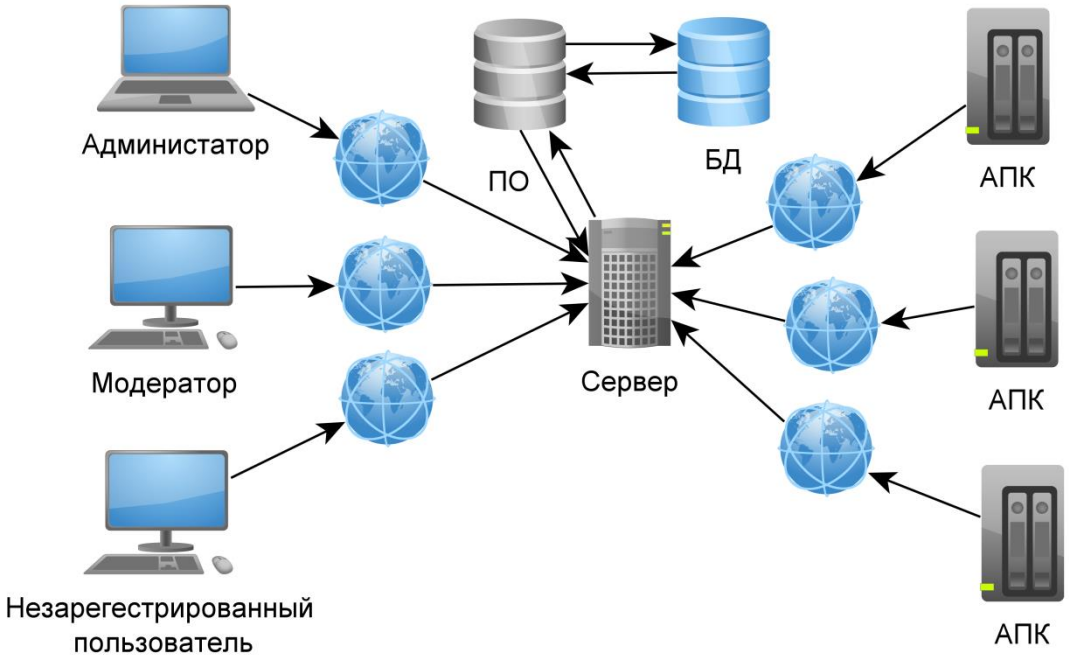


Рис 1. Структура автоматизированной системы «АКИАМ»

Функциональные модули системы написаны на языке PHP с использованием базы данных MySQL.Скрипт и база данных хранятся на удаленном сервере. Доступ к ним осуществляется через протокол http.Для регулирования прав доступа была использована дискреционной-ролевая модель политики безопасности. Эта модель позволяет создать гибкую схему безопасности, однако этот метод сложен в администрировании, потому что с программной точки зрения его реализация очень проста, но при достаточно большом количестве объектов и субъектов система становится практически неуправляемой.

В системе выделены три роли – администратор, модератор и не зарегистрированный пользователь. Администратор имеет возможность добавлять пользователей с существующими ролями и при помощи матрицы доступа регулировать их права.

Схема разделения АС «АКИАМ» на объекты защиты показан на рисунке 2.

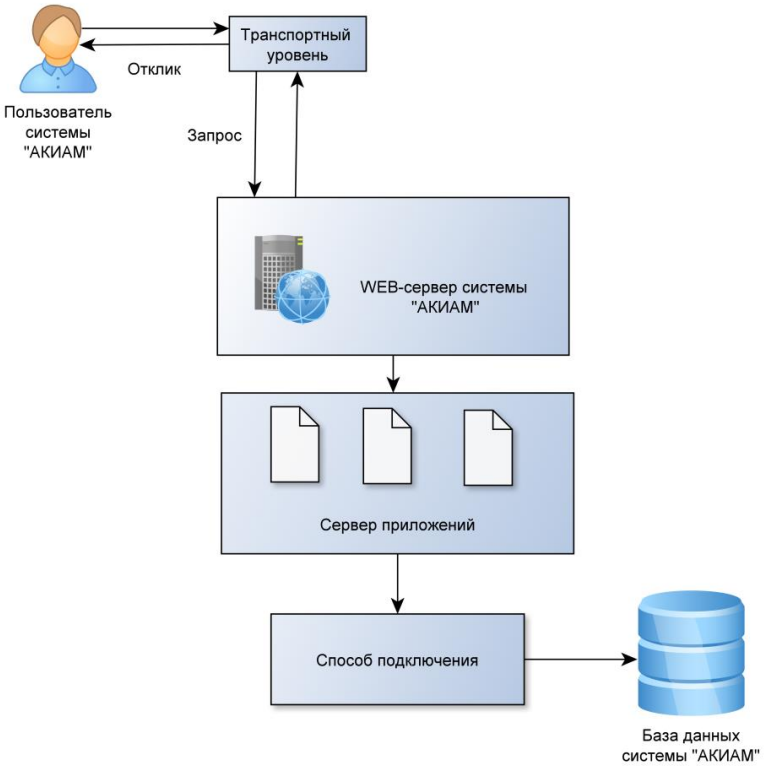


Рис 2. Схема разделения АС «АКИАМ» на объекты защиты

Для создания системы защиты АС «АКИАМ» использованы следующие механизмы и приёмы:

- для объекта «WEB-сервер» - это специальные настройки ограничений не доступ к контейнерам контента (каталоги и файлы на файловой системе, разделы сайта);

-для объекта «Сервер приложений» - это механизм управления доступом (дискретно ролевая модель политики безопасности) и механизм авторизации при помощи логина и пароля;

- для объекта «База данных» - это механизм авторизации;

- для объекта «Пользователь» - это антивирусные программы.

Реализация Внедрение предлагаемых приёмов и методов позволит повысить уровень защищенности АС «АКИАМ» от угроз нарушения конфиденциальности и целостности.

Библиографический список

1. Губенко Н.Є., Хімка С.С. Розробкаімітаційноїмоделізахистуінформації для автоматизованоїсистеми «Екологічний паспорт регіонівУкраїни»//Системнийаналіз та інформаційнітехнології у науках про природу та суспільство. — 2011. — №1. — С. 185-188.
2. Основные понятия защиты информации и информационной безопасности / Интернет-ресурс. – Режим доступа: URL: http://ypn.ru/102/introduction-to-information-protection-and-information-security/– Загл. с экрана.
3 Комплексный подход к организации системы защиты информации на предприятии: основные вопросы и технологии/ Интернет-ресурс. – Режим доступа: URL: http://www.epam-group.ru/about/news-and-events/in-the-news/2009/kompleksnyy-podhod-k-organizacii-sistemy-zaschity-informacii-na-predpriyatii-osnovnye-voprosy-i-tehnologii/– Загл. с экрана.
4.Развитие автоматизированной системы экологического мониторинга г.донецка/ Интернет-ресурс. – Режим доступа: URL: http://studydoc.ru/doc/580175/razvit_sist_monit/– Загл. с экрана.
5.Харечкин П. В. Методология построения функционально-ролевой модели управления доступом на основе среды радикалов // Молодой ученый. — 2009. — №12. — С. 22-27.

УДК 004.056
А. Д. Багимова, студентка 4-го курса
Научный руководитель: Н. Е. Губенко, доцент
Донецкий национальный технический университет (г. Донецк)

ЗАЩИТА WEB-СИСТЕМЫ ДЛЯ ОЦЕНКИ И РАНЖИРОВАНИЯ СТРАН МИРА ПО ПОКАЗАТЕЛЯМ ИХ РАЗВИТИЯ

Аннотация: В данной статье рассмотрены проблемы защиты web-системы для оценки и ранжирования стран мира по показателям их развития. Изложена актуальность проблемы защиты информации и структура web-системы. Проанализированы существующие угрозы безопасности и выявлены пути решения вопросов защиты.
Ключевые слова: web-система, показатели, база данных, сервер, функциональные модули, ранжирование, пользователь, методы, технологии..
Annotation: This article describes the problems of protection of web-system for the evaluation and ranking of countries in terms of their development of the world. Presented relevance of information security issues and the structure of web-systems. Analyzes the existing security threats and identifies ways to address protection issues.
Key words: web-system, indicators, database, server, functional modules, ranging, user, methods, technologies.

Проблема защиты информации на настоящем этапе развития информационных телекоммуникационных технологий является безусловно актуальной и определяется рядом объективных и субъективных факторов, среди которых можно отметить наиболее существенные:

• обострение противоречий между объективно существующими потребностями общества в расширении свободного обмена информацией и чрезмерными или наоборот недостаточными ограничениями на ее распространение и использование;

• расширение сферы использования компьютеров, многообразием и повсеместным распространением информационно-управляющих систем, высокими темпами роста и развития парка средств вычислительной техники и связи;

• повсеместное использование локальных и глобальных сетевых технологий людьми и организациями, не подготовленными в сфере защиты информации;

• концентрация больших объемов информации различного назначения и принадлежности на электронных носителях;

• отношение к информации, как к товару, переход к рыночным отношениям в области предоставления информационных услуг с присущей им конкуренцией и промышленным шпионажем

• многообразием видов угроз и возникновением новых возможных каналов несанкционированного доступа к информации

• увеличение потерь (ущерба) от уничтожения, фальсификации, разглашения или незаконного тиражирования информации (возрастанием уязвимости различных затрагиваемых субъектов)

• развитие рыночных отношений (в области разработки, поставки, обслуживания вычислительной техники, разработки программных средств, в том числе средств защиты).

Целью данной работы является защита web-системы для оценки и ранжирования стран мира по показателям их развития. Данная система реализована Web-приложениями, ее структура представлена на рисунке 1.

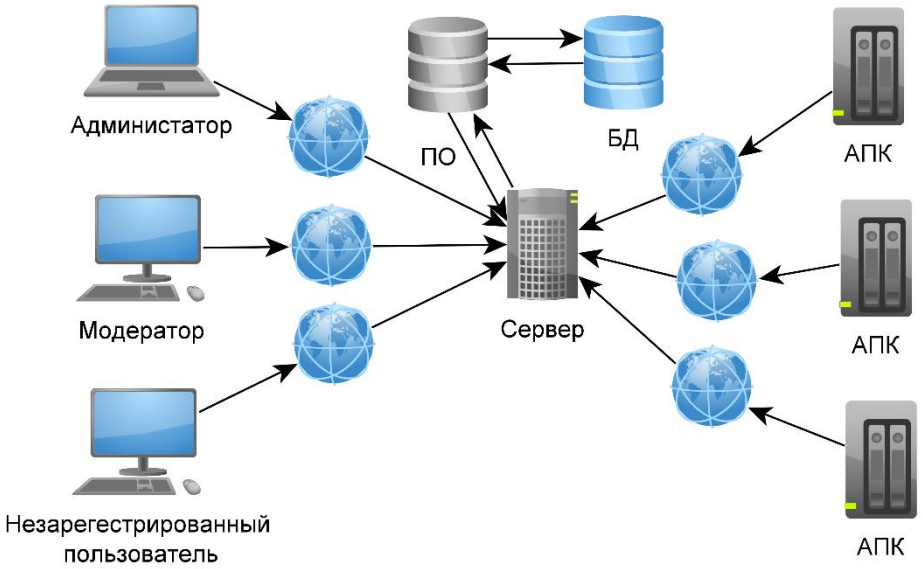


Рис. 1. Структура системы

Функциональные модули web-системы для оценки и ранжирования стран мира по показателям их развития разработаны с помощью языка программирования PHP и базы данных MySQL. Скрипты и база данных сохраняется на отдалённом сервере, доступ к которому осуществляется через протокол http. Защита от угрозы нарушения конфиденциальности осуществляется с помощью традиционной авторизации. Для регулирования прав доступа используется модель политики безопасности. Система имеет три роли – администратор, модератор и незарегистрированный пользователь. Администратор имеет возможность добавлять пользователей с существующими ролями.

Угроза информационной безопасности web-системы — это возможность воздействия на информацию, обрабатываемую в системе, приводящая к искажению, уничтожению, копированию, блокированию доступа к информации, а также возможность воздействия на компоненты информационной системы, приводящая к нарушению секретности, утрате, уничтожению или сбою функционирования носителя. Реализация любого из этих факторов может привести к нарушению конфиденциальности, целостности информации или средства управления программно-аппаратным комплексом системы и, как следствие, к потере основных функциональных свойств системы.

Угроза нарушения конфиденциальности данных включает в себя любое умышленное или случайное раскрытие информации, хранящейся в вычислительной системе или передаваемой из одной системы в другую. К нарушению конфиденциальности ведет как умышленное действие, направленное на реализацию несанкционированного доступа к данным, так и случайная ошибка программного или неквалифицированного действия оператора, приведшая к передаче по открытым каналам связи незащищенной конфиденциальной информации.

Угроза нарушения целостности включает в себя любое умышленное или случайное изменение информации, обрабатываемой в информационной системе или вводимой из первичного источника данных. К нарушению целостности данных может привести как преднамеренное деструктивное действие некоторого лица, изменяющего данные для достижения собственных целей, так и случайная ошибка программного или аппаратного обеспечения, приведшая к безвозвратному разрушению данных.

Первый шаг в анализе угроз — их идентификация. Рассматриваемые виды угроз следует выбирать исходя из соображений здравого смысла, но в пределах выбранных видов провести максимально подробный анализ.

Необходимо не только провести работу по выявлению и анализу самих угроз, но и изучить и описать источники возникновения выявленных угроз. Такой подход поможет в выборе комплекса средств защиты. Например, нелегальный вход в систему может стать следствием воспроизведения начального диалога, подбора пароля или подключения к сети неавторизованного оборудования. Очевидно, для противодействия каждому из перечисленных способов нелегального входа нужны свои механизмы безопасности.

Существуют различные методы защиты баз данных. Они условно делятся на две группы: основные и дополнительные

К основным средствам защиты относится:

- защита паролем;
- шифрование данных и программ;
- разграничение прав доступа к объектам базы данных;

К дополнительным средствам защиты БД можно отнести такие, которые нельзя прямо отнести к средствам защиты, но которые непосредственно влияют на безопасность данных. Их составляют следующие средства:

- встроенные средства контроля значений данных в соответствие с типами;
- повышение доверенности вводимых данных;
- обеспечение целостности связей таблиц;
- организация совместного использования объектов БД в сети.

В разрабатываемом ресурсе используется защита паролем (для регулирования прав доступа рядового пользователя и администратора) и разграничение прав доступа к объектам базы данных.

Концепции обеспечения безопасности базы данных нашей web-системы исключительно просты. Система поддерживает два фундаментальных принципа: проверку полномочий и проверку подлинности (аутентификацию).

Проверка полномочий основана на том, что каждому пользователю или процессу информационной системы соответствует набор действий, которые он может выполнять по отношению к определенным объектам. Проверка подлинности означает достоверное подтверждение того, что пользователь или процесс, пытающийся выполнить санкционированное действие, действительно тот, за кого он себя выдает.

Система назначения полномочий имеет в некотором роде иерархический характер. Самыми высокими правами и полномочиями обладает системный администратор или администратор сервера БД. Традиционно только этот тип пользователей может создавать других пользователей и наделять их определенными полномочиями.

СУБД в своих системных каталогах хранит как описание самих пользователей, так и описание их привилегий по отношению ко всем объектам.

В нашей СУБД вводится следующий уровень иерархии пользователей — это администратор БД. В системной СУБД один сервер MS SQL Server может управляет множеством СУБД. В стандарте SQL не определена команда создания пользователя, но практически во всех коммерческих СУБД создать пользователя можно не только в интерактивном режиме, но и программно с использованием специальных хранимых процедур. Однако для выполнения этой операции пользователь должен иметь право на запуск соответствующей системной процедуры.

Защита паролем представляет собой простой и эффективный способ защиты БД от несанкционированного доступа. Пароли устанавливаются пользователями или администраторами БД. Учет и хранение паролей выполняется самой СУБД. Обычно, пароли хранятся в определенных системных файлах СУБД в зашифрованном виде. После ввода пароля пользователю СУБД предоставляются все возможности по работе с БД.

Парольная защита является достаточно слабым средством, особенно если пароль не шифруется. Основной ее недостаток состоит в том, что все пользователи, использующие одинаковый пароль, с точки зрения вычислительной системы неразличимы. Неудобство

парольной системы для пользователя состоит в том, что пароль надо запоминать или записать. При небрежном отношении к записям пароль может стать достоянием других.

В целях контроля использования основных ресурсов СУБД во многих системах имеются средства установления прав доступа к объектам БД. Права доступа определяют возможные действия над объектами. Владелец объекта, а также администратор БД имеют все права. Остальные пользователи к разным объектам могут иметь различные уровни доступа. Разрешение на доступ к конкретным объектам базы данных сохраняется в файле рабочей группы.

Наибольший эффект с точки зрения защищенности и производительности системы достигается только при реализации элементов безопасности во время проектирования и разработки системы в целом и каждого отдельного её элемента, поэтому дополнительно необходимо решить вопрос преобразования форматов существующих и накопленных в мире баз данных показателей развития разных стран и регионов в формат My SQL без нарушения их достоверности и целостности.

Библиографический список

1. Защита информации в базах данных [Электронный ресурс]: – Режим доступа: http://bourabai.ru/dbt/dbms/13.htm. (дата обращения: 10.02.2016).
2. Особенности защиты информации в базах данных [Электронный ресурс]: – Режим доступа: http://sumk.ulstu.ru/docs/mszki/Zavgorodnii/11.7.html (дата обращения: 14.02.2016).
3. Герасименко В.А., Малюк А.А. Основы защиты информации / В.А. Герасименко - М.: Известия, 1997.
4. Актуальность проблемы обеспечения безопасности информационных технологий [Электронный ресурс]: – Режим доступа: http://asher.ru/security/book/itsi/01. (дата обращения: 15.02.2016).

УДК 681.8
В.В. Бойко, студент 4-го курса
Научный руководитель: Н.Е. Губенко, канд. техн. наук, доцент.
Донецкий национальный технический университет (г. Донецк)

РАЗРАБОТКА ЗАЩИЩЁННОГО ВЕБ-САЙТА ЖУРНАЛА «САИТ».

Аннотация: В данной статье рассматривается проблема защиты веб-сайта журнала. Проанализированы основне типы атак через сеть интернет и методы борьбы с ними. На основе проведеного анали выделены основные методы защиты веб-сайта журнала от глобальных и локальных атак.
Ключевые слова: веб-сайт, локальная атака, глобальная атака, сервер, безопасный код, антивирус.
Annotation: This article discusses the problem of protecting the magazine's Web site. It analyzes the main types of attacks via the Internet and methods of dealing with them. On the basis of anali identified the main methods of protection magazine's Web site on the global and local attacks.
Key words: website, local attack, global attack, a server, a secure code, anti-virus.

По данным антивирусной сетки KSN за 2014 год было осуществлено 6 167 233 068 вредносных атак на компьютеры и мобильные устройства. Для проведения атак через интернет злоумышленники воспользовались 9 766 119 уникальными хостами. Следует учитывать что эти данные справедливы только для пользователей KNS. На самом же деле количество атак приблизительно в 6 раза больше. Поэтому цель работы – анализ и использование современных методов защиты при разработке веб-сайта научно-технического журнала «Системный анализ и информационные технологии в науках про природу и общество» (САИТ) является актуальной.

Разрабатываемый веб-ресурс предназначен для публикации научных работ и защиты авторских прав. Для достижения данной цели были рассмотрены основные типы атак через интернет и предложены методы для защиты от этих типов атак. В настоящее время количество атак, нацеленных на веб-серверы возрастает по закону Мура, причём их географическое расположение никак не влияет на безопасность и связанно с их открытостью к пользователям. Зачастую серверы рассчитаны на обмен информации с пользователями. Злоумышленник может вмешаться в работу такой системы изменив начальную функциональность, однако существуют методы, которые позволяют уменьшить шанс быть атакованным.

Атаки на веб-серверы можно условно разделить на две категории:
- локальные атаки, которые обычно направлены на кражу информации или перехват управления на отдельном веб-сервере:
- глобальные атаки, которые обычно направлены на несколько веб-сайтов и ставят своей целью заражение всех их посетителей:

С помощью локальных атак злоумышленник может получить доступ к базе данных публикаций журнала, и нарушить их достоверность. С помощью глобальных атак можно заразить публикации вредоносным программным обеспечением (ПО) и заблокировать доступ к ресурсу. Поэтому для создания и использования безопасного веб-сайта журнала следует максимально защитить сервер, на котором он размещается.

Планирование является неотъемлемой частью системы безопасности, поскольку оно позволяет сократить ущерб от вирусов, шпионского ПО и других вредоносных программ. Большинство атак можно предотвратить, построив правильную структуру веб-ресурса, которая позволит выстроить более надёжную защиту и выявить успешную атаку, если такая будет. К сожалению, на данный момент нет гарантии, что атака на сайт не удастся, как бы ни была выстроена защита.

Веб-сервер журнала сформирован несколькими слоями ПО, каждый из которых подвержен разным видам атак. На рис. 1 представлена основная структура построения ресурса.

Основой любого сервера является операционная система. Обеспечить её безопасность можно, вовремя устанавливая последние обновления системы безопасности. Многие системы безопасности устанавливают обновления автоматически.

Злоумышленники склонны автоматизировать свои атаки, перебирая один сервер за другим в поисках сервера, где не было установлено обновление. В связи с этим важно следить

за тем, чтобы обновления устанавливались своевременно и надлежащим образом; любой сервер, на котором установлены устаревшие версии обновлений, может подвергнуться атаке.

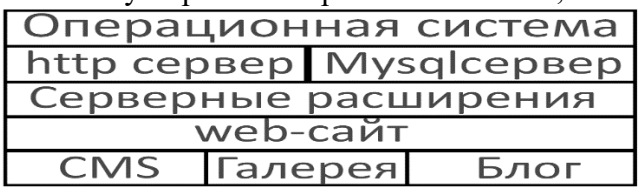


Рис.1. Структура веб-ресурса журнала.

Следует обновлять систему безопасности на сервере, и всё ПО работающее на веб-сервере. Любое ПО, не относящееся к необходимым компонентам (например, DNS-сервер либо средства удаленного администрирования наподобие VNC или служб удаленных рабочих столов), следует отключить или удалить. Если средства удаленного администрирования все же необходимы, следить за тем, чтобы не использовались пароли по умолчанию или пароли, которые можно легко угадать. Это замечание относится не только к средствам удаленного администрирования, но и к учетным записям пользователей, коммутаторам и маршрутизаторам.

Антивирусное ПО является одним из важнейших критериев. Его использование является обязательным требованием для веб-сервера журнала. Антивирусное ПО становится одним из самых эффективных способов защиты от угроз безопасности. Когда веб-сервер становится целью атаки, злоумышленник без промедления пытается загрузить инструменты взлома или вредоносное ПО, чтобы успеть использовать уязвимость системы безопасности до того, как она будет закрыта. При отсутствии качественного антивирусного пакета уязвимость системы безопасности может долгое время оставаться незамеченной.

Следующий по важности компонент программного обеспечения — сам HTTP-сервер, лучшими альтернативами здесь являются IIS и Apache.

Для защиты сайта от локальных атак следует использовать безопасный код. Разрабатывать безопасный код не всегда так просто, как кажется. Т.к. требуются глубоких знания о конкретных проблемах безопасности. Выделим несколько правил, которые позволяют снизить риск успешных атак на систему:

1. Глобальные переменные всегда следует отключать, поскольку их можно намеренно инициализировать с помощью подложного запроса GET или POST.
2. Сообщения об ошибках следует отключить; вместо них следует использовать запись сведений об ошибках в лог-файл, поскольку подобная информация может дать злоумышленникам возможность спровоцировать аналогичную проблему и использовать ее для поиска других уязвимостей.
3. Не следует считать надежными данные, поступающие от пользователей; для удаления специальных символов SQL и escape-последовательностей необходимо использовать функции фильтрации.

Для базовой защиты веб-ресурса от таких атак, как глобальные и локальные не следует устанавливать лишние компоненты, так как любой компонент несет с собой отдельную угрозу, и чем их больше, тем выше суммарный риск. Для успешной атаки достаточно единственной бреши в системе безопасности. Своевременные обновления системы безопасности для операционной системы и приложений поддерживают стабильную работу системы.

Библиографический список

1. Статья "Обеспечение безопасности веб-сайтов" [Электронный ресурс]. - Режим доступа: https://yandex.ru/support/webmaster/protecting-sites/intro.xml
2. Примеры атак Apache, [Электронный ресурс]. - Режим доступа: http://www.ossec.net/wiki/index.php/Apache_attack_samples
3. Kaspersky Security Bulletin 2014. Основная статистика за 2014 год [Электронный ресурс]. - Режим доступа: https://securelist.ru/analysis/ksb/24580/kaspersky-security-bulletin-2014-osnovnaya-statistika-za-2014-god/

УДК 681.3
Р.Г.Вихров, студент 1-го курса магистратуры
Научный руководитель: С.Н. Фисун, канд. техн. наук, доцент
Севастопольский государственный университет

ПРОГРАММНАЯ СИСТЕМА ДЛЯ ЗАЩИТЫ ИНФОРМАЦИИ

Аннотация: В статье раскрывается актуальность проблемы защиты информации в компьютерных системах. Рассматриваются наиболее распространённые программные технологии защиты антивирусных программ, их достоинства и недостатки. Дано краткое описание работы подсистем разработанной программной системы для защиты информации и дальнейшие пути их развития.
Ключевые слова: информационная безопасность, защита информации, антивирус, шифрование, сигнатурное сканирование.
Annotation:
Key words: information security, anti-virus, encryption, signature-based scanning.

Современное развитие мировой экономики характеризуется все большей зависимостью рынка от объема информационных потоков. Информация становится товаром, который можно приобрести, продать, обменять. При этом стоимость информации часто в сотни раз превосходит стоимость компьютерных систем, в которых она хранится [1].

В странах, где высок уровень компьютеризации, проблема борьбы с компьютерной преступностью уже довольно давно стала одной из первостепенных. Например, в США ущерб от компьютерных преступлений составляет ежегодно около 5 млрд долларов, во Франции эти потери доходят до 1 млрд франков в год, а в Германии при помощи компьютеров преступники каждый год ухитряются похищать около 4 млрд марок. И число подобных преступлений каждый год увеличивается на 30-40%[2, 3].

Согласно последним исследованиям, безопасность — это самая серьезная проблема, с которой столкнулись предприятия малого и среднего бизнеса. А проблемы безопасности, в свою очередь, негативно влияют на прибыльность сделок и удовлетворенность клиентов [4].

На рынке защиты информации предлагается много отдельных инженерно-технических, программно-аппаратных, криптографических средств защиты информации. В литературе также встречается описание множества разнообразных методов и средств защиты данных. Однако для того, чтобы создать на предприятии условия эффективной защиты информации, необходимо объединить отдельные средства защиты в систему.

Проанализировав современные системы защиты информации, используемые в них методы, их стоимость и трудоёмкость реализации для разработки программной системы были выбраны следующие средства антивирусной защиты:

- сигнатурный метод сканирования;
- контроль целостности данных;
- криптографическая защита;
- автозапуск съёмных носителей информации.

При поиске и уничтожении известных вирусов наиболее распространенным является сигнатурный метод сканирования. Данный метод заключается в выявлении компьютерных вирусов по их уникальному фрагменту программного кода (сигнатуре, программному штамму). Для этого создается некоторая база данных сигнатур известных компьютерных вирусов. Обнаружение вирусов осуществляется путем сравнения данных памяти компьютера с фиксированными кодами базы данных сканирования [5].

В программной системе для вычисления сигнатуры вируса (как и в большинстве антивирусных систем) используется хэш-функция, которая необратимым образом преобразовывает входные данные. На выходе хэш-функции — хэш-сумма файла. Хэш-сумма не меняется при перемещении или переименовании файла. Однако при внесении изменений в файл хэш меняется. Таким образом, можно сказать, что хэш-сумма является уникальной характеристикой файла и точно идентифицирует файл, поскольку в файл нельзя внести изменения так, чтобы хэш-сумма осталась неизменной.

К недостаткам сигнатурного сканирования следует отнести то, что оно позволяет обнаружить только те вирусы, которые уже изучены и для них определена сигнатура. Для эффективной работы сигнатурных сканеров необходимо оперативно пополнять базу данных сканирования. Однако с увеличением объема базы данных сканирования и числа различных типов искомых вирусов снижается скорость антивирусной проверки. Кроме того, разработка сигнатур — ручной процесс, требующий кропотливого анализа вирусов. Антивирусные компании вынуждены выпускать большое количество сигнатур для всех вариантов одного и того же вируса, и если бы не закон Мура, ни один современный компьютер уже не смог бы закончить сканирование большого числа файлов с такой массой сигнатур в разумное время. Так, ещё в марте 2006 года сканеру Norton Antivirus было известно около 72 131 вирусов, а база программы содержала порядка 400 000 сигнатур.

Основным достоинством сканирования на основе сигнатур имеет является возможность определять конкретную атаку с высокой точностью и малой долей ложных вызовов [6].

Выявление и ликвидация неизвестных вирусов необходимы для защиты от вирусов, пропущенных на первом уровне антивирусной защиты. Наиболее эффективным методом является контроль целостности системы (обнаружение изменений). Данный метод заключается в проверке и сравнении текущих параметров вычислительной системы с эталонными параметрами, соответствующими ее незараженному состоянию. Контроль целостности обеспечивает защищенность информационного ресурса от несанкционированных модификации и удаления в результате различного рода нелегитимных воздействий, сбоев и отказов системы и среды.

Работа модуля контроля целостности файлов состоит из двух этапов: фиксирование эталонных характеристик вычислительной системы (в основном диска) и периодическое сравнение их с текущими характеристиками. Обычно контролируемыми характеристиками являются контрольная сумма (CRC), длина, время, атрибут «только для чтения» файлов.

В разработанной программной системе в качестве эталонной характеристики файла является CRC-сумма. Суть проверки в том, что перво на перво создаётся база контрольных сумм указанных файлов. В случае повторного сканирования текущие значения контрольных сумм сравниваются со значениями из базы, если обнаруживается несоответствие контрольных сумм, то возникает подозрение на вмешательство в структуру проверяемого файла.

К недостатку данного метода можно отнести создаваемые ими различные неудобства и трудности в работе пользователя. Например, многие изменения параметров системы вызваны не вирусами, а работой системных программ или действиями пользователя-программиста. По этой причине CRC-сканеры не используют для контроля зараженности текстовых файлов, которые постоянно меняются.

CRC-сканеры обеспечивают высокий уровень выявления неизвестных компьютерных вирусов, однако они не всегда обеспечивают корректное лечение зараженных файлов[7].

Криптография играет весьма существенную роль в обеспечении безопасности. Замысел криптографического модуля заключается в том, чтобы закодировать файл, превратив его в зашифрованный текст. Следует подчеркнуть, что никакие аппаратные, программные и любые другие решения не смогут гарантировать абсолютную надежность и безопасность данных в компьютерных системах. В то же время свести риск потерь к минимуму возможно, но лишь при комплексном подходе к вопросам безопасности.

Шифрование — это способ повышения безопасности сообщения или файла, при котором их содержимое преобразуется так, что оно может быть прочитано только пользователем, обладающим соответствующим ключом шифрования для расшифровки содержимого. Например, при совершении покупки в Интернете данные сделки (такие как адрес, телефон, номер кредитной карты) обычно зашифровываются в целях безопасности.

Существуют две разновидности алгоритмов шифрования использованием ключей — симметричные и с открытым ключом (ассиметричные). Симметричным называют такой криптографический алгоритм, в котором ключ, используемый для шифрования сообщений,

может быть получен из ключа расшифрования и наоборот. В большинстве симметричных алгоритов применяют всего один ключ. Симметричные алгоритмы бывают двух видов. Одни из них обрабатывают открытый текст побитно и называются потоковыми алгоритмами. Согласно другим, открытый текст разбивается на блоки, состоящие из нескольких бит. Такие алгоритмы называются блочными. В разрабатываемой программной системе будет использован симметричный блочный алгоритм «Blowfish».

Подсистема криптографической защиты состоит из двух основных функций: шифрования и расшифрования. Перед тем, как приступать к шифрованию, необходимо убедиться в том, что введённый пользователем пароль является правильным.

Нередко заражение компьютера вирусом происходит посредством функции автозапуска съёмных носителей информации. За функцию автозапуска отвечает файл «autorun.inf», который находиться в корневом каталоге flash-накопителей, внешних HDD (переносных жестких дисков), DVD и CD-дисков, карт памяти и прочих внешних носителей информации.

В файле «autorun.inf» прописано, какое приложение необходимо запустить на выполнение в автоматическом режиме. Этим свойством пользуются злоумышленники, прописывающие в «autorun.inf» код запуска вредоносного программного обеспечения. Таким образом, при подключении заражённого накопителя информации ОС Windows запускает файл автозапуска на исполнение, в результате чего происходит заражение компьютера.

Решением проблемы является отключение функции автозапуска. Данная функция встроена в большинство современных антивирусных средств, поэтому было принято решения реализовать контроль автозапуском в разрабатываемой программной системе.

Интерфейс программной системы обладает интуитивной простотой и непергружен графическими элементами. Он включает четыре независимых вкладок, которые изображены на рисунке 1.

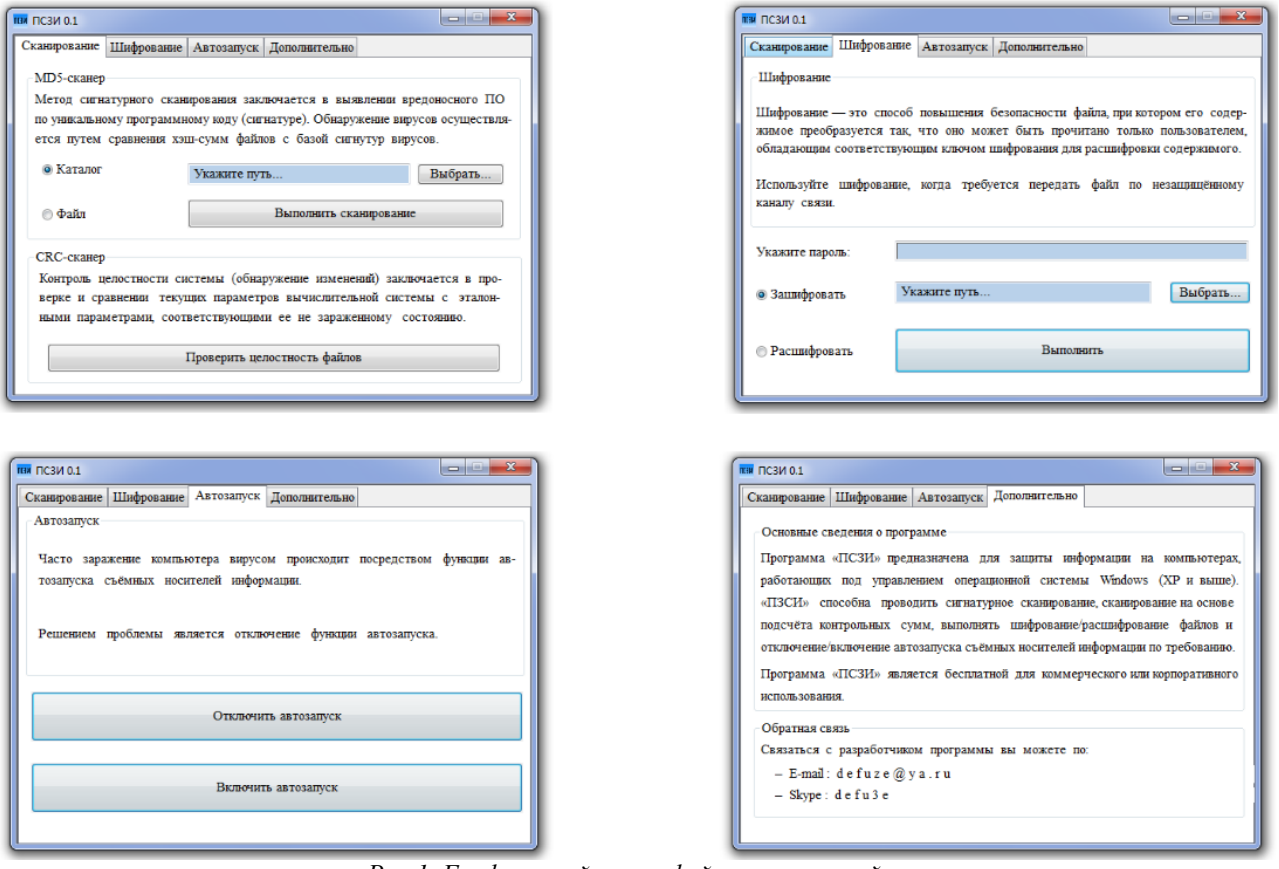


Рис.1. Графический интерфейс программной системы

В результате анализа технологий защиты антивирусных программ, для реализации были выявлены наиболее распространённые и доступные для понимания механизмы средств защиты.

В зависимости от расширения предметной области, увеличения адекватности используемых моделей, необходимости переноса программного продукта на другие платформы, накопления информации о функционировании системы можно выделить следующие пути развития:

- обновление базы сигнатур вирусов;
- добавление модуля защиты защиты от сетевых атак (фаервол);
- добавление модуля сканера портов компьютера;
- добавление модуля защиты компьютера в реальном времени;
- добавление модуля эвристического анализатора;
- создание версий под платформы Android и GNU/Linux;
- добавление возможности сканирования активных процессов;
- добавление функции виртуального запуска приложений (песочница);
- добавление модуля защиты от перехвата функций ввода (кейлоггер);
- расширение языковой поддержки;
- усовершенствование графического интерфейса.

Стоит отметить, что нельзя добиться абсолютной защиты компьютера от компьютерных вирусов отдельными программными средствами. Поэтому для уменьшения потенциальной опасности внедрения компьютерных вирусов и их распространения необходим комплексный подход, сочетающий различные административные меры, программно-технические средства антивирусной защиты, а также средства резервирования и восстановления.

Полученные результаты представляют большой интерес и практическую ценность в перспективе применения их в будущем.

Библиографический список

1. Романец, Ю. В., защита информации в компьютерных системах и сетях / Ю. В. Романец, П. А. Тимофеев, В. Ф. Шаньгин; Под ред. В. Ф. Шаньгина. – 2-е изд., перераб. И доп. – М.: Радио и связь, 2001. – 376с.: ил.
2. Анин, Б. Ю. Защита компьютерной информации. – СПб.: БХВ-Петербург, 2000. – 384 с.: ил.
3. Емельянова, Н. З Защита информации в персональном компьютере учебное пособие / Н. З. Емельянова, Т. Л. Партыка, И. И. Попов. – М. ФОРУМ, 2009. – 368 с. ил. – (Профессиональное образование).
4. Пять важнейших проблем безопасности для предприятий малого и среднего бизнеса [Электронный ресурс] – Режим доступа: http://www.cisco.com/web/RU/downloads/TOP_5_Sec_Issues_for_SMB.pdf
5. Гладких, А. А. Базовые принципы информационной безопасности вычислительных сетей: учебное пособи для студентов, обучающихся по специальностям 08050565, 21040665, 22050165, 23040165 / А.А. Гладких, В.Е. Дементьев;- Ульяновск : УлГТУ, 2009.- 168 с.
6. Обнаружение, основанное на сигнатурах [Электронный ресурс] : Материал из Википедии – свободной энциклопедии : Версия 56935803, сохранённая в 16:08 UTC 11 июля 2013 / Авторы Википедии // Википедия, свободная энциклопедия. – Режим доступа: http://ru.wikipedia.org/?oldid=56935803
7. Панов, А. С. Реверсинг и защита программ от взлома. – СПб. : БХВ-Петербург, 2006. – 256 с.: ил.

УДК 681.8
В.В. Еремеев, студент 4-го курса
Научный руководитель: Н.Е. Губенко, канд. техн. наук, доц.
Донецкий национальный технический университет (г. Донецк)

ОБФУСКАЦИЯ ПРОГРАММНОГО КОДА НА JAVASCRIPT

Аннотация:
В данной статье рассматривается проблема обфускации программного кода на языке JavaScript. Проанализированы основные методы и способы искажения программного кода, выявлены их приемущества и недостатки. На основе изученных методик обфускации предлагается свой алгоритм преобразования скриптов, дающий определенные преимущества по сравнению с рассмотренными существующими аналогами.
Ключевые слова: обфускация, JavaScript, шифрование, алгоритм.
Annotation: The title of the article is obfuscation of JavaScript code. Much attention is given to various distorting algorithmes with their advantages and limitations. Based on the study methods, author proposed obfuscation algorithm, which gives certain advantages in comparison with existing analogues discussed
Key words: obfuscation, JavaScript, algorithm, encryption.

Цель работы – анализ и совершенствование методов обфускации скриптов, написанных на JavaScript.

Для достижения данной цели был рассмотрен ряд методов обфускации кода, а также написан авторский скрипт, преобразующий программный код в нечитабельный байтовый шифр.

Обфускация программ сегодня — это актуальная отрасль криптографии. Анализ электронных ресурсов, посвященных данному вопросу, показал что за последние два года появилось более 70-ти научных статей по этой теме, она вызывает жаркие дискуссии, создает настоящие гонки между исследовательскими группами, открывает полигон для научных изысканий. Как оказалось, обфускация — фундаментальный примитив, дающий полигон для научных изысканий в современной криптографии.

Давая пользователям доступ к установочным файлам программ, компании неизбежно раскрывают свои профессиональные секреты и наработки, и ничто не останавливает конкурентов от беззастенчивого копирования и воровства чужих алгоритмов. Обратим внимание и на другой пример - важные обновления (патчи), исправляющие ошибки в операционных системах и программных продуктах. Почти мгновенно очередное обновление анализируется хакерами, они выявляют проблему, которую это обновление создает.

Эти две ситуации связывает одна общая проблема, а именно: написанная человеком программа может быть человеком же и понята, проанализирована, разобрана. Что если существовал бы способ переделать код до неузнаваемости, при этом сохраняя его функциональность? Такой алгоритм существует, и не один, и называется он «обфусцирующий», а программа, реализующая его, носит название «обфускатор».

Для разработки авторского обфускатора на первом этапе были проанализированы различные методы обфускации кода, использующие особенности языка JavaScript, представленные ниже.

Первый способ достаточно известен. Он реализована такими минимизаторами как JS Packer, JSmin, YUI Compressor, Closure compiler и рядом других (http://packer.50x.eu, http://crockford.com/javascript/jsmin, http://yui.github.io/yuicompressor, https://closure-compiler.appspot.com/home).

Например, они превращают код, представленный на рис. 1 в коды, представленные на рис. 2, 3 и 4:

```
function MyClass(){
    this.foo = function(argument1, argument2){
var addedArgs = parseInt(argument1)+parseInt(argument2);
    return addedArgs;}
var anonymousInnerFunction = function(){// do stuff here!}}
```

Рис. 1. Фрагмент исходного кода на JS

```
function MyClass(){this.foo=function(c,b){var
d=parseInt(c)+parseInt(b);return d};var a=function(){}};
```

Рис. 2. Фрагмент кода на JS после обфускации JS Packer

```
var _0xd799=["\x66\x6F\x6F"];function
MyClass(){this[_0xd799[0]]=function
(_0xefcax2,_0xefcax3){var
_0xefcax4=parseInt(_0xefcax2)+parseInt(_0xefcax3);return _0xefcax4;} ;var
_0xefcax5=function (){} ;} ;
```

Рис. 3. Фрагмент кода на JS после обфускации JSmin

```
eval(function(p,a,c,k,e,d){e=function(c){return
c};if(!''.replace(/^/,String)){while(c-
){d[c]=k[c]||c}k=[function(e){return
d[e]}];e=function(){return'\\w+'};c=1};while(c-
){if(k[c]){p=p.replace(new RegExp('\\b'+e(c)+'\\b','g'),k[c])}}return
p}('4 0="3 5!";9
2(1){6(1+"\\7"+0)}2("8");',10,10,'a|msg|MsgBox|Hello|var|World|alert|n|O
K|function'.split('|'),0,{}))
```

Рис. 4. Фрагмент кода на JS после обфускации YUI Compressor

Этот код легко восстанавливается с помощью деобфускатора jsbeautifier.org либо просто удалением eval, после чего получим код, принадлежность которого к JavaScript очевидна.

Способ второй позволяет изменить код до неузнаваемости, который превратит небольшой скрипт alert(0) в последовательность символов, представленную на рис. 5:

```
([][(![]+[])[!+[]+!+[]+!+[]]+(!![]+[][(![]+[])[+[]]+(![]+[]+[][[]])[+!+[
]+[+[]]]+(![]+[])[!+[]+!+[]]+(!![]+[])[+[]]+(!![]+[])[!+[]+!+[]+!+[]]+(!![]+[]
)[+!+[]]])[+!+[]+[+[]]]+(!![]+[])[+!+[]]+(!![]+[])[+[]]][([][(![]+[])[+[]]+(![
]+[]+[][[]])[+!+[]+[+[]]]+(![]+[])[!+[]+!+[]]+(!![]+[])[+[]]+(!![]+[])[!+[]+!+
[]+!+[]]+(!![]+[])[+!+[]]]+[])[!+[]+!+[]+!+[]]+(![]+[])[+!+[]]+(![]+[])[!+[]+!
+[]]+(![]+[])[!+[]+!+[]]()[(![]+[])[+!+[]]+(![]+[])[!+[]+!+[]]+(!![]+[])[!+[]
+!+[]+!+[]]+(!![]+[])[+!+[]]+(!![]+[])[+[]]])(+[])
```

Рис. 5. Фрагмент кода на JS после обфускации

На первый взгляд трудно определить, к какому языку принадлежит данный код.

Рассмотрим пример, подтверждающий возможность использования особенностей языка для обфускации кода. Пусть имеем после обфускации данным способом последовательность символов, представленную на рис. 6

```
($=[$=[]][(__=!$+$)[_=-~-~-~$]+({}+$)[_/_]+
($$=($_=!''+$)[_/_]+$_[+$])])()[__[_/_]+__
[_+~$]+$_[_]+$$](_/_)
```

Рис. 6. Фрагмент кода на JS после обфускации

На самом деле здесь зашифрован alert(1). Рассмотрим представленную запись как последовательность замен следующего вида. $=[] это пустой массив, $=[$=[]] это массив со ссылкой на массив. В переменной $ будет 0. Теперь у нас есть 0 и мы можем ссылаться на него.

Заменяем __ = "false" через (__ = !$ + $)

Далее _ = -~-~-~$

Оператор ~ в JavaScript означает -(N+1) поэтому -~ = +1

Если $ = 0 тогда -~-~-~$ = 3

Получаем: _ = 3

Таким образом имеем следующий промежуточный код: _/_ = 3/3 = 1

```
(__ = !$ + $ )[ _ = -~-~-~$]
("false")[_]
("false")[3]
"false"[3] = s
({} + $)[_/_]
(" object")[_/_]
(" object")[1]
" object"[1] = o
$$ = ( $_ = !" + $)[_/_]
$$ = ( "true")[1]
"true"[1] = r
$_[+$] = "true"[0] = t
$_ = "true" null
$$ = rt
($$ = ( $_ = !" + $)[_/_] + $_[+$] ))

!" = "true"
$_ = (true)
$_[1] = r
$_[0] = t
$$ = rt
```

Таким образом первая строка станет ($ = [] ["s" + "o"+ "r"+ "t"])()

($=[]["sort"])()

Идем дельше

```
[__[_/_]+__[_+~$]+$_[_]+$$](_/_)
$ = 0
_ = 3
__ = "false"
$_ = "true"
$$ = "rt"
```

Строка превращается…

[__[_/_]+__[_+~$]+$_[_]+$$](_/_)

Превращается…

[__[1] + __[3 + -1] + $_[3] + $$)(1);

Превращается…

["false"[1] + "false"[3 + -1] + "true"[3] + "rt"] (1)

Превращается в

["a" + "l" + "e" + "r" + "t"](1)

В конце мы получаем

($=[]["sort"])()["alert"](1)

Разобьём по частям

a = [] // Создает массив

b = a["sort"] // Получает ссылку на метод sort

c = b() // Выполяет метод вне контекста массива для получения ссылки на window

d = c["alert"] // Получает ссылку на window.alert

d(1) // Выполняет window.alert с аргументом 1

Итог: window["alert"](1)

Приведенный пример демонстрирует черезмерную трудоемкость данного метода.

Способ третий, в отличие от первых двух, может сделать код вообще невидимым. Именно этот прием используется в разработанном обфускаторе. Код, обработанный им, будет состоять из двух частей: видимая часть – декриптор (для его минимизации используются описанные выше способы) и невидимая часть – сам зашифрованный скрипт, который мы хотим скрыть.

Если с видимой все ясно, то секрет невидимой заключается в следующем: обфускатор превращает скрипт в строку состоящую из знаков табуляции (бит 1) и пробелов (бит 0).

В итоге мы получим код в 8 раз длиннее исходного, чем имели. Видимая часть будет декодировать невидимую часть и её выполнять: декодирует биты в число, а число переделает в символ String.fromCharCode(), ну а дальше используется eval.

Для реализации этого подхода был написан скрипт, представляющий собой шифратор введенного пользователем программного кода, а также дешифратор закодированной информации для проверки работоспособности полученного скрипта. И шифратор, и дешифратор имеют возможность выполнять введенный и зашифрованный коды.

Код реализованного алгоритма представлен на рис. 7.

```
      <textarea id='code' cols='60' rows='7'></textarea>
          <br />
          <button onclick='decode()'>Расшифровать
&darr;</button>   
          <button onclick='encode()'>Зашифровать &uarr;</button>
          <button style='float: right; margin-right:
20px'onclick='execInvis()'>Выполнить</button>
          <br /><br />
          Исходник:<br />
          <textarea id='source' cols='60' rows='7'>var test = "Текст";
      alert(test);</textarea><br />
          <button style='float: right; margin-right: 20px'
onclick='exec()'>Выполнить</button>
          <br />
          Замена для 1: <input type='text' value='   ' id='char_true' /><br />
          Замена для 0: <input type='text' value=' ' id='char_false' /><br />
      </div>
      </body>

      <script type='text/javascript'>

      function g(id)
      {return document.getElementById(id)}

      function init()
      {g('code').value = document.getElementById('invis_data').value}
```

```
      function decode(returnCode)
      {
          xcode = document.getElementById('invis_data').value.split("\n")
          result = ''
          char_true = document.getElementById('char_true').value || '\t'

          for(i in xcode)
          {encoded = ''
              for(j in xcode[i]) encoded += (xcode[i][j] == char_true)? "1" :
"0"
              chr = parseInt(encoded, 2) result +=
String.fromCharCode(chr.toString(10))
          }
          res = result.substr(0, result.length - 1)

          if(returnCode !== undefined) return res
          document.getElementById('source').value = res }

      function encode()
      {   newcode = document.getElementById('source').value.split("\n")
          result = ''
          char_true = document.getElementById('char_true').value || '\t'
          char_false = document.getElementById('char_false').value || ' '

          for (k in newcode) // strings
          for (i in newcode[k]) // chars
          {d = parseInt(newcode[k].charCodeAt(i)).toString(2)
           for (j in d) result += (parseInt(d[j]))? char_true : char_false
           result += '\n'
          }
          document.getElementById('code').value = result
          document.getElementById('invis_data').value = result}

      function exec()
      {eval(document.getElementById('source').value)}

      function execInvis()
      {eval(decode('return'))}
      </script>
```

Рис. 7. Разработанный алгоритм обфускации кода

Данный способ является действительно эффективным для обфускации программного кода, так как без дешифратора и знания алгоритма кодирования определение языка скрипта и его содержания весьма затруднительно, а рядовой пользователь при просмотре кода страницы вообще не видит зашифрованный скрипт. Однако его реализация показала, что он не эффективен для очень больших скриптов, работающих на ресурсах с высокой посещаемостью, по причине значительного (в 8 раз) увеличения исходного кода, а, следовательно, и объема передаваемой информации.

Библиографический список

1. Запуск кода из строки. Learn JS [Электронный ресурс]. - Режим доступа: https://learn.javascript.ru/eval
2. Как написать неплохой обфускатор. Записки программиста [Электронный ресурс]. - Режим доступа: http://eax.me/good-obfuscator/
3. Обфускация программ. Хабрахабр [Электронный ресурс]. - Режим доступа: https://habrahabr.ru/post/255871/

УДК 004.056.5
С.В. Медгаус, студент 4-го курса
Научный руководитель: А.В. Чернышова,ст. преподаватель.
Донецкий национальный технический университет

ПРОЕКТИРОВАНИЕ ОБФУСКАТОРА ДЛЯ ЯЗЫКА JAVASCRIPT

Аннотация: В тексте данной статьи описан процесс обфускации и существующие обфускаторы для языка JavaScript.Описаны возможные преобразования исходного кода для запутывания кода и для усложнения возможности реверс-инжиниринга. Исходя из анализа существующих обфускаторов, была предложена разработка обфускатора языка JavaScriptc открытым кодом, реализующая сложные преобразования исходного кода. Также описаны технологии, которые будут использоваться для разбора программного кода.
Ключевые слова: JavaScript, обфускация, реверс-инжиниринг, синтаксическое дерево, запутывающие преобразования.
Annotation: Obfuscation process and existing JavaScript obfuscators are described in this article. There are transformations for tangling and for making difficulties for revers-engineers. Designing opensource JavaScript obfuscator is offered following from existing obfuscators that can use complicated code transformation. Also technologies that will be used to disassemble source code are described.
Keywords: JavaScript, obfuscation, revers-engineering, syntax tree, tangling transformations.

На сегодняшний день проблема сохранения прав на владение интеллектуальной собственностью стоит очень остро. В то время как настольные приложения могут обеспечить себя различными средствами лицензирования и защиты, веб-приложения не могут защитить свой код, расположенный на стороне клиента от просмотра и изучения.

Частично эту проблему решает перенос особенно важных частей кода на сторону сервера, что повлечёт за собой увеличение нагрузки на сервер и может потребовать увеличения мощности сервера или покупки дополнительного оборудования.

Также решению этой проблемы может поспособствовать обфускация программного кода, передающегося клиенту. Обфускация не решит эту проблему полностью, так как возможность использовать реверс-инжиниринг остаётся, но вот потенциальная сложность деобфускации и большие человеческие ресурсы должны отбить желание у злоумышленников разбирать чужой код и извлекать из него алгоритмы.

Обфускация (от лат. obfuscare — затенять, затемнять; и англ. obfuscate — делать неочевидным, запутанным, сбивать с толку) или запутывание кода — приведение исходного текста или исполняемого кода программы к виду, сохраняющему его функциональность, но затрудняющему анализ и понимание алгоритмов работы.

Для создания запутанного ассемблерного текста могут использоваться специализированные компиляторы, использующие неочевидные или недокументированные возможности среды исполнения программы. Существуют также специальные программы, производящие обфускацию, так называемые обфускаторы [1].

На данный момент существуют 3 уровня обфускации. Первый и самый низкий уровень – обфускация машинных команд. Этот уровень обфускации наименее изучен, и поэтому наименее распространён.

Следующий уровень – это уровень «байт-кода», используемый для таких языков как Javaили C#, в которых исходный код программы сначала преобразуется в байт-код, а уже потом преобразуется в машинный код. Этот уровень обфускации достаточно распространён.

И последний, 3 уровень обфускации – это запутывание исходного кода на самом верхнем уровне. Этот уровень обфускации применим для скриптовых языков.

Процессы обфускации можно классифицировать по видам, в зависимости от способа модификации кода программы.

Лексическая обфускация заключается в форматировании кода программы, изменении его структуры, таким образом, чтобы он стал нечитабельным, менее информативным и трудным для изучения.

Обфускация такого вида включает в себя:

- удаление всех комментариев в коде программы, или изменение их на дезинформирующие;

- удаление различных пробелов, отступов которые обычно используют для лучшего визуального восприятия кода программы;
- замену имён идентификаторов на произвольные длинные наборы символов, которые трудно воспринимать человеку;
- добавление различных лишних операций;
- изменение расположения блоков (функций, процедур) программы, таким образом, чтобы это никоим образом не повлияло на её работоспособность.

Обфускация данных связана с трансформацией структур данных. Она считается более сложной, является наиболее продвинутой и часто используемой. Её принято делить на три основные группы:

- обфускация хранения заключается в трансформации хранилищ данных, а также самих типов данных (например, создание и использование необычных типов данных, изменение представления существующих и т.д.);
- обфускация соединения стремится к усложнению используемых структур данных, что достигается благодаря соединению независимых данных, или разделению зависимых;
- обфускация переупорядочения заключается в изменении последовательности объявления переменных, внутреннего расположения хранилищ данных, а также переупорядочивании методов, массивов (использование нетривиального представления многомерных массивов), определённых полей в структурах и т.д.

Обфускация управления осуществляет запутывание потока управления, то есть последовательность выполнения программного кода. Методы, позволяющие осуществить обфускацию управления, классифицируются на три основных группы:

- вычислительная обфускация, которая реализуется расширением условий циклов, добавлением недостижимого исходного кода, устранением библиотечных вызовов (если это возможно) и добавлением мёртвого (недостижимого) кода;
- обфускация соединения, при этом происходитобъединение или разделение определённых фрагментов кода программы для того, чтобы убрать логические связи между ними;
- обфускация последовательности заключается в переупорядочивании блоков (инструкций переходов), циклов, выражений.[2]

На данный момент существует несколько обфускаторов для JavaScript, но необходимо отметить, что все обфускаторы кроме GoogleClosureCompilerявляются в первую очередь упаковщиками или минификаторами (программы, которые минимизируют размер исходного кода, выполняют оптимизацию для увеличения производительности), а уже как следствие, они являются обфускаторами.

Сравнивая популярные упаковщики (YUIpacker и Packer) были проведены эксперименты по минимизации JavaScriptбиблиотеки JQuery и получили такие результаты: YUIpacker сжал неупакованную версию библиотеки до 0.47 от оригинального размера, Packerсжал до 0.34 от оригинального размера. Можно сделать вывод, что лучшим из них является Packer.

Однако данные программы не являются обфускаторами, которые выполняют сложные преобразования над исходным кодом программы, как упоминалось выше.

Также был рассмотрен коммерческий продукт JavaScriptObfuscator v2.18 [3], который предлагает бесплатную демонстрационную версию, обеспечивающий минимальную обфускацию на уровне упаковщиков. При обфускации JQuery исходный файл был обфусцирован до 0.38 от оригинального размера, но там не обфусцированы типы данных, поток логики приложения и т.д.

GoogleClosureCompilerразработан таким образом, чтобы на этапе написания кода необходимо было готовить код к обфускации, соблюдая определённые правила и добавляя определённые аннотации. Так как это не представляется возможным сделать для библиотеки

JQuery, то степень сжатия и обфускации GoogleClosureCompilerневозможно сравнить в данном контексте.

Так как в настоящее время нет обфускатора с открытым исходным кодом, который сможет предоставлять продвинутые способы обфускации, то создание такого обфускатора для языка JavaScript является актуальным.

После анализа существующих обфускаторов для языка JavaScript, была разработана диаграмма прецедентов (см. рис. 1).

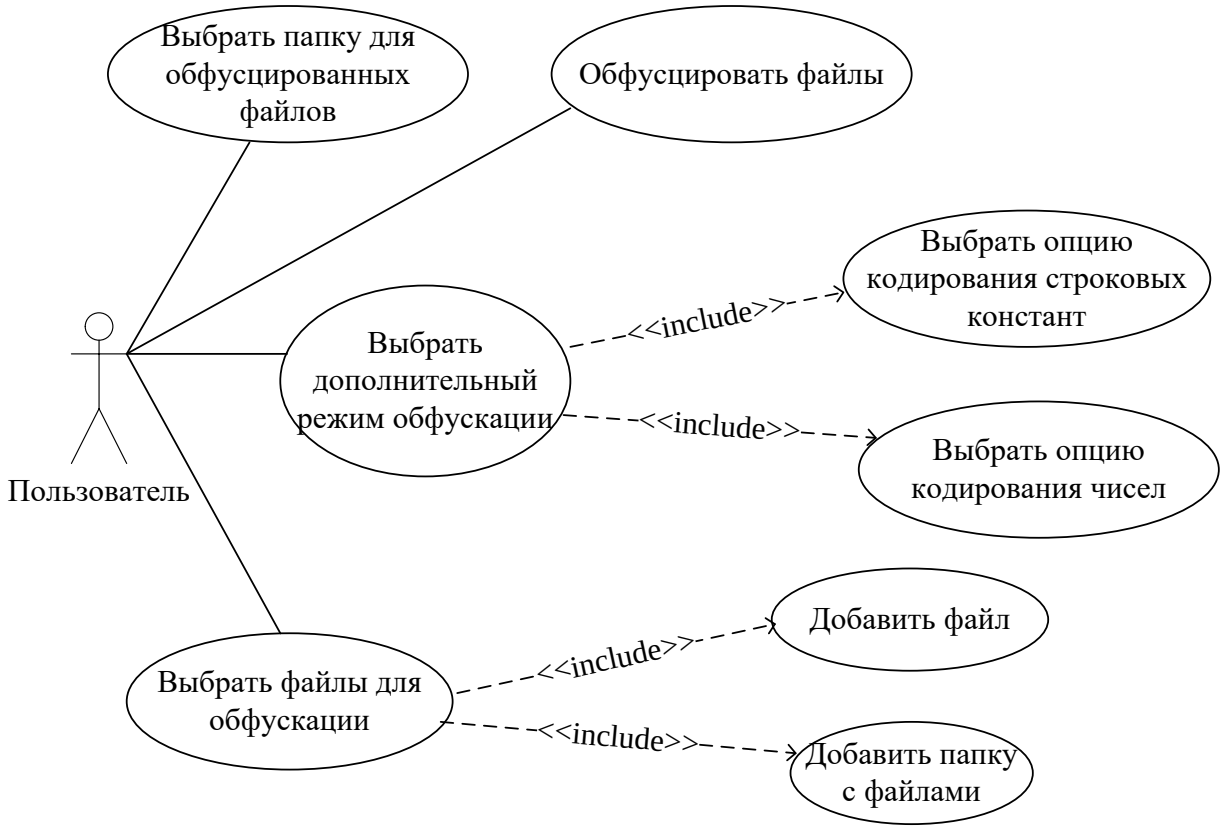


Рис.1. UML диаграмма прецедентов

На данном этапе разработки появилась задача распознавания исходного кода программы и построение абстрактного синтаксического дерева. Так как это задача достаточно объёмная, то для ускорения разработки необходимо использовать готовое решение (библиотеку). Сравнивая существующие синтаксические анализаторы, были обнаружены два лидера: библиотека Esprima[4]от JQueryи встроенный интерпретатор JavaScriptNashorn(потомок Rhino) [5]от Oracle.

Esprima – это библиотека на JavaScriptдля разбора исходного кода того же языка, но так как разработка программного продукта планируется на языке Java, то предпочтение отдано Nashorn, хотя в разработке не будет задействован весь интерпретатор, а только синтаксический анализатор. Nashornбудет принимать на вход исходный код JavaScriptи возвращать в виде результата текст в формате JSON (см. рис. 2), с которым уже можно тесно взаимодействовать и использовать для проведения обфускации.

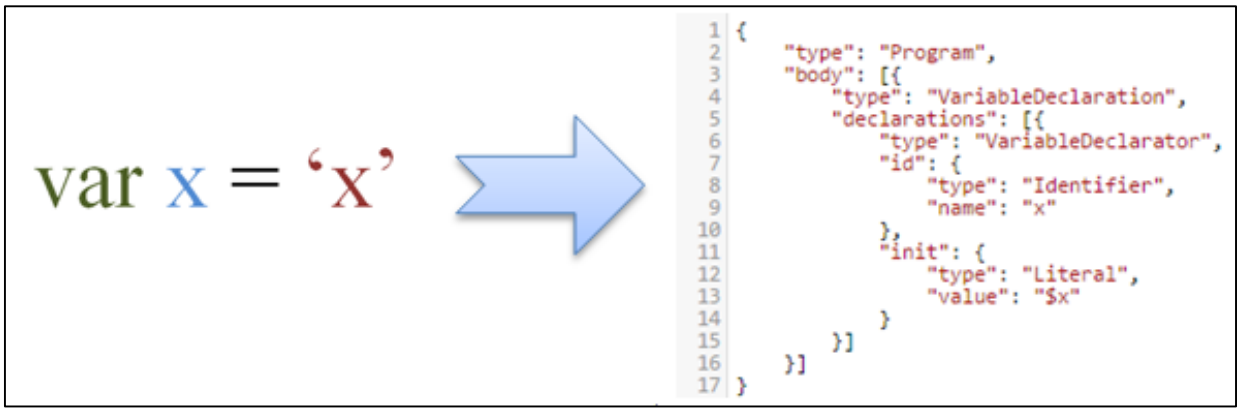


Рис.2. Пример разбора исходного кода JavaScript

Исходя из того факта, что использование готового синтаксического анализатора языка упростило работу, можно сделать акцент на применении различных стратегий обфускации. Таким образом, первый вид обфускации, который будет использован – это лексическая обфускация, наиболее простая в реализации и показывающая наибольший эффект.Вторым видом используемой обфускации будет обфускация управления, которая также показывает хорошие показатели эффективности. И, конечно, будет использовано кодирование строковых и числовых констант.

В данной работе были рассмотрены понятия обфускатора и обфускации. Также были представлены основные принципы обфусцирующего преобразования исходного кода, предложена UML диаграмма прецедентов для разрабатываемого обфускатора.

Исходя из результатов, полученных в ходе анализа существующих обфускаторов, можно сказать, что создание обфускатора с открытым исходным кодом для языка JavaScript, обеспечивающего возможность использования продвинутых запутывающих преобразований, актуально и востребовано.

Библиографический список:

1. Википедия. Обфускация [электронный ресурс]. – Режим доступа: https://ru.wikipedia.org/wiki/Обфускация
2. LynX. Обфускация и защита программных продуктов // CITForum. [электронный ресурс]. – Режим доступа: http://citforum.ru/security/articles/obfus/
3. JavaScriptObfuscator. Canada [электронный ресурс]. – Режим доступа: https://www.javascriptobfuscator.com/
4. Esprima[электронный ресурс]. – Режим доступа: https://www.esprima.org/
5. Nashorn [электронный ресурс]. – Режим доступа: https://ru.wikipedia.org/wiki/Nashorn_(движок_JavaScript)

УДК 681.3
С.А. Чернокуцатова, магистрант, студентка 2-го курса
Научный руководитель: Е.Н. Мащенко, канд. техн. наук, доц.
Севастопольский государственный университет

ИМИТАЦИОННАЯ МОДЕЛЬ ПОДСИСТЕМЫ УПРАВЛЕНИЯ ДОСТУПОМ К РЕСУРСАМ ИНФОРМАЦИОННОЙ СИСТЕМЫ

Аннотация: В работе представлена имитационная модель подсистемы управления доступом к ресурсам информационной системы, реализованная в системе имитационного моделирования Anylogic. Приведены результаты экспериментов с моделью.
Ключевые слова: управление доступом, дискретная модель, Anylogic, идентификация, аутентификация, потоки информации.
Annotation: The paper presents a simulation model of a subsystem access to resources information management system, implemented in the simulation system Anylogic. Results of experiments with the model was presented.
Keywords: access control, discrete model, Anylogic, identification, authentication, information flows.

Система безопасности всегда будет являться одним из наиболее важных факторов в работе всего производства или каждого отдельного пользователя. Один из наиболее ценных ресурсов в мире на данный момент – это информация. Учитывая быстрый рост численности шпионских программ и их разнообразие, а так же профессионализм хакеров, угроза целостности и конфиденциальности данных – это реальность, с которой постоянно нужно бороться, а улучшение уже существующих систем защиты информации – вынужденная необходимость.

Подсистема управления доступом к ресурсам информационной системы является одной из наиболее важных подсистем системы информационной безопасности. Конечно, нельзя приуменьшать вклад остальных составных системы защиты, но данная подсистема является основной целью для нападок хакеров и шпионских программ. Целью работы является создание имитационной модели данной подсистемы в среде AnyLogic.

Разделим данную подсистему на пять блоков, каждый из которых будет отвечать за определенный сегмент системы:

- идентификация;
- аутентификация;
- контроль доступа пользователей;
- управление потоками информации;
- очистка освобождаемых областей оперативной памяти и внешних накопителей.

Блок идентификации отвечает за процедуру создания уникального признака-идентификатора, который позволить контролировать опознавание полномочий субъекта, запрашиваемых при взаимодействии с системой.

В ходе работы данного блока не только создается признак-идентификатор, а так же определяются полномочия пользователя (они контролируются в ходе всего сеанса работы) и регистрация всех действий субъекта.

Следующим блоком является аутентификация. На данном этапе проверяется подлинность представляемого идентификатора и принадлежность субъекту доступа данного пользователя к определенному ресурсу. Механизм аутентификации задает нужные администратору правила доступа и безопасности, а также настройки подсчета трафика.

Объектом аутентификации могут выступать пользователи, различные устройства, приложения, текстовая и другая информация и т.д.

Эти два блока неразрывно связаны, так как способ проверки определяет необходимые параметры, которые будут необходимы для предоставления доступа к системе.

Есть множество различных способов аутентификации. Рассмотрим наиболее распространенные из них.

Самым простым способом является проверка IP –адреса. Проверятся его соответствие адресу, указанному в профиле пользователя. Более сложным является связка IP + MAC. Так как изменение IP-адреса уже довольно легкая и доступная всем процедура, добавляется MAC-адрес, привязанный к конкретной сетевой карте.

Так же существуют различные виды аутентификации по логину и паролю. Достоинства парольной аутентификации – это простота и привычность. Пользователь уже давно привык к данному способу защиты, так что выполнения данного процесса не вызывает у него сложностей. Но при этом данный способ является довольно слабым средством проверки подлинности. Так как зачастую пароли выбираются легкие и запоминающиеся, их записывают в блокноты, рассказывают третьему лицу и т.д. А так же существуют определенные программы подбирающие множество вариантов паролей для взлома. Поэтому существуют определенные ограничения для создания пароля: достаточная длина, цифры, буквы, знаки пунктуации, разный регистр и т.д. Ограничение по количеству ошибок ввода пароля, срок действия паролей, подтверждение паролей по телефону или почтовому ящику и т.д.

Контроль доступа пользователей отвечает за все возможные запросы к ресурсам системы, внешним устройствам, программам, файлам и т.д. Данный блок позволяет контролировать действия, проводимые пользователями или процессами над ресурсами системы. Здесь рассматривается логическое управление доступом, которое позволяет обеспечить конфиденциальность и целостность объектов, а так же устанавливает запрет доступа неавторизированным пользователям.

Очистка освобождаемых областей оперативной памяти и внешних накопителей должна производиться однократной произвольной записью в освобождаемую область памяти, ранее использованную для хранения защищаемыхданных (файлов)[1].

Управление потоками информации исключает возможность несанкционированное изменение и перенаправление информации, а также изменение правил доступа к ресурсам.

Три элемента данной подсистемы реализованы отдельными агентами.
1. Контроль доступа пользователей (процессов) к системе, внешним устройствам, программам, файлам и т.д. (Рис.1.1)

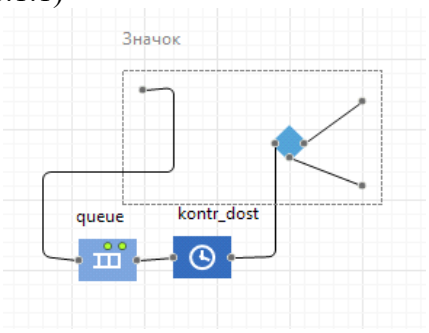


Рис. 1.1 Контроль доступа

2. Управление потоками информации (Рис.1.2.)

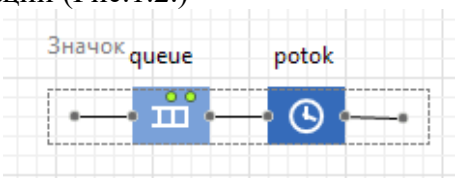


Рис. 1.2. Управление потоками информации

3. Очистка освобождаемых областей оперативной памяти и внешних накопителей (Рис.1.3.)

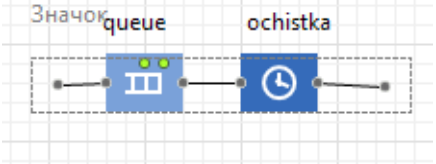


Рис. 1.3. Очистка ОП и внешних накопителей

При помощи среды моделирования Anylogic построили дискретно событийную модель (рис. 1.4.).

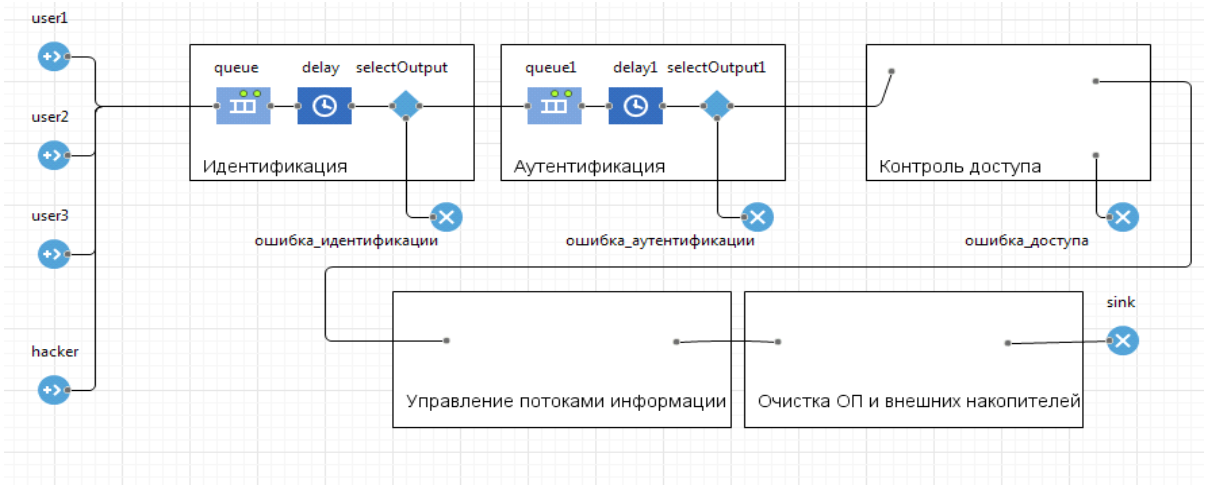


Рис. 1.4. Подсистема управления доступом к ресурсам информационной безопасности

Эксперименты с моделью

На входе системы работают три пользователя и один хакер. Каждый пользователь генерирует 100 заявок, хакер – 50 заявок. В первом эксперименте, на каждом этапе мы видим, что было 177 ошибок идентификации, 103 ошибки аутентификации, 39 ошибок доступа и 31 удачные заявки. Во втором эксперименте - 161 ошибка идентификации, 107 ошибок аутентификации, 17 ошибок доступа и 65 удачных заявок.

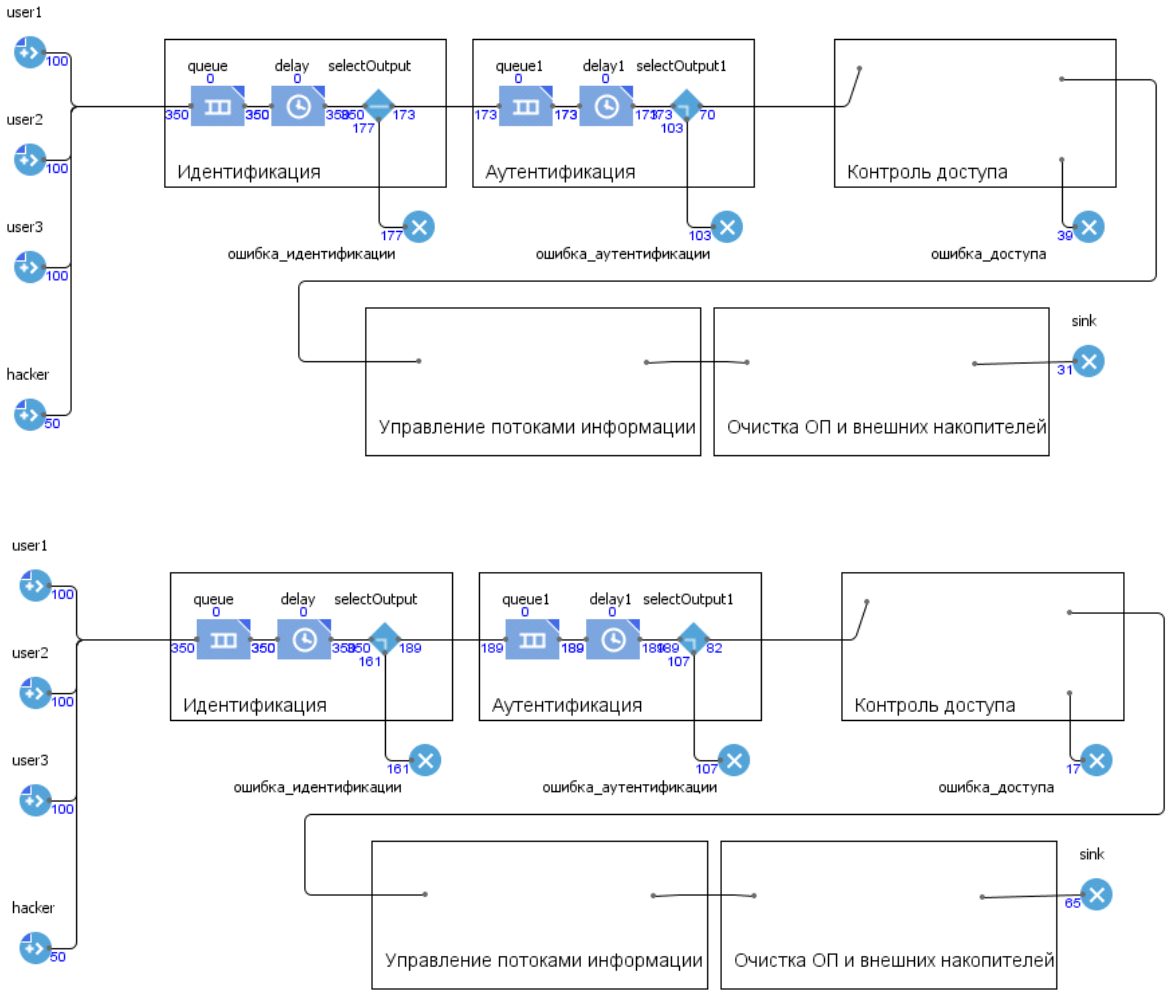


Рисунок 1.5. Эксперименты с моделью управления доступом к ресурсам информационной системы

Перспективы исследований. В дальнейшем модель будет усложняться и дополняться, в модели будут использоваться параметры реальных систем безопасности. Применение модели позволит проводить анализ работы системы и искать определенные блоки, из-за которых система не предоставляет полноценной защиты. После определения данных блоков будут искаться пути разрешения проблем: изменение методов, улучшение данных блоков и т.д. По результатам исследований предполагается выработка рекомендаций для систем поддержки принятия решений.

Библиографический список

1. Руководящий документ ГостехкомиссииРосии «Автоматизированные системы. Защита от несанкционированного доступа к информации» – 30 марта 1992г. [Электронный ресурс] – Режим доступа: http://fstec.ru/component/attachments /download/296 (дата обращения: 10.02.2016).
2. Anylogic.ru [Электронный ресурс]: Инструмент многоподходного имитационного моделирования / AnyLogic - единственный многоподходный инструмент имитационного моделирования: агентное, дискретно-событийное моделирование и системная динамика. – 2012. – 21 марта. – Режим доступа: http://www.anylogic.ru/ (дата обращения: 14.02.2016).

СЕКЦИЯ 9

Системное и прикладное программное обеспечение

УДК 65.011.56
Т. В. Ванжа, студентка 5-го курса
Научный руководитель: Ю. К. Орлов, канд. техн. наук, доц.
Донецкий национальный технический университет

РАЗРАБОТКА МОДЕЛЕЙ И МЕТОДОВ УПРАВЛЕНИЯ ГОРНЫМ ПРЕДПРИЯТИЕМ В УСЛОВИЯХ ЭКОНОМИЧЕСКИХ РИСКОВ

Аннотация: Предложено определение экономического риска предприятия, методы управления рисками, модели системы управления рисками: «Черный ящик» и функциональная модель, предложены этапы цикла принятия управленческих решений в сфере организации управления экономическими рисками. Для анализа использованы методы структурного анализа. На основе полученных в результате экспертизы данных построена таблица внутрипроизводственных факторов экономического риска шахты. Разрабатывается перечень антирисковых мероприятий и возможных альтернативных стратегий шахты.
Ключевые слова: экономический риск, угледобывающее предприятие.
Annotation: In the article the following definitions are given: the economical enterprise risk, the risk management, the risk management system models: the "Black box" and a functional model,(я бы разделил тут текст на 2 предложения - и в русском варианте тоже) there were proposed the stages of making managerial decisions within the scope of organizing the economical risk management. The analysis is based on the structural analysis methods. Based on the data, acquired as the result of an examination, there was build a table that consists of the internal economical risk factors during mining. The list of counter-risk events and possible alternative mining strategies is in development.
Keywords: economical risk, coal-mining enterprise.

В условиях рыночных отношений проблема оценки и учета экономического риска приобретает самостоятельное теоретическое и прикладное значение как важная составная часть теории и практики управления.

Большинство управленческих решений принимается в условиях риска, что обусловлено рядом факторов – отсутствием полной информации, наличием противоборствующих тенденций, элементами случайности и многим другим.

Экономический риск – это возможность случайного возникновения нежелательных убытков, измеряемых в денежном выражении [1].

Эффективное управление риском требует не только внимательного наблюдения за размером риска, но также стратегию минимизации убытков. Для борьбы с рисками применяются известные методы управления [2] (Рисунок 1).

Рис. 1. Методы управления риском

Модель «черный ящик». Для построения этой модели необходимо определить границы системы управления, внешнюю среду, входы и выходы системы [3].

Граница системы – само горное предприятие.

В качестве внешней среды для системы управлениями риском выступает среда функционирования предприятия.

Входами в систему есть внешние факторы, которые несут за собой ситуации риска (политические, социально-экономические, экологические, научно-технические).

К выходам системы относятся программы антирисковых мер, т.е. регулирование риска и обеспечение компенсации возможных потерь и убытков.

Модель «Черный ящик» представлена на рисунке 2.

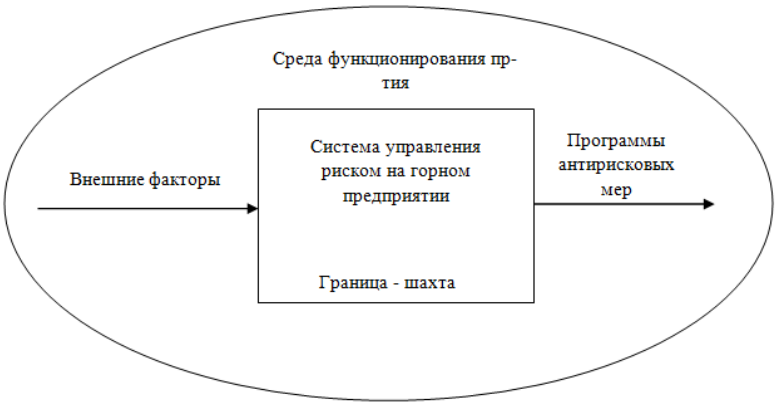


Рис. 2. Черный ящик

Функциональная модель системы. Функциональная модель системы управления рисками представлена на рисунке 3.

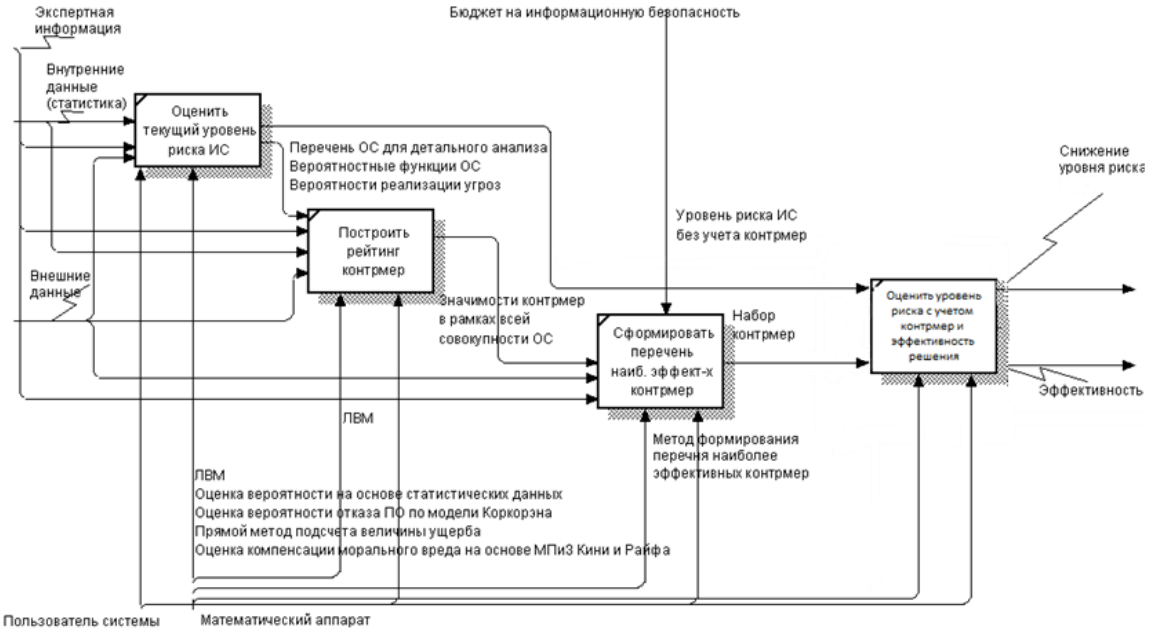


Рис. 3. Функциональная модель системы

Важным этапом организации риск-менеджмента являются контроль за выполнением намеченной программы, анализ и оценка результатов выполнения выбранного варианта рискового решения. При этом рекомендуется аккумулировать всю информацию об ошибках и недостатках разработки программы, проявившихся в ходе ее реализации. Такой подход позволит провести разработку последующих программ мероприятий по снижению рисков на более качественном уровне с использованием новых полученных знаний о риске [4].

Результаты каждого этапа становятся исходными данными для последующих этапов, образуя систему принятия решений с обратной связью. Такая система обеспечивает

максимально эффективное достижение целей, поскольку знание, получаемое на каждом из этапов, позволяет корректировать не только методы воздействия на риск, но и сами цели управления рисками [5]. Этапы цикла принятия управленческих решений в сфере организации управления экономическими рисками представлены на рисунке 4.

Рис. 4. Этапы цикла принятия управленческих решений в сфере организации управления экономическими рисками

Как экономическое понятие риск возникает вместе с формированием товарно-денежных отношений и развитием рыночной экономики. В Большом экономическом словаре риск определяется как «возможность того или иного результата от принимаемого хозяйственного решения» [6]. Риск можно рассматривать как сочетание негативных и позитивных факторов.

Определив экономический риск как вероятность получения хозяйствующим субъектом экономических потерь сверх предусмотренных прогнозными расчетами, можно выделить два похода к теории рисков: линейный и нелинейный, их графическое описание представлено на рисунке 5.

В рамках данного исследования экономический риск понимается как вероятность реализации угроз, способных оказать существенное негативное влияние на исследуемую экономическую систему и изменить ее текущее состояние. Негативное влияние экономического риска носит самый различный характер: материальные, денежные потери, потери времени, скорости оборота капитала и др.

Функция распределения вероятности поведения экономического показателя, характеризующего нелинейную экономическую систему, позволяет произвести оценку риска.

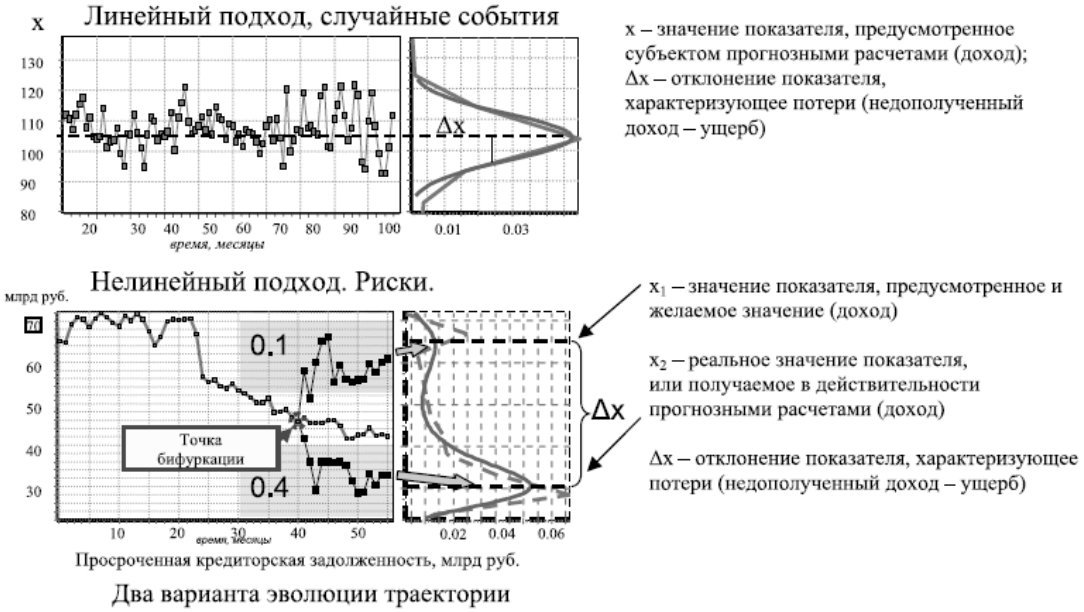


Рис. 5. Линейный и нелинейный подходы к теории риска

При наличии малых флуктуаций нелинейная система описывается вероятностной функцией распределения g (плотностью вероятности), вычисляемой по исходному временному ряду и связанной с потенциальной функцией системы F посредством уравнения Фоккера-Планка [7]:

$$\frac{dg}{dt} = \nabla(g\nabla F) + \nabla^2(Dg), \quad (1)$$

где g – плотность вероятности;

F – потенциальная функция рассматриваемой нелинейной системы, характеризующая количество устойчивых и неустойчивых точек равновесия, соответствующих экстремума функции.

На основе полученных в результате экспертизы данных построена таблица внутрипроизводственных факторов экономического риска шахты, представлена ниже [8].

Таблица 1– Внутрипроизводственные факторы экономического риска шахты

Производственно-технические факторы	**Хозяйственные факторы**	**Факторы риска нанесения экологического ущерба**
1	2	3
Нарушение правил техники безопасности	Сбои программного обеспечения (потеря информации)	Отсутствие или несвоевременное проведение мерприятий по отслеживанию уровня шахтных вод, газообильности, провалов и деформаций земной поверхности
Степень изношенности горно-шахтного оборудования	Нехватка железнодорожных вагонов для отгрузки угля	Неэффективные технологии утилизации и экологическая безопасность размещения породы
Горные удары и суфлярные выделения	Невыполнение обязательств потребителями угля	Изношенность очистных сооружений
Неподготовленность фронта очистных работ	Недобросовестные сделки и умышленное мошенничество	Низкоэффективные технологии и устройства очистки от тонкодисперсной пыли и газообразных веществ

Продолжение таблицы 1

1	2	3
Непредвиденные горно-геологические нарушения	Отсутствие договоров на поставку сырья и материалов и оказание услуг	Несоответствие уровня экологозащиты уровню добычи
Нехватка и высокий коэффициент текучести промышленно-производственного персонала	Высокий коэффициент текучести и низкая квалификация непромышленной группы персонала	Несвоевременное проведение рекультивации нарушенных под производственные нужды земель и ландшафта
Газообильность и водообильность месторождения	Ошибки в экономических расчетах основных показателей работы шахты	Неэффективные технологии физико-химической и биологической очистки шахтных вод

На основе анализа динамики коэффициентов рискоустойчивости дается оценка реального положения угледобывающего предприятия, выявляются резервы улучшения сложившегося положения, разрабатывается перечень антирисковых мероприятий и возможных альтернативных стратегий шахты.

Разработка стратегии и стратегических управленческих решений играет ведущую роль среди множества разнообразных управленческих решений, принимаемых на шахтах, поскольку они определяют пути развития шахты. Стратегические и управленческие решения определяются характером решаемых задач, внутренними и внешними условиями производства. Отсутствие научно обоснованных подходов к формированию экономической стратегии и стратегических управленческих решений приводит к недостаточно эффективной направленности годовых и перспективных планов предприятий, что не позволяет стабилизировать работу шахт и создать условия для их развития.

Библиографический список

1. Гранатуров В.М. Экономический риск: сущность, методы измерения, пути снижения: Учебное пособие. – М.: Изд-во «Дело и Сервис», 1999. – 112 с.
2. Методы анализа рисков. [Электронный ресурс]. Режим доступа - http://www.risk24.ru/metodianaliza.htm.(Дата доступа – 1.03.16).
3.Т.В. Ванжа, Ю. К. Орлов. Разработка моделей и методов управления горным предприятием в условиях экономических рисков. Компьютерная и программная инженерия. Сборник материалов международной научно-технической конференции студентов, аспирантов и молодых ученых 15-16 декабря 2015 года. – Донецк, ДонНТУ – 2015. – 314-317 с.
4. Вяткин В.Н., Гамза В.А., Екатеринославский Ю.Ю., Хэмптон Дж.,Дж. Управление риском в рыночной экономике. – М.: Экономика, 2002. – 195 с.
5. Балабанов И.Т. Риск-менеджмент. – М.: Финансы и статистика, 1996. – 135 с.
6. Мамаева, Л.Н. Управление рисками: Учебное пособие / Л.Н. Мамаева. –М.: Дашков и К, 2013. –256 с.
7. Соложенцев *Е. Д.* Сценарное логико-вероятностное управление риском в бизнесе и технике: изд. 2-е. — СПб.: Издательский дом «Бизнес-пресса», 2006. — 530 с.
8. Леонова О.А. Принципы и подходы к оценке рисков горнодобывающего предприятия // Известия вузов Северо-Кавказского региона, Технические науки – Ростов-на-Дону, 2003. – Приложение 2. – С. 107 – 110.

УДК 681.3
М.С. Ветренко, магистрант
Научный руководитель: Ю.К.Апраксин, профессор
Севастопольский государственный университет

БАЛАНСИРОВКА НАГРУЗКИ В КЛАСТЕРЕ JENKINS

Аннотация: В статье рассматривается подход балансировки нагрузки в кластере Jenkins в условиях неопределенного закона распределения потока входных заявок. Данный подход малочувствителен к способу определения весов и дуг графа заявки.
Ключевые слова: Балансировка нагрузки, распределенные системы, Jenkins.
Annotation: The article deals with an approach to load balancing cluster Jenkins in conditions of uncertainty of the law of distribution of the input stream applications. This approach is not sensitive to the method of determining the weights of the arcs and the graph of the application.
Key words:Load balancing, distributed systems, Jenkins.

Введение

Процесс так называемой непрерывной интеграции (Continuous Integration) программного кода подразумевает автоматическую или автоматизированную сборку программного проекта с целью компиляции проекта, обнаружения интеграционных проблем и статистического анализа. Одним из примеров системы непрерывной интеграции является система Jenkins. Для большей производительности сервера Jenkins объединяют в кластер, что позволяет значительно ускорить скорость обработки заявок к системе. Однако для оптимизации распределения заявок между узлами кластера необходима балансировка нагрузки.

В статье рассматривается задача балансировки нагрузки в кластере Jenkins. Обработка заявок подразумевает выполнение типовых задач. Задачи могу выполняться независимо или в заданном порядке. В кластере Jenkins распределение интервалов поступления заявок от пользователей суммарный поток трудно описать каким-либо известным законом распределения, потому в таком случае целесообразно выделять ресурсы для обработки независимо от каждой заявки с учетом загруженности вычислительной системы. Далее будет рассматриваться обработка пакетных заданий.

Планирование обработки заявки

Каждая заявка в системе Jenkins является списком работ на построение или тестирование проекта и имеет иерархическую структуру. Есть одна основная работа на построение которая порождает нисходящие построения. Так же есть одна задача формирования результата выполнения всего списка работ. Для планирования обработки такого пакета данных, каждую заявку в кластере Jenkins удобно представлять в виде направленного графа. Каждая вершина это отдельная работа, а дуги указывают на последовательность работ. Каждый граф имеет одну вершину-начало – работу по формированию исходных данных для пакетной обработки. Так же есть одна вершина-конец для заключительных действий.

Проблема решения задач в виде направленных ацикличных графов решена различным подходами, например приоритетной обработкой критических путей и графа по уровням [1], или группировкой узлов в кластеры [2].

В кластере Jenkins продолжительность выполнения работы зависит от конкретного вычислительного узла, то и способы оценки веса вершины графа различны, например, максимальное значение, минимальное значение, среднее значение и так далее. Исследование [3] показывает, что различные способы оценки вершин и дуг графа ведут к различным способам планирования обработки данных.

Huimin Zhao [4] предлагает ранжировать вершины направленного графа для того, чтобы разбить их на группы работ, выполнение которых независимо друг от друга. Этот способ один из наименее чувствителен к способу назначения весов дугам графа.

На вход планировщика поступает невзвешенный граф, матрица M (1) и четырехмерный массив G (2).

$$M = \begin{pmatrix} m_{11} & .. & m_{1m} \\ .. & m_{ij} & .. \\ m_{n1} & .. & m_{nm} \end{pmatrix}, i = 1..n, j = 1..m \quad (1)$$

$$G = (g_{ijkr}), i = 1..n, i \neq j, k = 1..m, r = 1..m \quad (2)$$

Матрица G задает стоимость выполнения каждой из n работ на каждом из m узлов вычисления. Массив G задает стоимость передачи данных между пары работ для всех комбинаций узлов вычисления, на которых эти работы могут выполняться. Планирование обработки осуществляется в три этапа:

1. ранжирование работы;
2. группировка работ;
3. планирование в группах работ.

На этапе ранжирования работ каждой вершине (стоимость выполнения) и дуге (стоимость передачи) назначается вес, который рассчитывается как средняя величина по всем узлам (для вершин) или парам доступных узлов (для дуг) по формуле 3.

$$r_i = \omega_i + max(c_{ij} + r_j), \forall j \epsilon S_i \quad (3)$$

В выражении (3) ω_i – вес вершины i, S_i – множество дочерних узлов узла i, c_{ij} – вес дуги между узлами i, j.

Второй этап заключается в сортировке вершин в порядке убываниях их рангов и разбиения на группы. Вершина, у которой максимальный ранг, добавляется в группу 0. Далее в эту группу добавляются вершины в порядке убывания рангов до тех пор, пока не обнаруживается связь дугой в графе. Тогда вершина с наименьшим рангом образует новую группу с номером предыдущей группы плюс 1. Цикл повторяется до тех пор, пока все вершины не попадут в какие-либо группы.

На третьем этапе происходит планирование выполнения работ внутри каждой группы, начиная с наименьшего номера группы. Для этого составляется матрица ESG (4) где элемент ESG_{ij}- ближайшее возможное время выполнения работы i на узле j и определяется выражением (5).

$$ESG = \begin{pmatrix} ESG_{11} & .. & ESG_{1m} \\ .. & ESG_{ij} & .. \\ ESG_{n1} & .. & ESG_{nm} \end{pmatrix} \quad (4)$$

$$ESG_{ij} = max(FT_j, EGG_{ij}) \quad (5)$$

В выражении (5) FT_j – время, к которому узел j закончит выполнение всех работ из предыдущих групп, EGG_{ij} – время, к которому все ресурсы для работы i на узле j станут доступны.

В [6] описан один из способов планирования выполнения работ в группе Balanced Minimum Completion Time (сбалансированное минимальное время завершения). BMCT планирование происходит в два этапа. Во время первого этапа происходит первоначальное распределение заявок по узлам. Каждая работа передается в узел, который обеспечит минимальное время выполнения. Распределение выполняется с учетом ресурсов каждого узла. В результате все работы группы выстраиваются в порядке, общее время выполнения которого минимально. Во время второго этапа происходит оптимизация распределения работ из первого этапа. Если перенос какой-либо работы позволяет сократить общее время обработки заявки, то производится переназначение соответствующей работы.

Применение планирования обработки заявки для балансировки нагрузки

Вышеописанный способ может быть применен для балансировки нагрузки кластер Jenkins в условиях неопределенного характера входного потока задач на пакетную обработку.

Все заявки поступают в очередь балансировщика кластера Jenkins. Каждая заявка подвергается планированию, используя способ, описанный выше. Поскольку распределение потока заявок неизвестно, то выделять ресурсы заранее невозможно потому при балансировке нагрузки учитывается текущая загруженность вычислительных ресурсов. Так же для балансировки производится прогнозирование освобождение ресурсов с применением значений ESG для всех работ. На рисунке 1 приведена упрощенная диаграмма загрузки одного сервера Jenkins. Из этого примера видно, что в момент времени 2 доступны ресурсы для выделения полному объему ресурсов за исключением ресурсов, необходимых для работы 1. В момент времени 12 доступны все ресурсы кроме ресурсов для работы 5.

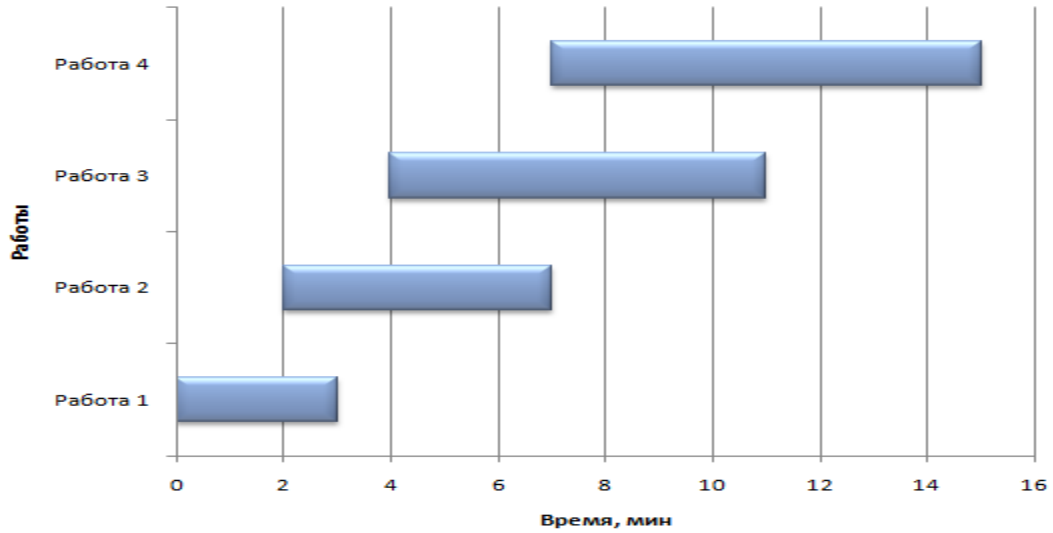


Рис. 1. Временная диаграмма загрузки вычислителя

Заключение

В статье рассмотрен гибридный способ планирования обработки пакетных вычислений в кластере Jenkins в условиях неизвестного закона распределения потока заявок в очередь балансировщика. Способ планирования BCMT позволяет свести к минимуму чувствительность механизма планирования к способу назначения весов узлам и дугам графа заявки.

Библиографический список

3. Iverson M., Ozguner F., Follen G. Parallelizing existing applications in a distributed heterogeneous environment /IPPS 95, Proceedings of the Heterogeneous Computing Workshop. 1995. – p.93-100.
4. Cirou B., Jeannot E. Triplet: A clustering scheduling algorithm for heterogeneous systems/International Conference on Parallel Processing Workshop, 2001. – p.231-236.
5. Zhao H., Sakellariou R. An experimental investigation into the rank function of the heterogeneous earliest finish time scheduling algorithm/Euro-Par. 2003. – p.189- 194.
6. Zhao H., Sakellariou R. A Hybrid Heuristic for DAG Scheduling on Heterogeneous Systems/Proceedings of 13th Heterogeneous Computing Workshop (HCW 2004).2004.

УДК 004.457
А. А. Геращенко, магистрант
Научный руководитель: С.Н.Фисун, канд. техн. наук, доц.
Севастопольский государственный университет

АНАЛИЗ ОСНОВНЫХ ПРОГРАММНЫХ ПРОДУКТОВ ПО СИСТЕМЕ ЭЛЕКТРОННОЙ КОММЕРЦИИ

Аннотация: В работе описывается суть и основы функционирования технологий электронной коммерции. Указываются функции составных элементов и осуществлена классификация Интернет-магазинов. Проводится подробный анализ современных программных средств по системе Э-коммерции, с целью их эффективного выбора в процессе применения.
Ключевые слова: Технологии и системы электронной коммерции, Интернет-магазин, программные средства и продукты, готовое программное решение.
Abstract: The paper describes the nature and foundations of the technologies of electronic commerce. Specify components and functions implemented classification online stores. Conduced a detailed analysis of modern software tools for e-commerce system, with a view to their effective choice in the application.
Keywords: Technology and e-commerce, online store, software and products, ready-made software solution.

Основным направлением экономического развития 21-го века, является электронная коммерция. Она представляет собой сделки, исполняемые и заключаемые с применением информационных и коммуникационных технологий, т.е. под электронной коммерцией подразумевают любую форму бизнес-процесса, в котором взаимодействие между субъектами совершается электронным образом – с применением Интернет-технологий. В широком смысле Э-коммерция – это купля-продажа товаров, либо услуг предприятиями, домашними хозяйствами, физическими лицами, правительствами и другими государственными или частными организациями, осуществляемая по компьютерным сетям. Данное понятие включает в себя заказы, получаемые или размещаемые с использованием любого онлайнового приложения, применяемого при осуществлении автоматизированных сделок, таких, как Интернет-приложения, технология EDI или интерактивные телефонные системы.

У систем электронной коммерции имеется модульная структура, позволяющая производить процедуры продаж товаров определенного вида (книги, программное обеспечение, компьютерная и бытовая техника, одежда и т. д.) с помощью электронных платежных средств (кредитные карты, смарт-карты, цифровые деньги и электронные чеки).

Как и электронная коммерция, другие электронные бизнес-процессы уже оказывают глубокое и преобразующее воздействие на экономику и общество. Они оказывают влияние на структуру фирм, а также весь цикл предложения и потребления, начиная с разработки продукта и заканчивая после – продажным обслуживанием.

Электронный бизнес включает в себя как внутренние (внутрифирменные), так и внешние бизнес-процессы (включая бизнес-процессы, реализовываемые экономическими субъектами при реализации коммерческой деятельности), проходящие при использовании любых форм электронных сетей существующих в мире (глобальные, локальные, внутрифирменные и др.).

Нынешние информационные технологии в системе электронной коммерции, включают в себя специальную инфраструктуру аппаратного и программного обеспечения, специальные приложения, общие службы, а также правовую структуру и надлежащие стандарты и правила [3].

Основу функционирования системы электронной коммерций составляют электронные магазины. Они играют роль представительства в сети Интернет на основе создания web-сервера. Основная цель создания такого предприятия заключается в обеспечении продажи товаров и оказании услуг другим пользователям сети Интернет.

Электронная коммерция может осуществляться на интернациональном и национальном уровнях. Основу различия проведения деловых операций на этих уровнях, составляют не технико-технологические составляющие (так как электронная коммерция отличается глобальным характером), а законодательные. На интернациональном уровне реализация системы электронной коммерции значительно усложняется. Это вызвано такими

факторами, как использование разных систем налогообложения, таможенных сборов, принятие индивидуальных и вместе с тем неодинаковых соглашений между разными странами, существенные различия в используемых правилах реализации банковских операций. Функционирование систем электронной коммерции на национальном уровне связано, в основном, с представительством фирмы в сети, рекламой, а также предпродажной и послепродажной поддержкой [4].

Современная модель электронной коммерции состоит из 4-х элементов – объектов, субъектов, бизнес-процессов и компьютерных сетей. Среди субъектов электронной коммерции имеются частные и государственные организации, а также физические лиц.

Объектами модели электронной коммерции являются товары и услуги, в том числе информационного характера, на исполнение которых направлены определенные действия со стороны вышеупомянутых субъектов. Перечень товаров и услуг, которые могут являться объектом электронной коммерции, очень обширен: интеллектуальные товары и услуги – компьютерные программы, консультации, статьи и другая информация, которые могут быть доставлены непосредственно с компьютера продавца на компьютер покупателя; товары, которые могут быть отправлены (доставлены) почтой или курьером – фильмы, книги, а также многое другое, включая бытовую и компьютерную технику, в том числе крупногабаритную, транспортировка которой осуществляется службой доставки продавца.

Бизнес-процессами считаются отдельные конечные процессы, связанные с воздействием субъекта на объект, а также взаимодействием субъектов между собой, в совокупности образующие коммерческую деятельность: торговые операции (процессы купли-продажи и обмена), страхование, платежи и т.п.

Компьютерные сети – это то, посредством чего связываются участники и обеспечиваются процессы.

Бесспорно, существует ряд факторов, непосредственно не участвующих в модели электронной коммерции, все же оказывающих влияние на ее функционирование и развитие. Экономические, политические и географические факторы, состояние конкуренции, государство – способны повлиять на каждый отдельный элемент модели электронной коммерции, так и на модель в целом. Классифицировать Интернет-магазины (ИМ) можно по различным критериям [1]. Среди методов розничной продажи товаров в Сети можно выделить: Интернет-магазины; web-витрины; торговые ряды; торговые системы; контентные проекты (потребительские энциклопедии, системы Интернет-заказов товаров и т.д.).

Виды Интернет-магазинов. В зависимости от функциональных возможностей Интернет-магазин, работающий в режиме онлайн, может быть отнесен к одной из трех категорий: Интернет-витрина; торговый автомат; автоматический магазин.

Аппаратное обеспечение Интернет-магазина можно условно разделить на две части: клиентское и серверное. Клиентские компьютеры – это, как правило, недорогие персоналки, оснащенные средствами мультимедиа. Собственно компьютеры ИМ – это мощные и надежные компьютеры, выполняющие продажу товаров и услуг.

Анализ некоторых современных систем электронной коммерции. На рынке наибольшее распространение получили «коробочные» (комплексные, готовые к непосредственному применению) решения, купив которые, можно сравнительно быстро и дешево организовать инфраструктуру (бэк-офис и фронт-офис) виртуального магазина.

К наиболее известным в России, иностранным программным продуктам для Э-коммерции этого класса можно отнести Intershop 3 (Intershop, www.intershop.com; распространяется фирмой TopS, www.tops-msk.com) и Microsoft Merchant Server, который входит в состав Microsoft Site Server/Commerce Edition [2].

Рассмотрим следующие системы электронной торговли, а именно: «Паутина 2.0» (http://www.webtovar.ru); «Shop-Script Premium» (ООО «Артикус»); «Commerce. Optimizer» (ООО «Гарант-Парк-Интернет»); «InternetMag Бизнес» (ООО «Арабикас»); «Битрикс: Управление сайтом – Бизнес» (компания «Битрикс»); «OSG» (компания «Вимком-Оптик ТС»).

Для сравнительного анализа систем, приводится характеристика основных возможностей этих систем.

«Паутина 2.0». Главными особенностями Интернет-магазина «Паутина 2.0» являются:

Простота использования. От пользователя не требуется быть web-программистом или web-дизайнером – достаточно владеть компьютером на уровне среднего пользователя и отчетливо представлять, чего именно пользователь хочет добиться.

Мобильность. Пользователи не привязаны к определенному хостинг-провайдеру, могут выбирать любого. Готовый Интернет-магазин «Паутина 2.0» использует стандартное серверное программное обеспечение, позволяющее размещать его у всех крупных хостинг-провайдеров. Вместе с тем не исключается возможность размещения готового Интернет-магазина на личном выделенном сервере.

Возможность работы с Интернет-магазином в off-line. Пользователю не требуется беспрерывное высокоскоростное Интернет-подключение. На компьютере находится полнофункциональная копия Интернет-магазина.

Совместимость с существующими решениями. Готовый Интернет-магазин «Паутина 2.0» позволяет обмениваться данными с другими приложениями, применяя текстовые файлы.

Качественная поддержка. Имеется полный доступ ко всем обновлениям Интернет-магазина на ftp-сервере, который содержит исправление ошибок и введение новой функциональности.

«Shop-Script Free, Pro, Premium» (www.articus.ru). Продуктами Shop-Script поддерживается большое число возможностей, которые позволят пользователю разработать преуспевающий Интернет-магазин или каталог.

Ниже представлены главные функции решений Shop-Script:

- Удобное управление заказами. SMS-уведомления о заказах. Обратная связь.
- Защита от спамеров и роботов. Поддержка конфигурируемых продуктов.
- Личный кабинет покупателя. Простая интеграция готового дизайна.
- Перекрестный маркетинг или рекомендуемые товары.
- 100% открытый исходный код. Модуль обмена ссылками.
- Фотогалерея для каждого продукта, формирование счетов на оплату и квитанций.
- Оптимизация для поисковых систем и др.

Готовые Интернет-магазины на платформе скриптов Shop-Script. На базе скриптов Shop-Script работают несколько сотен Интернет-магазинов.

Commerce.Optimizer (www.parking.ru) – это готовое решение для быстрого развертывания Интернет-магазина. Он не требует программных доработок под конкретный проект, при этом обеспечивает достаточную гибкость, предоставляет широкие возможности по настройке.

Возможности Commerce.Optimizer в качестве готового решения, предоставляет следующие возможности:

- создание и управление структурой каталога товаров;
- настройка свойств товара. Можно добавлять/изменять свойства;
- настройка свойств покупателя; настройка свойств заказа;
- настройка страниц интерфейса покупателя и менеджера;
- настройка внешнего вида страниц;
- настройка состава информации на страницах.

«InternetMag Бизнес» (http www.internetmag.ru). Internetmag – это российский продукт, учитывающий предпочитаемый в России и странах СНГ уровень расходов на содержание ресурса, потребности большинства отечественных потребителей и основы построения сетевых магазинов, обеспечивающие максимальную отдачу в условиях нашей страны.

Список компонентов системы, которые обязаны находиться в базовом комплекте для обеспечения минимальной работоспособности магазина:

- личный кабинет покупателя; корзина товаров;
- система регистрации покупателей;
- система оформления заказов; платежная система;
- автоматический прайс-лист.

«Битрикс: Управление сайтом – Бизнес» (www.bitrxsoft.ru). Модуль «Интернет-магазин» позволяет организовать продажу товаров, контента и услуг через Интернет, создать Интернет-магазин и управлять электронными каналами продаж товаров и услуг.

Интернет-магазин состоит из следующих функциональных элементов:

- процедура оформления заказа; корзина товаров и услуг;
- персональный раздел покупателя;
- административный раздел.

«OSG WebShop 2/0» (www.osg.ru). OSG WebShop позволяет владельцу:

- создать электронный каталог предлагаемых им на рынок товаров или услуг, который размещается на удаленном компьютере;
- получить эффективную рекламную поддержку своему бизнесу в Интернет;
- организовать круглосуточный канал сбыта без дополнительных финансовых затрат на торговые площади и персонал;
- самостоятельно управлять работой Интернет-магазина (ИМ), используя дружественный интерфейс;
- создавать минимальный трафик в сети Интернет при обновлении базы данных ИМ с удаленного компьютера;
- провести анализ работы ИМ на основании статистики, автоматически формируемой в процессе работы магазина;
- обеспечить обратную связь для создания клиентской базы данных;
- задавать режим автоматического определения категории покупателя (опт, розница и т. п.) и выдавать до четырех соответствующих цен на товар;
- организовать работу по принципу бизнес-бизнес для обслуживания удаленно;
- провести интеграцию ИМ с офисными системами, такими, как склад и бухгалтерия, для автоматизации процесса переноса информации в базы данных ИМ.

OSG Интернет-магазин Start. Программный комплекс «OSG Интернет-магазин Start» предназначен для создания Интернет-магазинов и позволяет с наименьшими затратами времени и средств разработать многофункциональный Интернет-магазин и организовать продажу товаров и услуг в сети Интернет. Он ориентирован как на розничную (B2C), так и на оптовую (B2B) торговлю, и может не только оптимизировать работу с существующими клиентами, но и решить задачу привлечения и удержания новых.

Готовое решение для электронной коммерции «Паутина 2.0». Интернет-магазин «Паутина 2.0» является готовым решением, позволяющим быстро и с минимальными затратами создать собственный Интернет-магазин. Главными особенностями Интернет-магазин «Паутина 2.0» являются: простота использования; мобильность; возможность работы с Интернет-магазином в off-line; качественная поддержка.

Основная функциональность Интернет-магазина сосредоточена в модулях. Системные модули – это программные продукты сторонних производителей, которые могут быть установлены как из дистрибутива Интернет-магазина «Паутина 2.0», так и отдельно (Apache, MySQL). Для того чтобы разместить Интернет-магазин на удаленном сервере с поддержкой виртуального хостинга, необходимо, чтобы хостинг-провайдер предоставлял доступ к этим программным продуктам. Прикладные модули – это части Интернет-магазина.

Основа Интернет-магазина включает в себя средства доступа пользователей к модулям Интернет-магазина через браузер и средства управления модулями для администратора Интернет-магазина:

- Web-интерфейс Интернет-магазина – для запуска Интернет-магазина в браузере.

- Администратор Интернет-магазина – для настройки расположения модулей на сайте Интернет-магазина, дизайна и информационного наполнения модулей.
- Эффективность Менеджер модулей – используется для установки, удаления и обновления модулей.

Однако, для полноценного анализа качественной оценки, для выбора системы, этого недостаточно, поэтому он должен иметь и количественный характер. Для этого выбирается определенный критерий. Приводится перечень главных характеристик потребительского качества программных продуктов для электронной торговли, а именно [2]:

1. Функциональная полнота. Масштабируемость.
2. Пользовательский интерфейс фронт-офиса.
3. Качество поддержки и сопровождения.
4. Качество помощи и документации.
5. Пользовательский интерфейс бэк-офиса.
6. Переносимость. Безопасность.

Этот перечень ранжирован по мере снижения важности.

Известно множество показателей, используемых для создания характеристики качества программных систем, сравнительной оценки их потребительской ценности.

Среди этих показателей важными являются следующие:

- Функциональная полнота. Как обозначается в печати и подтверждает практика, рыночные программные продукты обычно, существенно различаются по критерию функциональной полноты.
- Завершенность разработки. Полностью завершенные программные продукты встречаются довольно редко.
- Быстродействие (затраты времени на решение задачи пользователя).
- Уровень требований к комплексу технических средств (занимаемый объем памяти, быстродействие процессора, свободное пространство на диске и др.).
- Степень и простота настройки на техническую среду (монитор, модем, принтер, и др.).
- Стоимость. Комплексность решения задачи.
- Возможность перенастройки на новые условия применения (например, в связи с изменением законодательства, появлением новых подразделений фирмы и т. д.).
- Возможность работы в Сети.
- Качество помощи пользователю в процессе работы (например, наличие ситуативной, контекстно-зависимой и гипертекстовой помощи с оглавлением).
- Требования к уровню квалификации пользователя.
- Трудоемкость освоения и внедрения.
- Качество пользовательского интерфейса.

Таким образом, были подробно анализированы основные особенности существующих программных средств в области применения системы Э-коммерции. Результаты анализа могут быть применены при использовании системы Э-коммерции или при выборе имеющихся программных средств согласно установленной цели перед организацией или компанией.

Библиографический список

1. Е.В.Сибирская, О.А.Старцева. Э-коммерция. М., Форум, 2010, 288 стр.
2. О.Г.Нельзина. Информационные системы для электронной коммерции. Ростов н+в, фемикс, 2008.
3. Определение и содержание термина "Электронная коммерция" [Электронный ресурс] // Интернет публикации [Офиц. сайт]. URL: http://mirznanii.com/info/opredelenie-i-soderzhanie-termina-elektronnaya-kommertsiya_138475 (дата обращения: 17.02.2016).
4. Введение в электронную коммерцию [Электронный ресурс] // Интернет публикации [Офиц. сайт]. URL: http://www.studfiles.ru/preview/4326866/ (дата обращения: 19.02.2016).

УДК 681.3.053
А.В. Гримута, студент 3-го курса
И.В. Смирнов, студент 3-го курса
Научный руководитель: С.Н. Фисун, канд. техн. наук, доцент
Севастопольский государственный университет

РАЗРАБОТКА, ОТЛАДКА И ТЕСТИРОВАНИЕ УЧЕБНОГО ПРОГРАММНОГО КОМПИЛЯТОРА ЯЗЫКА ВЫСОКОГО УРОВНЯ.

Аннотация: В статье анализируются значение разработки компиляторов для компьютерных инженеров. Уделяется внимание основным этапам работы компилятора. Рассматриваются алгоритмы обхода и способы обработки этапов.
Ключевые слова: компилятор, байт-код, виртуальная машина java.
Annotation: The article analyzes the importance of compilers development for the computer engineers. Particular attention was paid to the compilers` key stages. Traversal algorithm and stages processing method were considere.
Key words: compiler, byte-code, java virtual machine.

На сегодняшний день существует более 8000 языков программирования (ЯП), для выполнения которых необходима трансляция исходного кода высокого уровня в эквивалентную программу низкого уровня, т.е. необходима реализация компиляции программы. Изучение основы компилятора того или иного ЯП, предоставляет возможность, компьютерному инженеру, пользоваться его особенностями, а основываясь на принципы обработки компилятора – улучшать исходный код с точки зрения производительности или объема. В связи с этим целью данного исследования является комплексная разработка учебного компилятора, на входе которого будет код, который использует разработанную авторами грамматику, а результатом будет являться байт-код, выполняемый на «виртуальной машине Java» (JVM) (рисунок 1).

Рис. 1. Выполнение программного обеспечения

Стоить отметить, что данное исследование может быть использовано в рамках курсового проекта дисциплины «Системное программное обеспечение». Разработкой компилятора выполняется группой студентов: каждый студент реализует свой модуль. Этапы работы компилятора представлены на рисунке 2. В рамках этого исследования рекомендуется использовать методологию Scrum [1].

Данная методология предполагает, что самый высокий показатель эффективности разработки достигается в том случае, если команда самостоятельно принимает решения в отношении того, как она будет двигаться к цели, что является не стандартным подходом во время выполнения курсового проекта студентами. Однако роль ScrumMaster’а может принадлежать преподавателю, который будет следить за тем, чтобы соблюдались правила Scrum.

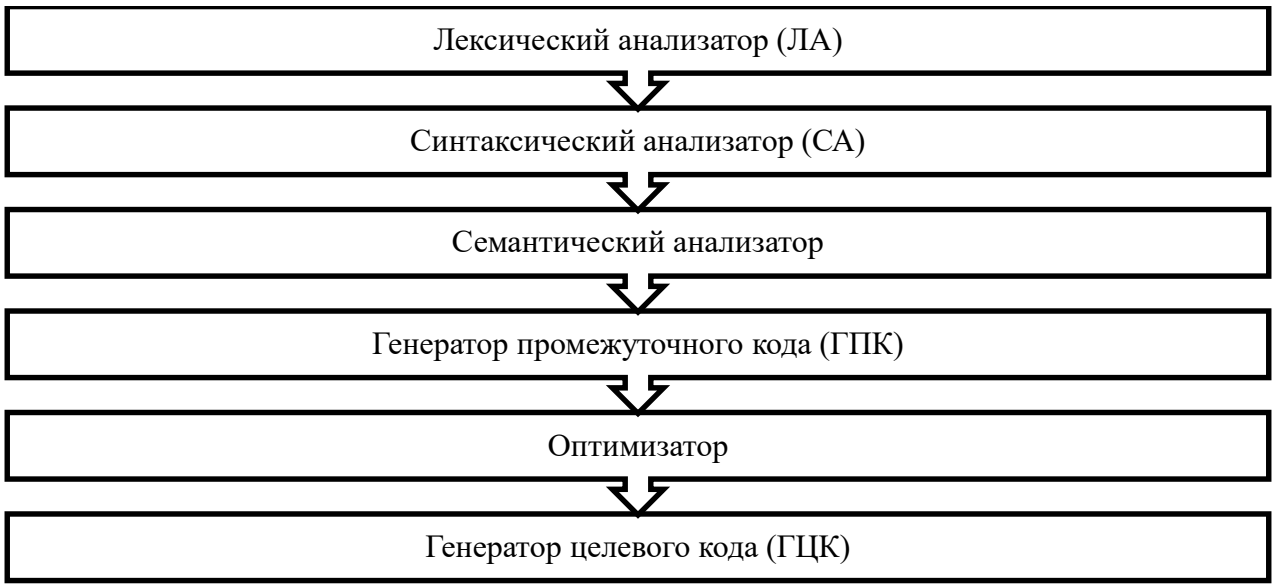


Рис. 2. Этапы работы компилятора

Как говорилось ранее, компилятор работает поэтапно, т.е. в процессе каждого этапа происходит преобразование исходной программы с одного представления к другому.

Роль первого этапа, лексического анализатора (ЛА), состоит в том, чтобы сделать программу понятной остальным этапам. На его вход приходит текст исходный программы, в котором выделяются лексемы входного языка и выводятся перечень всех найденных лексем либо сообщение об ошибке (рисунок 3). При этом основной проблемой разработкой ЛА, является определение границ лексем, т.к. в исходном тексте они никак не ограничены. Но смотря на возможность исключения данного этапа, из-за схожих функций синтаксического анализа, рекомендуется его включение [1].

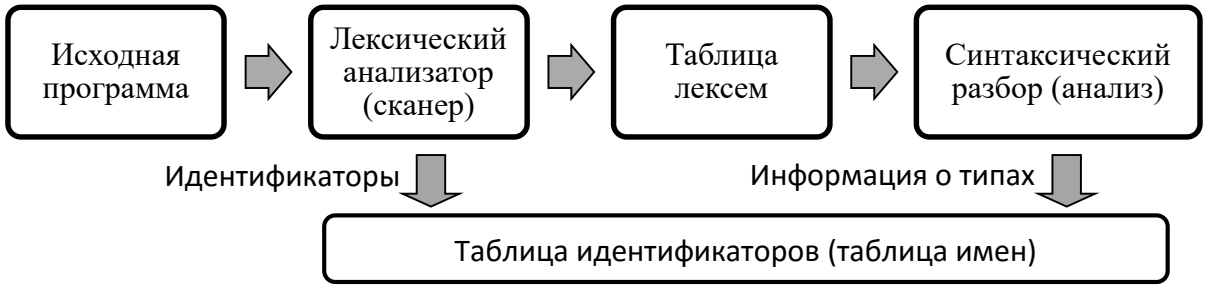


Рис. 3. Лексический анализатор

В данном исследовании авторами было решено объединить синтаксический анализатор (СА) и семантический анализатор. Потому второй модуль выполняет: поиск и выделение синтаксических конструкций, установки типов и проверки их правильности и, наконец, располагает их в виде структуры. Существуют различные алгоритмы разбора входных цепочек: LL(1), LR(1), рекурсивный спуск, предшествования и т.п.. В данной работе был использован метод LL(1), реализация и понимание, которого более понятна для студентов. Для обработки ошибок использовалась стратегия «в режиме паники», что позволило более точно проверить на соответствие грамматике. При этом не стоит упускать тот факт, что семантический анализ выполняется на двух этапах: семантического разбора и в начале генерации промежуточного кода (ГПК).

ГПК является следующим этапом трансляции программы, который использует промежуточное представление программы, предназначенное для эффективной генерации кода и проведения различных оптимизаций программы. Классическими методами ГПК являются: обратная польская нотация, синтаксическое дерево и трехадресный код [2]. Простейшей

формой промежуточного представления – является синтаксическое дерево, где вершины кодируется записью с полем для операции и полями для указателей на потомков. Обратной польской нотацией же называется список вершин дерева, в котором каждая вершина следует непосредственно за своими потомками. На сегодняшний день наиболее часто используемые являются трехадресный код, в котором явные имена соответствуют внутренним узлам графа. Из всех возможных реализаций: четверок, троек и косвенных троек (КС), авторами был выбран косвенные тройки, т.к. по сравнению с четверками при использовании КС можно сэкономить память если одно и то же временное значение используется более одного раза.

Предпоследним этапом разработки данного компилятора является оптимизация промежуточного кода. Его роль заключается в удаление избыточных команд и упрощение кода с сохранением его смысла. Новейшие компиляторы используют более осложненные алгоритмы, нежели классические. Потому был использован простой и всеми известный метод свертки [3]. Данный метод был реализован следующим алгоритмом:

- *Если операнд есть переменная, которая содержится в таблице T, то операнд заменяется на соответствующее значение константы.*
- *Если операнд есть ссылка на особую триаду типа C(K,0), то операнд заменяется на значение константы K.*
- *Если все операнды триады являются константами, то триада может быть свернута. Тогда данная триада выполняется и вместо нее помещается особая триада вида C(K,0), где K - константа, результат выполнения свернутой триады. (При генерации кода для особой триады объектный код не порождается, а потому она в дальнейшем может быть просто исключена).*
- *Если триада является присваиванием типа A:=B, тогда:*
    - *если B - константа, то A со значением константы заносится в таблицу T (если там уже было старое значение для A, то это старое значение исключается).*
    - *если B - не константа, то A вообще исключается из таблицы T, если оно там есть.*

Заключающим этапом является генерация целевого кода, который получает на вход промежуточное представление исходной программы и выводит эквивалентную целевую программу. Разработанный учебный компилятор формирует байт-код JVM для последующего выполнения на «виртуальной машине Java», которая представляет собой стековую машину (не имеет прямого доступа к памяти). В процессе генерации целевого кода (ГЦК) были использованы заранее определенные команды для JVM [4]:

- помещение констант на стек,
- помещение локальных переменных на стек,
- запоминание значений из стека в локальных переменных,
- обработка массивов,
- управление стеком,
- арифметические команды,
- логические команды,
- передача управления,
- возврат из функции,
- табличный переход,
- обработка полей объектов,
- вызов метода,
- обработка исключительных ситуаций,
- прочие операции над объектами,
- отладка.

После чего разработанный модуль вырабатывает байт-код JVM. Полная схема работы компилятора представлена на рисунке 4.

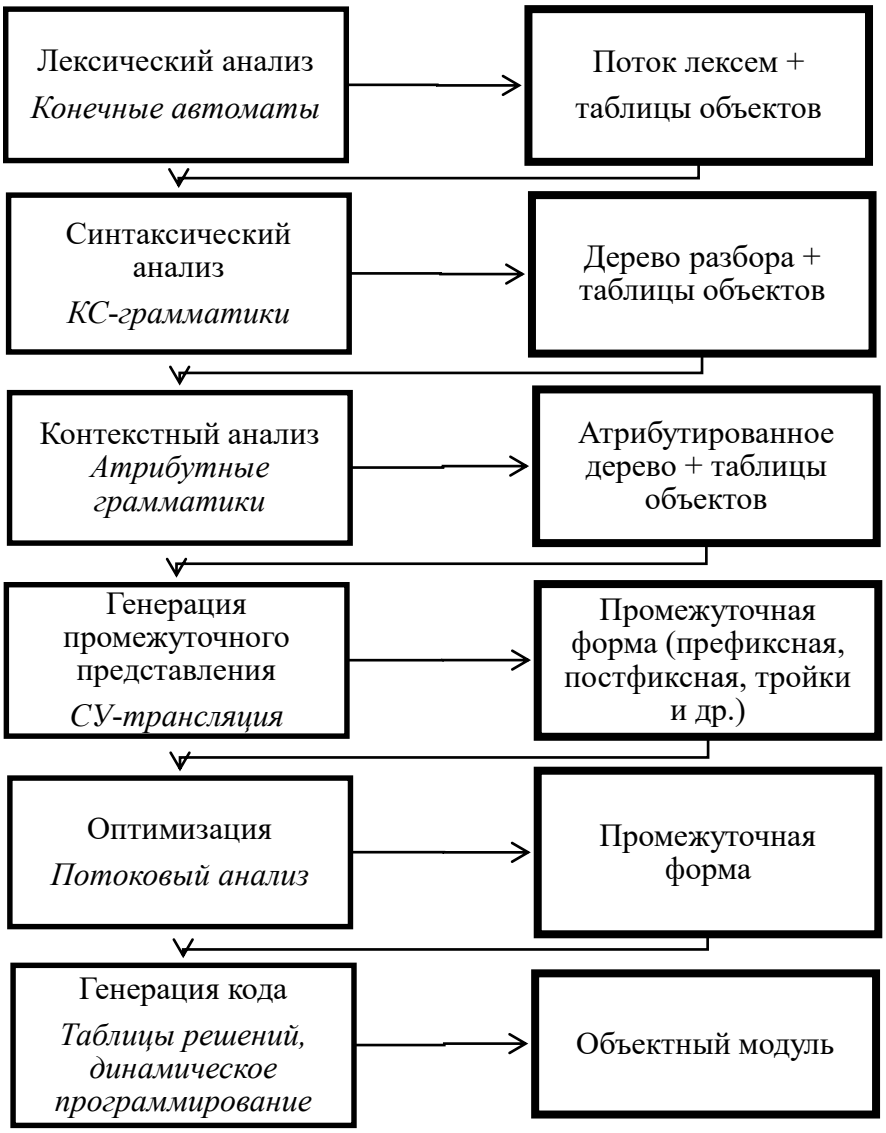


Рис. 4. Полная схема работы компилятора

Таким образом, в ходе данного исследования, авторами был разработан учебный программный компилятор, который переводит разработанную грамматику в байт-код Java. Данное исследование направлено на развитие понимания студентов: основных этапов функционирования компилятора, о работе в команде, при использовании современной и популярной методологии Scrum, а так же на укрепление знаний полученных в рамках дисциплин: «Системное программное обеспечение» и «Инженерное программное обеспечение».

Библиографический список

1. Кеннет Р.С. Основы Scrum: практическое руководство по гибкой разработке ПО / Р.С. Кеннет - Москва: "Вильямс", 2016. - 544 с.
2. Лебедев В.Н. Введение в системы программирования / В.Н. Лебедев - Москва: "Статистика", 1975. - 312 с.
3. Ахо А. Компиляторы. Принципы, технологии, инструменты / А. Ахо - Москва: "Вильямс", 2012. - 768 с.
4. Серебряков В.А. Основы конструирования компиляторов / В.А. Серебряков - Москва: "УРСС", 2001. - 182 с.

УДК 681.3.053
С.Н. Дзюба, студент 3-го курса
Научный руководитель: С.Н. Фисун, канд. техн. наук, доцент
Севастопольский государственный университет

РАЗРАБОТКА СИНТАКСИЧЕСКОГО АНАЛИЗАТОРА С ИСПОЛЬЗОВАНИЕМ СРЕДСТВ АВТОМАТИЗАЦИИ

Аннотация: В статье рассматриваются принципы разработки синтаксического анализатора.
Ключевые слова: компилятор, синтаксический анализ, ANTLR, Теренс Парр.
Annotation: В статье рассматриваются принципы разработки синтаксического анализатора.
Key words:compiler, syntax analyze, ANTLR, Terence Parr.

Базой для разработки компилятора является грамматика рассматриваемого языка программирования. Для этой цели наиболее удобны контекстно-свободные грамматики. Они предоставляют точную и сравнительно легкую в плане понимания синтаксическую спецификацию языка программирования, для них можно автоматически сконструировать эффективный синтаксический анализатор, а также, разработанные на базе контекстно-свободных грамматик компиляторы могут быть достаточно легко расширены. Но следует отметить, что контекстно-свободные грамматики определяют контекстно-свободную составляющую языка программирования, а именно то, каким образом записывается та или иная конструкция языка.

Синтаксический анализ является одним из начальных процессов компиляции программы. Он определяет, принадлежит ли некотораяпоследовательность лексемпорождаемому грамматикой языку. Для любой контекстно-свободной грамматики может быть построен синтаксический анализатор, сложность которого не будет превышать $O(n^3)$для входной строки длинной n, но возможно сконструировать и более быстрый анализатор, разработав иную грамматику.

Входными данными синтаксического анализатора является последовательность лексем входного языка, поступающих с выхода лексического анализатора. Лексический и синтаксический анализатор работают в режиме взаимного исключения: синтаксический анализатор производит обращение к лексическому каждый раз, когда возникает потребность в очередном терминальном символе.

Выходными данными синтаксического анализатора является дерево разбора и таблицы идентификаторов и типов, которые будут использоваться следующим блоком.

Большая часть известных на сегодняшний день методов анализа входят в один из двух классов: нисходящие или восходящие алгоритмы. Классы алгоритмов названы в соответствии с методикой построения синтаксического дерева: от корня к листьям или от листьев к корню, соответственно.

Нисходящие алгоритмы строят вывод от аксиомы грамматики к цепочке терминальных символов. LL- грамматика может быть проанализирована без возврата, входная цепочка просматривается слева-направо, строится левый вывод цепочки.

Нисходящие алгоритмы популярны благодаря тому, что эффективный нисходящий анализатор можно построить вручную, а также LL-грамматики легко обобщаются: не LL-грамматики могут быть проанализированы при помощи метода рекурсивного спуска с возвратами.

При разработке компиляторов для языков программирования грамматика исходного языка часто изменяется, вносятся коррективы. Любое изменение грамматики влечет за собой серьезные последствия, выражающиеся в необходимости перестроения управляющей таблицы. Соответственно, чем больше правил содержит язык, тем сложнее и длительнее процесс перестроения управляющей таблицы. Для упрощения разработки синтаксического анализатора в условиях частых корректировок грамматики были разработаны средства автоматизации разработки блока синтаксического анализа. В настоящее время существует множество таких средств, самые распространенные: ANTLR[2], YACC/BISON[3], LEX/FLEX и др.

ANTLR (отангл. ANotherToolforLanguageRecognition- «еще одно средство распознавания языков»)- генератор нисходящих анализаторов для формальных языков, разработанный профессоромТеренсомПарром из Университета Сан-Франциско. ANTLRпреобразует контекстно-свободную грамматику, представленную в виде расширенной Бэкус- Науровой формы, в программу на таких языках программирования, как C++, Java, C#, Python, Ruby.Широко применяется для разработки компиляторов, интерпретаторов и трансляторов. Преимуществами ANTLRявляется свободное распространение, использование единой нотации, предоставление сообщений об ошибках и восстановление после них, наличие визуальных средств разработки, например, ANTLRWorksдля IDENetBeans.

Рассмотрим принцип работы с ANTLRна примере разработки программы-примитивного калькулятора, входными данными которой будут являются арифметические выражения, разделенные переводом строки, а выходными данными- результат вычислений. Поддерживаемые операции: сложение, вычитание, умножение и деление над целыми числами.

Грамматика записывается по следующим правилам: правило начинается с имени;имена для правил лексического анализатора обязаны начинаться с заглавной буквы;после имени идет символ ":", за которым следуют «альтернативы», которые разделяются сиволом "|" и должны заканчиваться символом ";". Также, существуют и другие элементы языка:

- (...) – подправило;
- (...)* - повторение подправила 0 или более раз;
- (...)+ - Повторение подправила 1 или более раз;
- (...)? - подправило, может отсутствовать;
- {...} - семантические действия (на языке, использующемся в качестве выходного - напр., Java);
- [...] - параметры правила;
- | - оператор альтернативы;
- .. - оператор диапазона;
- ~ - отрицание;
- . - любой символ;
- = - присваивание;
- : - метка начала правила;
- ; - метка конца правила.

Рассмотрим пример для грамматики учебного языка ASPLE-3.

Грамматика выглядит следующим образом:

1. <S>→WVD<Seq>E
2. <Seq>→<Op>|<Op>;<Seq>
3. <Op>→O|<Seq>

В данной грамматике: <S>-оператор цикла, <Seq>-последовательность операторов, <Op>-оператор. Так как второе правило является леворекурсивным, то грамматика не может являться LL(1). Устранив левую рекурсию, грамматика примет следующий вид:

1. <S>→WVD<Seq>E
2. <Seq>→<Op><X>
3. <X>→ε
4. <X>→;<Seq>
5. <Op>→O
6. <Op>→<S>

Начиная с ANTLRверсии 4 исходную грамматику можно записывать с леворекурсивными правилами, но должно соблюдаться отсутствие скрытой рекурсии. Соответственно, для создания анализатора грамматика будет выглядеть следующим образом:

```
grammarExample;
options
{
   language = Java;
}
s
   :'w''v''d' seq 'e'
   ;
seq
   :op
   |op';'seq
   ;
op
   :'o'
   |s
   ;
```

Генерация анализатора происходит при помощи средств утилиты ANTRLWorksEditor. После применения команды генерации анализатора к файлу грамматики, в котором находится исходная грамматика, генерируются классы синтаксического анализатора (рис. 1).

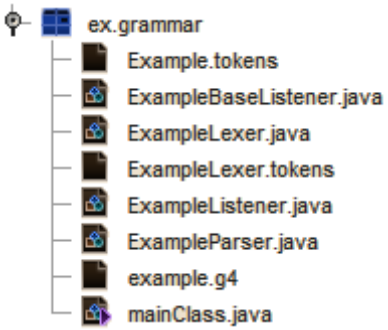


Рис. 1. Структура синтаксического анализатора

Результатом разработки собственного синтаксического анализатора для приведенной грамматики с графическим интерфейсом является один класс с 314 строками кода, что значительно меньше разработанной программы при помощи ANTLR, но трудоемкость разработки собственного синтаксического анализатора значительно выше, в особенности для больших грамматик.

Средства автоматизации построения синтаксических анализаторов значительно упрощают работу разработчиков при разработке и отладке модуля синтаксического анализа компилятора, транслятора или интерпретатора.

Библиографический список

1. Лебедев В.Н. Введение в системы программирования / В.Н. Лебедев - Москва: "Статистика", 1975. - 312 с.
2. Definite ANTLR Reference, Terence Parr, 2007.ISBN- 10: 0-9787392-5-6, ISBN-13: 978-09787392-4-9..
3. Серебряков В.А., Галочкин М.П. Основы конструирования компиляторов. М., 1993. - 193с.

УДК 681.3
Д.А. Лось, студент 4-го курса
Научный руководитель: В.А. Строганов, старший преподаватель
Севастопольский государственный университет

ПОДСИСТЕМА ПРИЕМА И ПЕРЕДАЧИ СООБЩЕНИЙ СИСТЕМЫ АНАЛИЗА И УПРАВЛЕНИЯ ГОРОДСКИМ МАРШРУТНЫМ ТРАНСПОРТОМ

Аннотация: Статья посвящена вопросам реализации подсистемы обмена сообщениями (Message-OrientedMiddleware, MOM) системы анализа и управления городским маршрутным транспортом. Автор раскрывает основные отличия между механизмами синхронного взаимодействия подсистем (RPC) и асинхронного (MOM). Формируются требования и описывается реализация подсистемы обмена сообщениями.
Ключевые слова: Асинхронное взаимодействие, Message-Oriented Middleware, Scala, Akka, актор
Annotation: The article focused on questions of implementation of Message-Oriented Middleware (MOM) in analytical and case-management system of urban transport. Author discloses main differences between synchronous (Remote Procedure Call, RPC) and asynchronous (MOM) interaction, describes implementation of MOM and forms its requirements.
Keywords: Asynchronous interaction, Message-oriented Middleware, Scala, Akka, actor

Современные программные системы становятся всё больше и больше. Они работают в сложном окружении, состоящем из различных подсистем, написанных на различных языках программирования с использованием различных технологий. Вместе с этим возрастают и требования, предъявляемые к системам. Они должны иметь высокую пропускную способность, надёжность круглосуточной работы и гарантию защищённости от злоумышленников. В таких условиях традиционная модель удалённого вызова процедур (RPC, RemoteProcedureCall) не справляется с предъявленными к системе требованиями.

На замену RPC появился новый механизм – MOM (Message-OrientedMiddleware), который предоставляет возможность общения между различными подсистемами. MOM является основой, на которой создаются распределённые системы. MOM выступает в роли посредника между отправителем и получателем.

В распределённых компьютерных средах преобладают две основных модели взаимодействия – синхронная, на которой базируется RPC, и асинхронная, на которой базируется MOM.

Когда функция или метод вызываются с использованием синхронной модели взаимодействия (Рис.1), происходит передача управления вызываемому объекту и ход выполнения вызвавшей программы вынужден приостановиться. В таком случае подсистемы не являются независимыми.

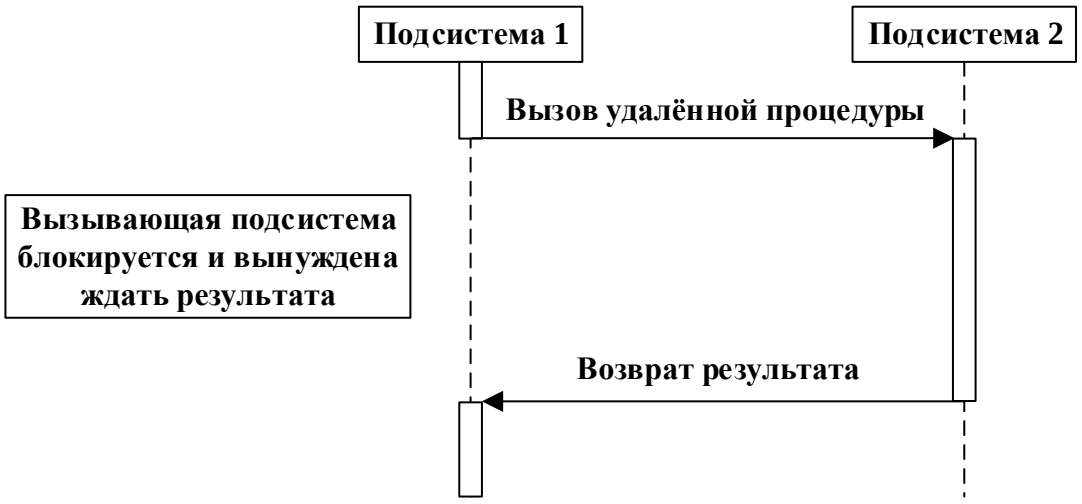


Рис.1. Синхронная модель взаимодействия

Асинхронная модель взаимодействия (Рис.2) позволяет вызывающей программе продолжать свой ход работы. Это позволяет подсистеме функционировать не зависимо от результата удалённого вызова. В этом случае нет гарантии того, что вызывающийся код начнёт работать сразу после вызова. Данная модель нуждается в посреднике между подсистемами, которым является МОМ [1].

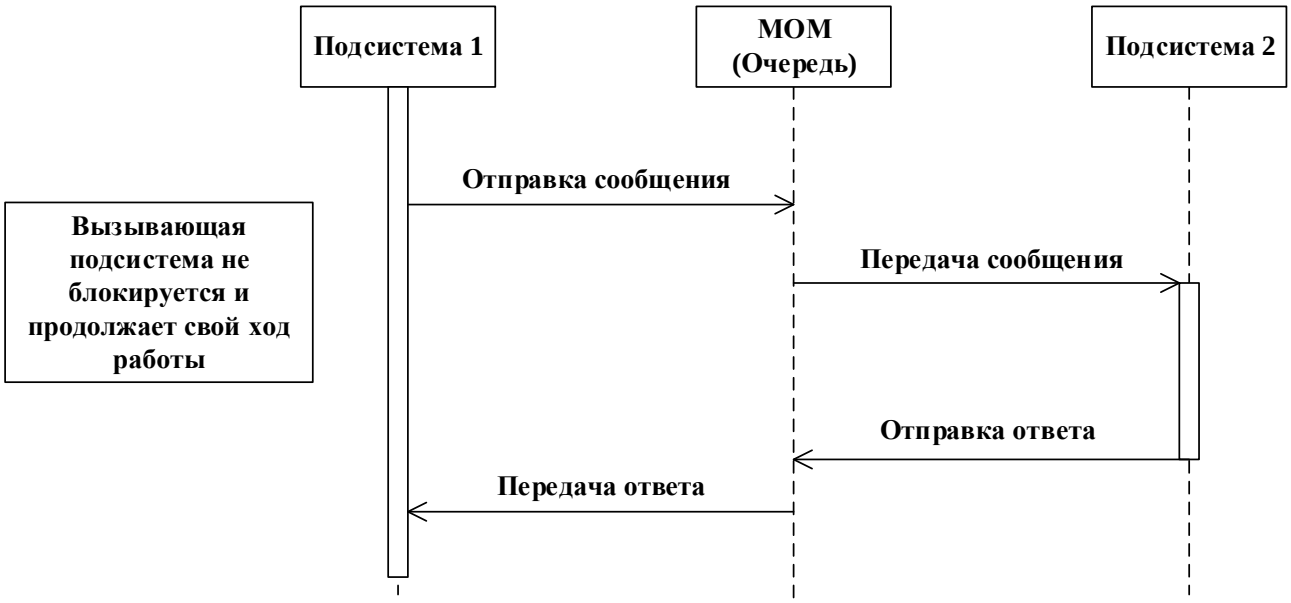


Рис.2. Асинхронная модель взаимодействия

В рамках данной статьи рассматривается реализация подсистемы обмена сообщениями (МОМ) в системе анализа и управления городским транспортом, которая выступает посредником между клиентами (мобильными приложениями) и серверной частью.

К подсистеме предъявлены следующие требования:

1) Надежность. Поскольку связь между подсистемой и клиентами происходит через мобильное интернет-соединение с непостоянным качеством сигнала, нужно обеспечить гарантированную доставку сообщения от сервера к клиенту, даже если в момент отправления сообщения клиент был в зоне плохого приёма сигнала.

2) Защищённость. Сообщения должны быть переданы получателю и только ему. Недопустимо, чтобы третье лицо имело доступ к информации, содержащейся в сообщении.

Подсистема должна также выполнять сбор статистических данных по качеству доставки сообщений в различных местах города.

Для реализации был выбран язык программирования Scala и один из самых популярных веб-фреймворков для Java/Scala– PlayFramework. Scala совмещает в себе парадигмы объектно-ориентированного подхода и функционального программирования, это позволяет проводить декомпозицию предметной области на объекты (как в обычных ООП-языках), и, в то же время, использовать всю мощь функционального подхода (простота и выразительность функциональных программ, отсутствие побочных эффектов, что позволяет легче писать многопоточный код и т.д.). Код на языке Scala компилируется в код на языке Java и исполняется на виртуальной машине JVM. Это позволяет запускать Scala-программы на всех устройствах, где поддерживается JVM. Также это позволяет использовать огромное количество уже существующих программных средств и библиотек, изначально разрабатываемых для языка Java. Всё это делает язык Scala подходящим под современные нужды создания программных продуктов. Поскольку подсистема должна параллельно обрабатывать большое количество запросов, было принято решение использовать модель

акторов и её реализация для Scala – Akka. Модель акторов представляет собой математическую модель параллельных вычислений, которая трактует понятие «актор» как универсальный примитив параллельного расчёта. Модель акторов схожа с ООП подходом, где «всё – это объект», только в данном случае «всё – это актор». Отличие заключается в том, что в объектно-ориентированном подходе программы, как правило, выполняются последовательно, а в модели акторов вычисления проводятся параллельно [2].

Связь подсистемы с клиентами (мобильными приложениями) и серверной частью происходит посредством HTTP запросов. Клиент отправляет POST-запрос с целью передачи информации о транспортном средстве на сервер (Рис.3). В свою очередь подсистема передаёт сообщение на сервер, а клиенту отправляет ответ об успешном завершении передачи. Клиент ждёт этого ответа заданной количество времени. Если, по истечении этого времени, ответ не получен, сообщение считается потерянным по причине плохой связи и происходит повторная отправка сообщения. Подсистема ведет учёт всех принятых сообщений, и, если первоначальное сообщение дошло, а ответ потерялся, повторно принятое сообщение не будет передано на сервер. Таким образом реализуется надёжность передачи сообщения.

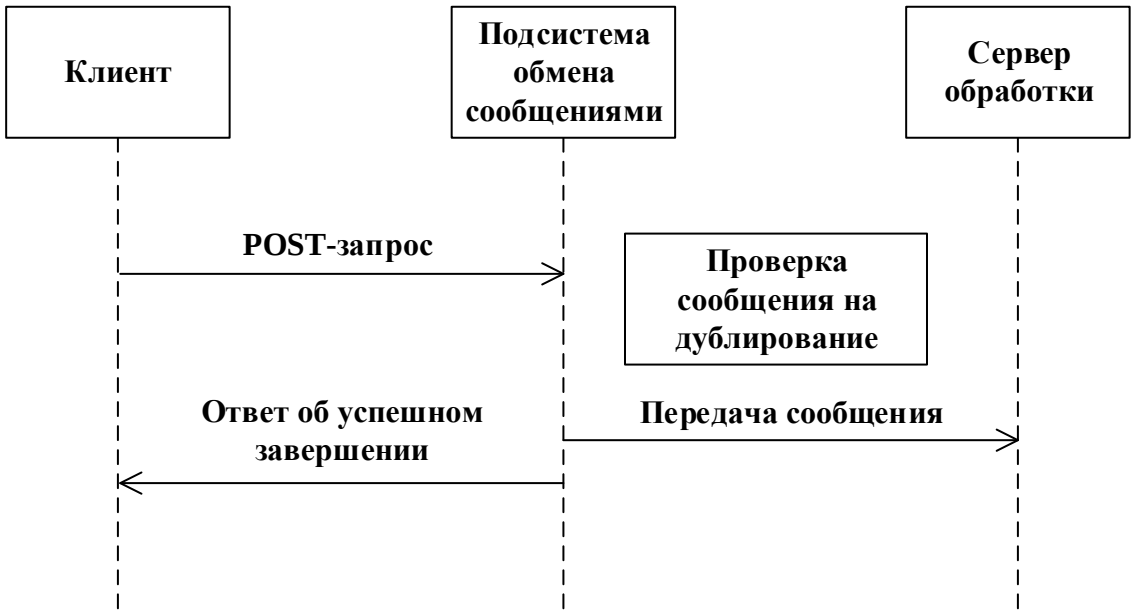


Рис.3. Пример POST-запроса

Когда сервер отправляет сообщение клиенту, оно попадает в почтовый ящик (mailbox) этого клиента и находится там, пока его не запросят. Клиенту предоставляется два способа взаимодействия со своим почтовым ящиком с помощью GET-запросов (Рис.4): узнать, сколько в ящике непрочитанных сообщений и получить самое раннее сообщение из всех непрочитанных. В случае с GET-запросами, так же, как и с POST, на клиенте происходит ожидание ответа в течении определенного интервала времени, после чего, если ответ не было получен, происходит повторная отправка запроса.

Для обеспечения защиты от третьих лиц каждому клиенту при аутентификации в подсистеме выдается уникальный токен, который действителен до конца рабочей сессии. Клиент указывает этот токен в заголовках к каждому HTTP-запросу и используя информацию из заголовка подсистема выполняет авторизацию запроса. Сбор статистики по качеству связи мобильного интернет-соединения основан на учёте дублирующихся запросов, которые происходят тогда, когда клиент отправил запрос, запрос дошел до подсистемы, но, из-за проблем со связью, ответ подсистемы не дошёл и клиент отправил запрос повторно.

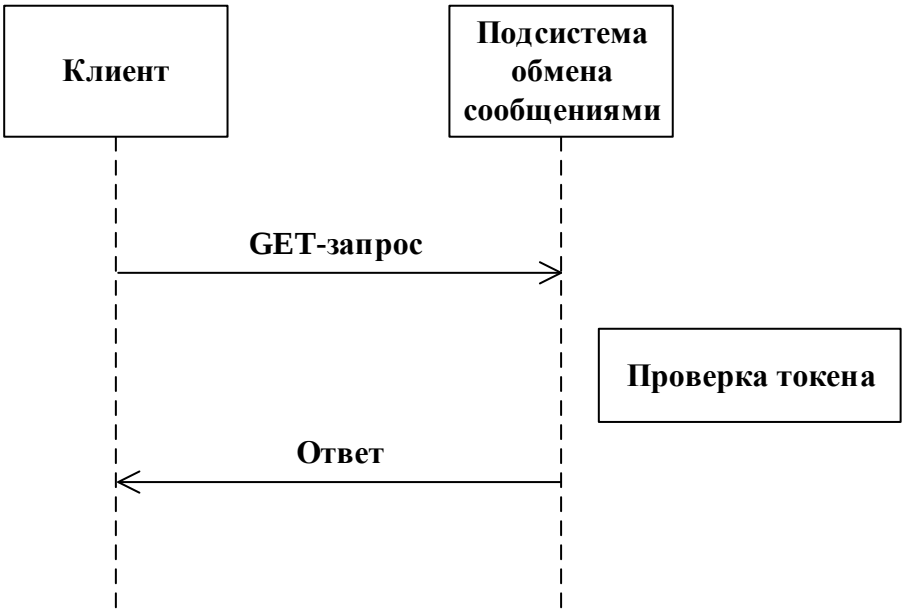


Рис.4. Пример GET-запроса

Таким образом, была реализована подсистема обмена сообщениями, которая представляет собой реализацию механизма Message-Oriented Middleware, и достигнуты поставленные цели, а именно: обеспечена надёжность доставки сообщений, реализована защита от третьих лиц и предусмотрен сбор статистики по качеству связи мобильного интернет-соединения.

Библиографический список

1. Qusay H. Mahmoud. Middleware for Communications [Электронный ресурс]. URL: http://www.edwardcurry.org/publications/curry_MfC_MOM_04.pdf (дата обращения: 06.03.2016).
2. Typesafe Inc. Akka Scala Documentation, Release 2.4.2 [Электронный ресурс]. URL: http://doc.akka.io/docs/akka/2.4.2/AkkaScala.pdf (дата обращения: 06.03.2016).

УДК 004.9
М.М. Маруга, магистр, студент 5-го курса.
Научный руководитель: И.С. Грундский, проф. техн. наук, доц.
Донецкий национальный технический институт

СОВРЕМЕННЫЕ ПРОБЛЕМЫ МОБИЛЬНОСТИ ПО И ВОЗМОЖНЫЕ МЕТОДЫ ИХ РЕШЕНИЯ

Аннотация: В статье рассмотрены современные средства переносимости программного обеспечения, эмуляций операционных систем, исполнения программного обеспечения.
Ключевые слова: мобильность программного обеспечения, кроссплатформенность.
Annotation: The article deals modern means of software portability, emulation of the operating systems, execution software .
Key words: mobile software, cross-platform.

На текущий момент существует множество различных операционных систем (ОС), которые выполняют свои задачи. ОС классифицируются на проприетарные и бесплатные (Open Source), распространяющимися под свободными лицензиями. Все они имеют свои недостатки и преимущества. У каждой из ОС свои особенности, своя архитектура, типы представления данных и т.д., под которую пишется определенный исполняемый код, выполняемый только на данной ОС. Поэтому в большинстве случаев невозможен перенос такого кода и его запуск на других ОС, порой даже на прежние версии той ОС, под которую он писался. В таких случаях возникают определенные проблемы мобильности приложений.

Кроссплатформенность (мобильность)- это способность работать более чем на одной аппаратной платформе и/или ОС.

На текущий момент придуманы такие средства и технологии реализации кроссплатформенности, как высокоуровневые языки программирования (Java, C#), интерпретируемые языки исполнения. В рамках мобильности приложений – все они выполняют свою задачу, но так же имеются основные проблемы их использования. Например, интерпретируемые языки, несмотря на свою простую реализацию, используют большее количество ресурсов машины, вследствие того, что им необходимо помимо выполнения самой программы, в режиме реального времени интерпретировать его для использования машиной. В случаях с высокоуровневыми языками переносимость приложений между разными операционными системами требует довольно много усилий. Это связано с различными взаимодействиями с приложениями и разнородностью функционала ОС.

Одна из далеко шагнувших в этом деле является компания Oracle, которая позволила создать язык высокого уровня, который будет исполняться на любой машине. Но для этого они поддерживают великое множество различных платформ, включающих в себя Unix, Windows, Mac и другие ОС. Они используют особенности каждой платформы для реализации интерпретатора Java байт-кода в исполняемый под данной платформой код (Рис.1). Идея очень хороша и имеет смысл, но в таком своем существовании крайне ненадежна и нестабильна. В каждой разрабатываемой ОС необходимо предусмотреть стандарты, которые будут вынесены единым центром стандартизации ОС и ПО, позволяющие интерпретировать и выполнять различные приложения, для различных ОС на уровне текущей ОС.

Основной проблемой программ, написанных на Java и C# является то, что использование данных языков требует установки дополнительных инструментов и средств виртуализации и исполнения (JDK, JVM). На данный момент они не являются лучшими решениями. Возникают постоянные проблемы невозможности запуска какого-либо ПО из-за старой версии Java на исполняемой машине либо постоянные сбои в работе Java машины.

Для выполнения какого-либо ПО, разработанного для других платформ, часто можно встретить виртуализацию или эмуляцию определенной ОС в текущей. Например, для этого используются технологии, предоставляемые программами Virtual Box, VMware Workstation. Хотя они довольно корректно будут работать, в ряде случаев возникают ошибки, которые связаны с эмуляцией данного ПО.

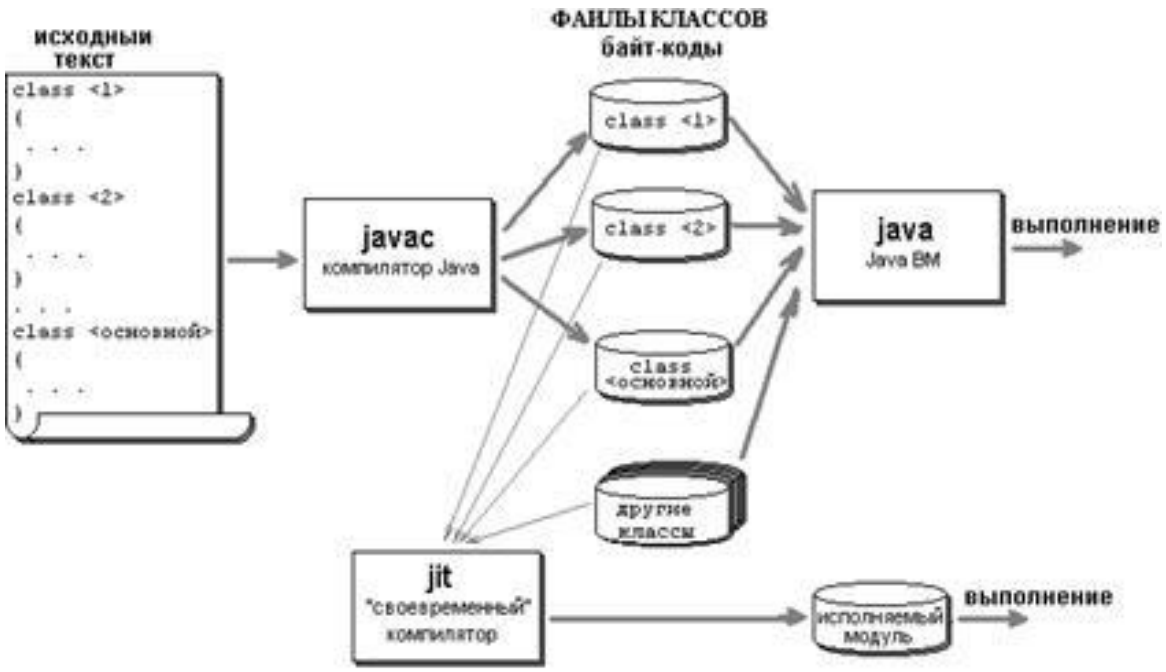


Рис.1. Принцип работы Java.

Иногда мы не можем видеть некоторые аппаратные средства или использовать потенциал машины полностью. Кроме того, как и в случае с Java, технология виртуализации требуют дополнительной установки, настройки, что занимает немало времени. Последний, решающий фактор, который следует учитывать: для работы данной технологии необходимо большее количество системных ресурсов, т.к. необходимо будет поддерживать основную ОС и программу эмуляции, которая, в свою очередь, будет брать ресурсы для эмулируемой ОС.

Также в ряде случаев решения проблемы кроссплатформенности является использование слоев совместимости. Например, wine и cygwin. Первый используется для совместимости WinApi для Unix, MacOS и прочих систем. Второй же представляет собой слой совместимости Linux для ОС Windows. Хотя в текущем этапе развития эти программы имеют достойную производительность, обусловленную тем, что они поддерживают переносимость на бинарном уровне, они имеют все те же недостатки что и для предыдущих способов: установка, настройка и некорректное использование программы при некоторых ситуациях.

Для решения данной проблемы необходимо учитывать невозможность переноса ПО из-за различных архитектур и строений ОС. Чтобы убрать этот недостаток, необходимо, чтобы разрабатываемые ОС имели встроенную прослойку, единый стандарт, который позволял бы выполнять код, написанный для других ОС непосредственно в этой машине. Основной проблемой, связанной с невозможностью перенесения ПО является то, что не происходит стандартизации на бинарном уровне. Такой подход не популярен. Попытки создания стандартизации была приняты для Unix систем консорциумом Open Software Foundation (OSF) и называлась OSF/1. Одним из наиболее успешных оказался стандарт для дистрибутивов Linux.

Для взаимодействия необходимо использовать единый стандартизационный центр разработки ПО, который будет иметь привилегированные права стандартизации различных типов, архитектур и т.п. для возможного их взаимодействия. В качестве примера можно взять консорциум 3W, который за все время своего существования позволил создать интернет таким, какой он есть. Какой бы ни существовал браузер, он будет придерживаться стандартов 3W. Как бы ни был реализован сайт, какие бы он не использовал CMS, настройки, базовый принцип реализации - все это опирается на стандарты, разработанные 3W.

В данной статье мы рассмотрели существующие системы, модели и технологии для работы кроссплатформенного ПО, эмуляцию систем, методы и возможные пути решения проблем мобильности ПО.

Библиографический список:

1. Хорошилов Алексей Владимирович Россия, Москва, Институт системного программирования РАН, Силаков Денис Владимирович Россия, Москва, Институт системного программирования РАН, МОБИЛЬНОСТЬ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ В СОВРЕМЕННЫХ УСЛОВИЯХ.
2. Дж. Рихтер CLR via C#. Программирование на платформе Microsoft .NET Framework 4.5 на языке C#. 4-е изд:: Питер, 2013.-896с.
3. David Thomas. The Pragmatic Programmer: From Journeyman to Master 1st Edition. Addison-Wesley Professional; 1 edition (October 30, 1999).
4. Герберт Шилдт. Java 8. Полное руководство, 9-е издание// Вильямс, ISBN: 978-5-8459-1918-2, 2015. -1377с.
5. Ссылка на рисунок: http://ermak.cs.nstu.ru/trans/Trans471.htm.

УДК 681.3
А.Ю. Поддубный, студент 4-го курса
Научные руководители: И.В. Дымченко, старший преподаватель; В.А. Строганов, старший преподаватель
Севастопольский государственный университет

РАЗРАБОТКА МОБИЛЬНОГО ПРИЛОЖЕНИЯ ДЛЯ АВТОМАТИЗАЦИИ ПОЛЕВЫХ АРХЕОЛОГИЧЕСКИХ РАБОТ

Аннотация: В настоящей работе представлено описание приложения для мобильного устройства (смартфона/планшета), позволяющего автоматизировать процесс сбора и передачи данных, полученных в результате археологических полевых работ.
Ключевые слова: археологические исследования, мобильное приложение, автоматизация, мобильное устройство
Annotation: This paper presents the description of the application for a mobile device (smartphone / tablet), allowing to automate the process of data collection and transmission, derived from archaeological fieldwork.
Keywords: archaeological research, mobile application, automation, mobile device

Введение. Принятие стратегии развития информационного общества в Российской Федерации, а также программы развития Федерального государственного бюджетного учреждения науки Института археологии Российской академии наук на 2015-2019 г.г., обязывает к привлечению самых современных технологий, для реализации задач, признанных приоритетными этими документами [1,2]. В том числе, это касается и вопросов разработки программного обеспечения для информатизации процессов научных исследований в целом и для археологической деятельности, в частности. Мобильные устройства с их огромным функционалом и различными сенсорами способны автоматизировать большинство работ, проводимых археологами-исследователями в полевых условиях, и обеспечить оперативные доступ и передачу информации. Таким образом, привлечение мобильных технологий и разработка программного обеспечения для мобильных устройств в этих целях, являются актуальными.

Цель статьи. Дать представление о деятельности полевых археологов-исследователей, описать процессы, которые были автоматизированы, и представить мобильное приложение, разработанное в ходе выполнения проекта «Археолог-Херсонес».

Основная часть. При выполнении полевых археологических работ (археологические разведки и раскопки), исследователям необходимо заполнять и составлять ряд документов: полевые дневники, полевые описи, опись массового материала, картографические и фотографические материалы [3]. Традиционно, в качестве полевого дневника используется тетрадь, каждая из страниц которой разделена на 2 части, на левую часть наносятся чертежи, рисунки и другие обозначения, а на правую – детальное описание; в качестве фото-устройства используется цифровой фотоаппарат, а вся картографическая информация наносится на бумажную карту [5].

И сходя из вышеизложенного описания рационально использовать вместо традиционных инструментов современные мобильные устройства – планшет или смартфон, поскольку размеры этих устройств, схожи с рекомендуемыми параметрами для полевого дневника.

Благодаря большому объему встроенной памяти, в современный планшет/смартфон можно поместить большое количество записей, что делает его независимым от площади и характера исследований. Планшет/смартфон имеет достаточные размеры, благодаря которым, текст удобно вводить руками, без помощи стилуса. Если же возникает потребность в данных приспособлениях, то следует отметить их доступность в большом диапазоне моделей и ценовых категорий. Современный планшет/смартфон среднего класса оснащен различными датчиками геолокации, что значительно упрощает процессы работы с картами и фиксации местоположения находок. Также, качество камер современных мобильных устройств уже не уступает цифровым фотоаппаратам, поскольку имеют широкий спектр настроек, таких как:

- регулировка выдержки;
- фокусировка;

- регулировка чувствительности;
- регулировка яркости;
- управление вспышкой;
- наличие сетки кадрирования;
- выбор соотношения сторон.

Для автоматизации процесса сбора и обработки материалов с помощью мобильного устройства, в рамках проекта ««Археолог – Херсонес», было разработано одноименное мобильное приложение.

Приложение имеет следующие разделы:

- полевой дневник;
- опись материала;
- карта находок;
- маршрут разведки.

Доступ к каждому из пунктов раздела осуществляется с помощью «Гамбургер меню».

Полевой дневник имеет интерфейс максимально схожий с традиционным (бумажным) полевым дневником. В начале работы необходимо внести всю информацию, идентифицирующую археологическую экспедицию, после чего становятся доступными для заполнения разделы полевого дневника. Для каждой записи, предусмотрена соответствующая форма. К записи могут быть прикреплены: изображения, видеофайл или аудиозапись, которые можно выбрать из ранее созданных и сохраненных в Галерее, или создать новый файл, средствами самого устройства, не выходя из приложения. Интерфейс полевого дневника изображен на рис.1-2.

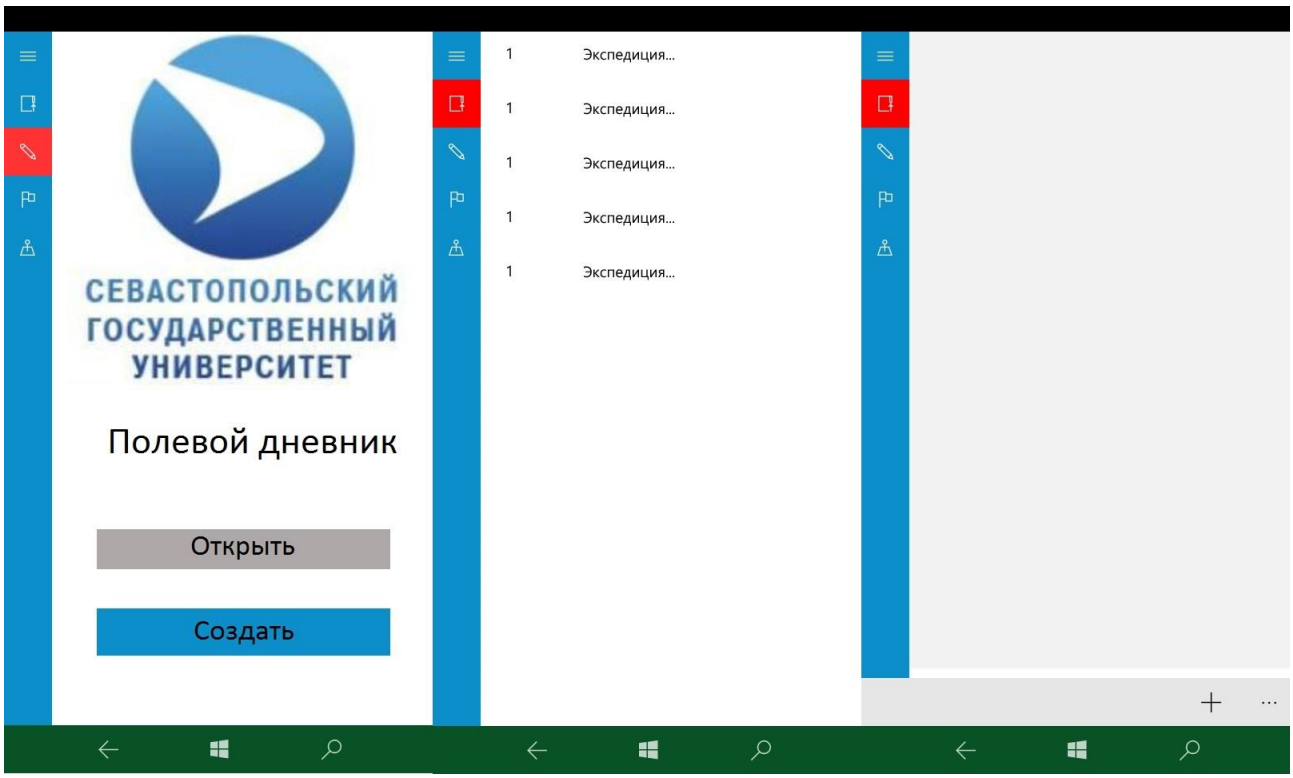


Рис. 1. Интерфейс раздела «Полевой дневник»

На рис. 2. Изображена форма для добавления записей в полевой дневник, а также форма для регистрации нового полевого дневника.

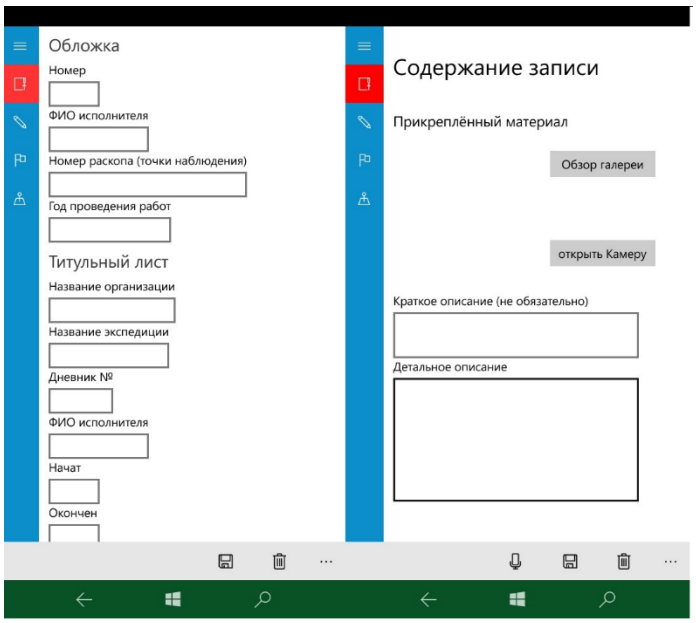


Рис. 2. Интерфейс раздела «Полевой дневник»

Раздел «Опись материала» позволяет получить доступ к описи массового материала и полевой описи. Доступ к описи массового материала ограничен, пользователь может только просматривать информацию, которая содержится в базе данных. Полевая опись содержит информацию о находках, которые были найдены на протяжении текущей экспедиции. Пользователь имеет возможность создать новую полевую опись, для чего потребуется заполнить соответствующую форму, а также редактировать, добавлять и удалять записи в текущей полевой описи. Интерфейс раздела «Опись материала» представлен на рис.3.

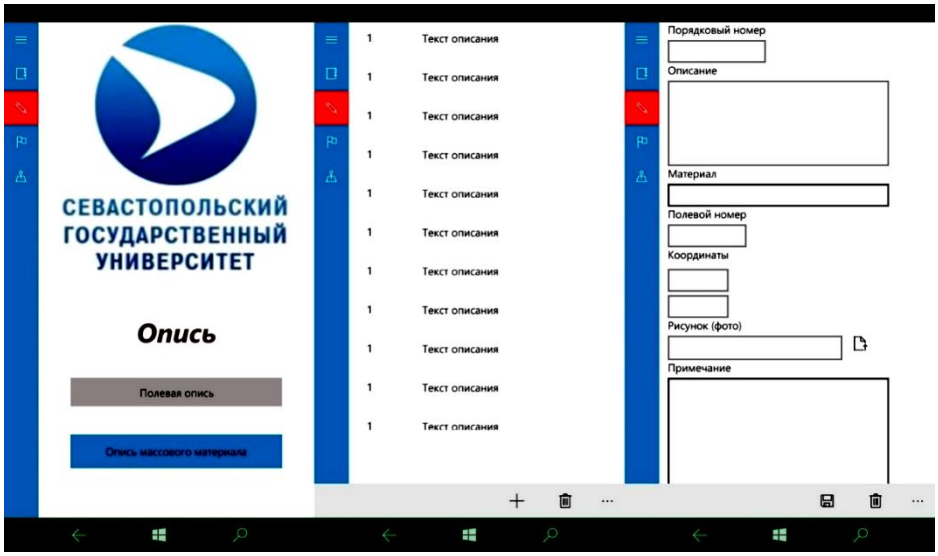


Рис. 3. Интерфейс раздела «Опись»

Раздел «Карта находок» позволяет отмечать на карте находки, которые были обнаружены в процессе экспедиции. Для этого в разделе текстовых записей необходимо заполнить поля, описывающие находку, в результате чего на карте появятся соответствующие отметки. Местоположение находки определяется автоматически с помощью технологии GPS/ГЛОНАСС. Интерфейс раздела «Карта находок» представлен на рис.4.

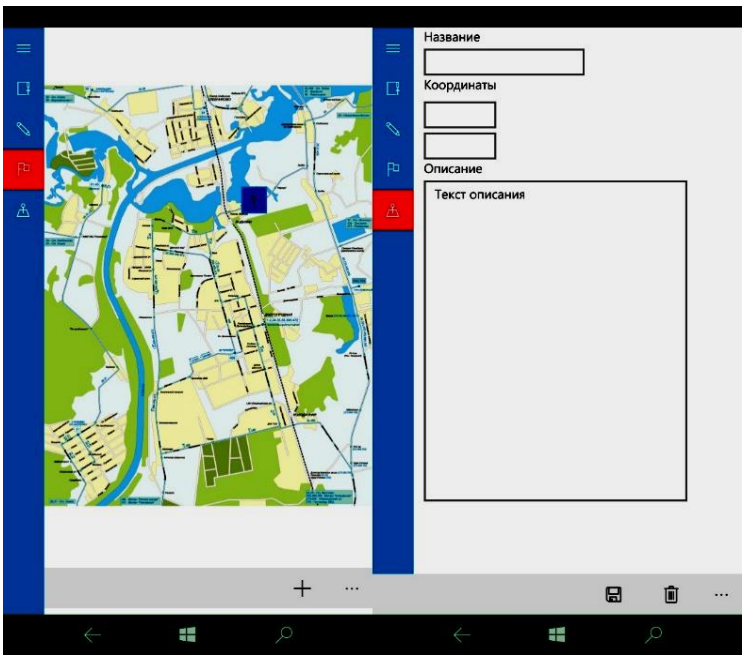


Рис. 4. Интерфейс раздела «Карта находок»

Раздел «Маршрут разведки» предназначен для автоматизации процесса археологической разведки и служит для нанесения координат и характеристик с детальным описанием точек маршрута. Координаты точек также устанавливаются в автоматическом режиме с помощью технологии GPS/ГЛОНАСС. Интерфейс раздела «Маршрут разведки» представлен на рис.5.

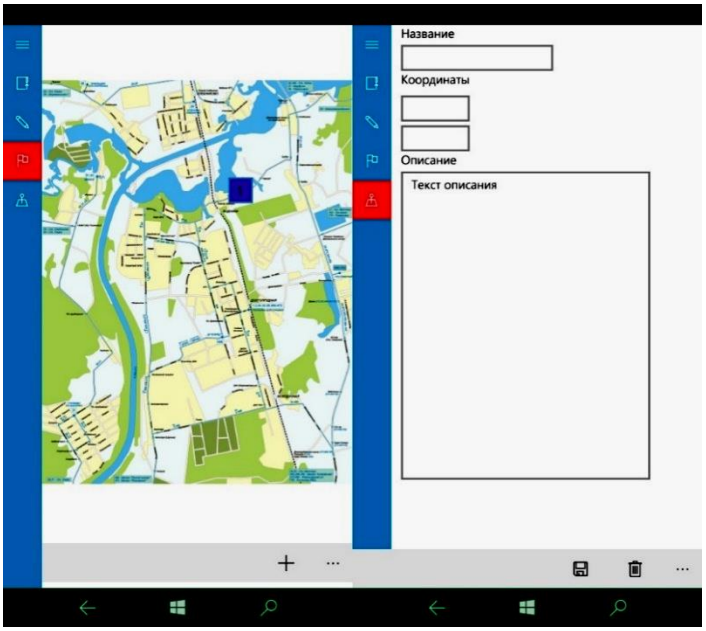


Рис.5. Интерфейс раздела «Маршрут разведки»

Было принято решение в качестве целевой платформы приложения использовать WindowsPhone. Выбор обусловлен стремительным набором популярности в Республике Крым и городе Федерального значения Севастополе. По данным аналитической компании Strategy Analytics за 3-й квартал 2013 года WindowsPhone признана самой быстрорастущей мобильной операционной системой. Доля рынка для данной платформы за год увеличилась почти в 2 раза. По данным компании «Связной», за первые пять месяцев 2015 года продажи смартфонов под управлением ОС Windows Phone составили 690 тысяч штук, что составило восемь процентов

от общего объема продаж в России. При этом в количественном выражении реализация Windows-телефонов превысила продажи Apple iPhone впервые в этом году. Еще один аргумент в пользу выбора платформы WindowsPhone – рынок мобильных приложений еще не настолько велик, как у Android и iOS, вследствие чего там намного легче найти свободный сегмент. Также с выходом платформы Windows 10 данное приложение можно запустить не только на смартфоне, а также на планшете и настольном ПК.

Развертывание приложения осуществлялось на смартфоне Microsoft Lumia 535, который имеет следующие характеристики:

- Процессор - Четырехъядерный Snapdragon 200 (1.2 ГГц)
- Разрешение: 960 x 540 пикселей
- Камера - 5 Мп
- Оперативная память 1 ГБ
- 8 ГБ встроенной памяти
- Поддержка карт памяти MicroSD (до 128 ГБ)
- Bluetooth 4.0
- Wi-Fi 802.11 b/g/n
- GPS, A-GPS, GLONASS
- Емкость аккумулятора: 1905 мА*ч
- Операционная система Windows 10 Mobile

Для разработки мобильного приложения была выбрана бесплатная среда разработчика Visual Studio 2015 Community Edition. В качестве языка программирования был выбран C# [4].

Мобильное приложение использует ресурсы бесплатного картографического сервиса Bing Maps [6].

Организации хранения данных осуществляется при помощи свободно распространяемых СУБД PostgreSQL с расширением PostGis.

Выводы и дальнейшее направление исследований. В процессе работы над проектом были изучены процессы проведения полевых археологических работ, обоснована необходимость привлечения современных информационных технологий для их автоматизации. Разработано мобильное приложение для смартфона Microsoft Lumia 535 с операционной системой Windows 10 Mobile.

В дальнейшем планируется тестирование приложения в полевых условиях, расширение функционала по требованиям археологов-исследователей, а также портирование на платформу Android OS, которая пока является несомненным лидером рынка мобильных операционных систем в России и Крыму.

Библиографический список

1. Стратегия развития информационного общества в Российской Федерации (утв. Президентом РФ 07.02.2008 № Пр-212) Режим доступа – http://www.consultant.ru/document/cons_doc_LAW_92004/ © Консультант Плюс, 1992-2016.
2. Программа развития Федерального государственного бюджетного учреждения науки Института археологии Российской академии наук на 2015-2019 г.г. Режим доступа – http://archaeolog.ru/?id=11.
3. Положение о порядке проведения археологических полевых работ и составления научной отчетной документации (утверждено постановлением Бюро Отделения историко-филологических наук Российской академии наук от 27 ноября 2013 г.) № 85. Режим доступа – http://www.archaeolog.ru/?id=36.
4. Пугачев С.В. Разработка приложений для Windows 8 на языке C#/ С.В. Пугачев, К.А. Кичинский. – СПб.: БХВ-Петербург, 2013. – 416 с.
5. Шакиров З.Г. Методы фиксации в археологии/ З.Г. Шакиров: Казань, 2015. – 110 с.
6. Сайт Microsoft Developer Network. Режим доступа – http://msdn.microsoft.com/ru-ru/

УДК 621.391
С.С. Райков, студент 4-го курса
Научный руководитель: В.С. Чернега, канд. техн. наук, доц.
Севастопольской государственный университет

УНИВЕРСАЛЬНОЕ УСТРОЙСТВО СОПРЯЖЕНИЯ ДЛЯ СИСТЕМЫ ИНФОРМАЦИОННОЙ ПОДДЕРЖКИ РАБОТЫ ХИРУРГА

Аннотация: описана возможная реализация системы информационной поддержки работы хирурга во время проведения операции
Ключевые слова: микроконтроллер, хирургия, сопряжение, хирургическаяоперация, устройство, USB, STM32.
Annotation: describes a possible implementation of the system of information support of the work of the surgeon during the operation.
Key words: microcontroller, surgery, pairing, surgical operation, device, USB, STM32.

Операционная современной больницы насыщена большим количеством разнородного и разнотипного электронного оборудования. Так, например, в операционной урологического отделения имеется компьютеризированный комплекс анестезиолога, гольмиевый лазер, служащий для резания, и коагуляция биотканей, наноимпульсныйлитотриптор, предназначенный для контактного разрушения органоминеральных конкрементов во внутренних полых органах человека и другие устройства. За пультом управления таких устройств находится врач-ассистент или медсестра, которые устанавливают рабочие параметры устройств по командам оперирующего хирурга, а также сообщают о текущих параметрах по его требованию. В процессе операции хирург вынужден постоянно отвлекаться на запрос текущих параметров и подачи команд на их изменение. С целью повышения эффективности проведения операции все электронные аппараты операционной целесообразно объединить в единую информационную систему, в которой вся необходимая хирургу информация отображалась на центральном мониторе, расположенном в поле зрения хирурга, а управление параметрами медицинской аппаратуры осуществлялось голосовыми командами (или с помощью манипуляторов), поступающими на центральный компьютер информационной системы.

В настоящее время существуют системы информационной поддержки работы хирурга во время проведения операции [1]. Однако в таких системах по-прежнему информация хирурга о текущих параметрах больного и работы оборудования осуществляется обслуживающего его персоналом. Одна из основных проблем, затрудняющая объединение медицинской аппаратуры операционной в единую систему, состоит в наличии в медицинской аппаратуре различных интерфейсов (RS-232, RS-485, CAN и пр.), не совместимых между собой.

Целью настоящей работы является создание универсального устройства сопряжения, позволяющего осуществлять съем показаний с медицинской аппаратуры и вводить ее в центральный компьютер информационной системы по универсальному интерфейсу USB. Выбор данного интерфейса обусловлен его распространенностью (большинство современных компьютеров поддерживает USB версии 2.0), относительной дешевизной соединительных кабелей и высокой скорости передачи данных (гораздо выше, чем у вышеперечисленных интерфейсов – до 480Мбит/с).

Устройство реализовано на базе однокристальной микро-ЭВМ (микроконтроллере) типа STM32. Микроконтроллеры данного семейства являются одними из наиболее перспективных на рынке 32-битных микроконтроллеров благодаря своим высоким характеристикам приотносительно низкой цене. В отличии от микроконтроллеров меньшей разрядности (8 и 16 бит), они обеспечиваютбольшую точность вычислений и работу с длинными типами данных, что позволяет в полной мере использовать такие языки какC и C++.

Структурная схема такой системыизображена на рисунке 1.

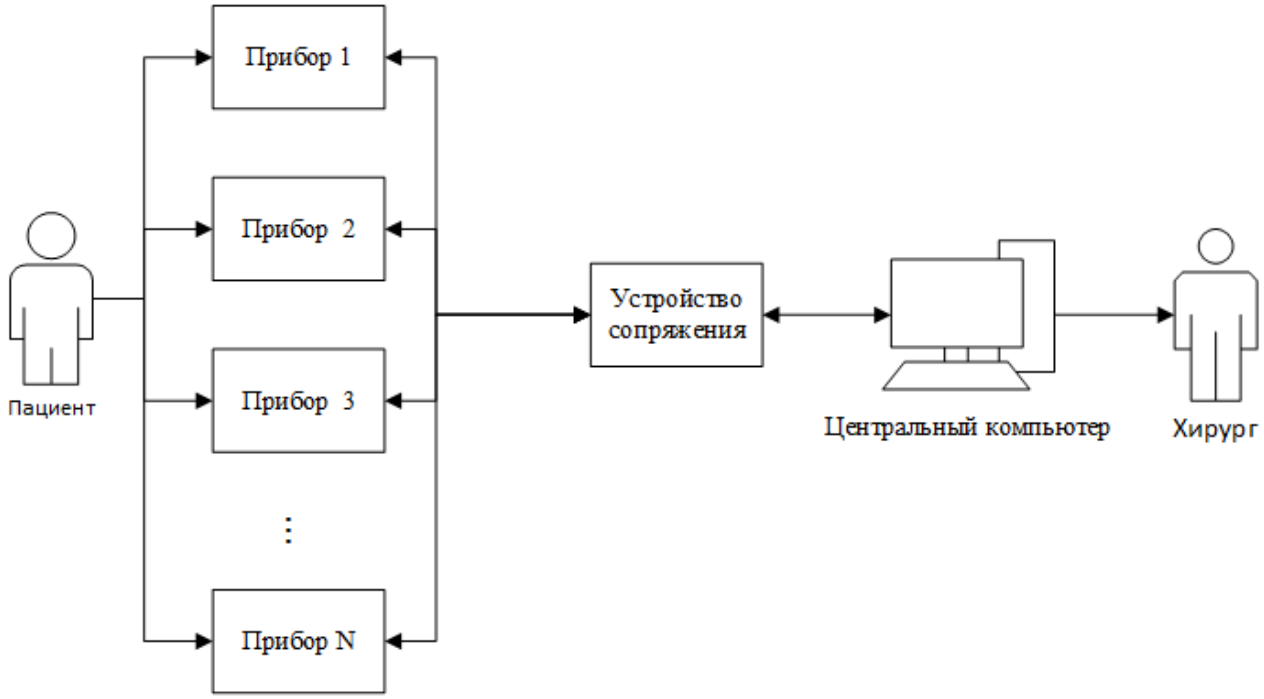


Рис.1. Структурная схема системы

На данной схеме, к пациенту подсоединены различные медицинские приборы, которые анализируют его состояние и поддерживают жизнедеятельность (например, аппарат искусственного дыхания). Посредством устройства сопряжения происходит обмен данными между медицинскими приборами и центральным компьютером – на компьютер хирурга выводятся показания приборов и с него же производится управление.

На физическом уровне, обмен данными производится через соответствующие интерфейсы – USB (только между центральным компьютером и устройством сопряжения), RS-232, CAN, RS-482 и RS-422, что требует кабельного соединения между медицинскими приборами и устройством сопряжения.

Данную систему можно легко модифицировать, если работе хирурга по каким-либо причинам будет мешать USB кабель, ведущий от устройства сопряжения к центральному компьютеру. Для этого можно на стороне отправителя и получателя установить Wi-fi адаптеры, которые будут передавать данные по радиоканалу. Данные меры также стоит предпринять, если длина USB кабеля составляет более пяти метров (ограничения стандарта USB) и нет возможности устанавливать USB-хабы. В том случае, если есть подобная возможность, то суммарную длину кабеля можно увеличить до тридцати метров.

Таким образом, перечень обязательных элементов устройствасопряжения имеет следующий вид.

1. Порты CAN, RS-232, RS-422, RS-485, USB.
2. Преобразователь логических уровней RS-232 ->USART – MAX3160(или ее аналоги)
3. Преобразователь логических уровней RS-485 ->USART – MAX3161(или ее аналоги)
4. Преобразователь логических уровней RS-422 ->USART – MAX3162 (или ее аналоги)
5. Блок питания (необходим для работы микроконтроллера)
6. Микроконтроллер (STM32F401)

Помимо аппаратных средств, для работы системы разработано специальное программное обеспечение. Для микроконтроллера это его «прошивка» – программа, предназначенная для загрузки во flash-память (ППЗУ), а для центрального компьютера – оконное приложение, с помощью которого будут интерпретироваться и отображаться данные с медицинских приборов.

Программное обеспечение для микроконтроллера обеспечивает следующую функциональность.

1. Опознавание подключенных медицинских приборов
2. Обмен данными с медицинскими приборами по соответствующим протоколам (в данном случае – эмуляция)
3. Обмен данными с центральным компьютером
4. Реакция на разрыв соединения с одним или несколькими устройствами
5. Реакция на отказ оборудования

Последние два пункта особенно важны, так как недопустима ситуация, когда хирург наблюдает неправильные или старые показания приборов (например, последнее значение может быть занесено вбуфер и отправляться даже после отказа прибора).

Исходя из вышесказанного, можно заключить, что внедрение данного устройства позволит снизить влияние человеческого фактора на проведение хирургических операций, а также нагрузку на рабочий персонал хирургического отделения.

Библиографический список

1. Егошин М.А. Система информационной поддержки хирурга при проведении трансуретральной резекции предстательной железы /*М. А. Егошин, р. Г. Хафизов, Ю. Е. Гарипова //* информационно-управляющие системы № 3, 2011. – с.66-69.
2. Агуров П. В.Интерфейсы USB. Практика использования и программирования. — СПб. БХВ-Петербург, 2004. -576 с: ил.

УДК 004.415
Е.П.Рыбалкин, магистрант 2-го курса
Научный руководитель: Ю.Г. Шаталова, канд. техн. наук, доц.
Севастопольский государственный университет

СИСТЕМА ВИЗУАЛИЗАЦИИ ПРОЦЕССА ВЫПОЛНЕНИЯ АЛГОРИТМОВ

Аннотация: Предложена система визуализации процесса выполнения алгоритма. Сформулированы требования к системам визуализации. Разработана и описана структура системы. Представлен и описан графический интерфейс пользователя разработанной системы. Предложенная программная система позволяет ускорить процесс изучения различных алгоритмов.
Ключевые слова: алгоритм, визуализатор, визуализация, программная система, структура системы.
Annotation: Is offered visualization system of the process of algorithm execution.Presented requirements for visualization systems.Developed and described the structure of the system.Presents and describes a graphical user interface of the developed system.The proposed system allows to accelerate the process of studying different algorithms.
Keywords: algorithm, visualizer, visualization, software, structure of the system.

При изучении таких дисциплин как программирование или информатика, требуется знание различных алгоритмов. Часто эти алгоритмы являются весьма сложными и, поэтому, трудны для изучения. Одним из решений данной проблемы является применение систем визуализации алгоритмов. В общем случае, система визуализации алгоритмов — это программная система, в процессе работы которой на экране компьютера динамически демонстрируется применение алгоритма к выбранному набору данных. В настоящей статье рассматривается построение системы визуализации процесса выполнения алгоритма для использования в учебном процессе.

На данный момент существуют различные системы визуализации алгоритмов, рассмотрим некоторые из них. С более подробным обзором можно ознакомиться в работе [1].

BALSA является одной из таких систем, она была разработана в Brown University (США) и основана на визуализации программ. Эта система и опыт ее использования подробно описаны в работах [2, 3]. Визуализаторы в этой системе описываются на специально разработанном языке описания сценариев. Для этого требуется достаточно высокая квалификация. Также недостатком системы BALSA и систем, построенных на ее базе, является сложность описания визуализаторов и ручное задание действий, осуществляемых при обратном проходе по алгоритму. Другими недостатками является слабая переносимость и невозможность задания входных данных пользователем.

В Brown Univesity также была разработана система Tango. Визуализация в этой системе состоит в отображении структур данных, которыми оперирует визуализируемый алгоритм. Система Tango ориентирована на визуализацию данных, в ней сложно визуализировать операции, не связанные с изменением данных, в том числе отслеживать выполнение алгоритма.

Система визуализации алгоритмов Jeliot, разработанная в University of Joensuu (Финляндия), является Java-версией системы визуализации Eliot, разработанной там же. Для создания визуализируемых программ применяется язык Java. При визуализации Java-код преобразуется во внутреннее представление.Основным недостатком системы Jeliot является отсутствие возможности контроля визуализации и излишняя детализация всех изменений.

Polka – система визуализации, разработанная в Georgia Institute of Technology (США) [4]. Samba является графическим интерфейсом к системе Polka и образует с ней единую систему визуализации. Она специализирована на визуализации параллельных программ. Описание визуализатора выполняется на специальном языке описания сценариев.

Система визуализации данных Leonardo разработана в Universita di Roma «La Sapienza» (Италия) и имеет такие же недостатки, как и любая другая система визуализации данных, в Leonardo отсутствует непосредственная возможность визуализации действий, не изменяющих данные.

Рассмотренные системы визуализации и другие существующие системы обладают различными недостатками, такими как плохая управляемость процесса визуализации, чрезмерная сложность создания визуализаторов алгоритма или плохая переносимость систем

на другие платформы, в частности, многие системы устарели и не поддерживают современные платформы.

Учитывая недостатки и достоинства существующих систем можно сформировать следующие требования, предъявляемые к системам визуализации алгоритмов, применяемых в учебном процессе.

1. Возможность ввода данных, на которых демонстрируется алгоритм.
2. Двунаправленность – система визуализации должна позволять совершать шаги алгоритма как вперед, так и назад.
3. Непрерывный и пошаговый режимы работы – система визуализации должна предоставлять возможность работы в пошаговом режиме и без вмешательства пользователя.
4. История – визуализатор должен обеспечивать возможность пошагового возврата назад, вплоть до начального состояния.
5. Отображение текущего хода выполнения алгоритма – система визуализации должна отображать не только изменения в данных, производимые исследуемым алгоритмом, но и другие действия, например, сравнения при сортировке.
6. Наличие комментариев для шагов алгоритма– на каждом шаге алгоритма должны отображаться комментарии, поясняющие производимое действие.
7. Простота использования –интуитивно понятный графический интерфейс.
8. Платформонезависимость – визуализатор должен работать на распространенных операционных системах.
9. Автономность – система не должна требовать подключения к сетевым ресурсам.

В соответствии с этими требованиями была разработана система визуализации для использования в учебном процессе. Программная система была реализована с использованием языка программирования Java для выполнения требования платформонезависимости. Структура системы представлена на рис. 1.

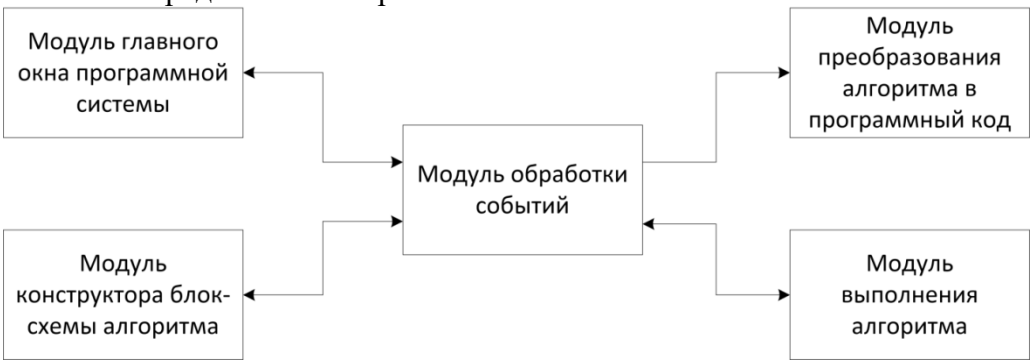


Рис. 1. Структура системы визуализации

На рис. 1 видно, что система состоит из пяти модулей. Рассмотрим каждый модуль.

Модуль главного окна программной системы реализованв виде двух классов Java: MainFrameи MainMenu. Первый класс описывает непосредственно окно программы и унаследован от класса JFrameстандартной библиотеки Swing. Второй описывает строку меню главного окна и унаследован от класса JMenuBar библиотеки Swing. Вместе эти классы описывают графический интерфейс пользователя и обеспечивают управление всей программной системой.

К графическому интерфейсу также можно отнести модуль конструктора блок-схемы алгоритма. В этом модулереализуется графическая составляющая главной функции системы – визуализация. Логика визуализатора обеспечивается другими модулями системы. Рассматриваемый модуль содержит четыре класса ElementPanel, MainPanel,TablePanel и TextPanel унаследованные от класса JPanel библиотеки Swing и один класс Elementунаследованный от класса JComponent библиотеки Swing.

Класс Element описывает элементы алгоритма, из которых строятся блок-схемы. Все типы блоков алгоритма унаследованы от этого класса. Также он обеспечивает визуальные эффекты при выполнении алгоритма, например, такие какподсветкаэлемента блок-схемы

натекущем шаге. Класс ElementPanel описывает графическую панель, которая содержит все типы блоков для составления алгоритма. Пользователь перемещает блоки из этой панели врабочую область, в которой собирается блок-схема. Эта область описывается классом MainPanel, в котором реализована возможность перемещения элементов и соединения их между собой для создания блок-схемы. Панель TablePanel является контейнером для таблицы описываемым классом JTablebиблиотеки Swing. Эта таблица состоит из двух колонок: «Переменные» и «Значения». Как можно понять из названия колонок, таблица предназначена для ввода исходных данных алгоритма и отображения значений всех переменных алгоритма во время его выполнения, а так же после завершения.В классеTextPanel реализована панель с текстовым компонентом JTextArea библиотеки Swing. Эта панель предназначена для вывода сообщений во время выполнения алгоритма.

Модуль преобразования алгоритма в программный код отвечает за формирование класса, реализующего Java интерфейс Algorithm и компиляцию полученного класса в байт-код Java для последующего использования модулем выполнения алгоритма. Блок-схема алгоритма преобразуется в программный код путем сопоставления каждого блока алгоритма и операторов языка Java. При этом после каждого изменения данных или после участка кода соответствующего одному элементу блок-схемы в код вставляется функция, оповещающая всех слушателей событий типа AlgorithmEvent унаследованного от класса EventObject стандартной библиотеки. Объект события, передаваемый слушателю, содержит информацию необходимую для визуализации всех изменений при выполнении исследуемого алгоритма.

Модуль выполнения алгоритма по команде пользователя, полученной через графический интерфейс, загружает класс, сформированный вышеописанным модулем. Затем создает экземпляр этого класса и добавляет для него слушателя событий типа AlgorithmEvent. После чего выполняет метод startAlgorithm() интерфейса Algorithm для созданного объекта. Кроме того создаются структуры данных для хранения состояния текущего шага алгоритма и всех предыдущих для обеспечения требования двунаправленности. На этом работа модуля выполнения алгоритма заканчивается.

Вышеописанные модули преобразования и выполнения алгоритмаиспользуют стандартную систему обработки событий Java, но при этом они не могут работать в одном потоке со слушателями стандартных событий. Выполнение этих модулей приведет к блокированию всего графического интерфейса на время их работы, а обращение к объектам графического интерфейса не из потока рассылки событий может привести к ошибке, так как большинство методов этих объектов не обладают встроенной синхронизацией[5]. Для решения этой проблемы применяются методы invokeLater() и invokeAndWait() класса EventQueue библиотеки AWT. Этим методам передается ссылка на интерфейс Runnable, метод run() этого интерфейса будет выполнен потоком рассылки событий[5].

Рассмотрев компоненты системы визуализации, описанные выше, можно перейти к модулю обработки событий, именно этот модуль обеспечивает связь всех компонентов между собой. Этот модуль состоит из четырех классов: ElementListener, MenuListener, AlgorithmListener и AlgorithmEvent. Первые три класса описывают слушателей событий, а последний класс описывает нестандартное событие, генерируемые в методе startAlgorithm() интерфейса Algorithm при выполнении алгоритма.

Класс ElementListener реализует интерфейс MouseInputListener. В этом классе обрабатываются события от мыши, возникшие в компонентах Element. Эти события отслеживаются для осуществления возможности перетаскивания блоков алгоритмов из панели с элементами в рабочую панель, а также для перемещения блоков элементов в рабочей панели и составления алгоритма. MenuListener реализует интерфейс ActionListener и обеспечивает реакцию системы на действия пользователя, связанные со стандартными компонентами графического интерфейса. Класс AlgorithmListener является слушателем событий типа AlgorithmEvent. Именно этот класс связывает графическое представление алгоритма и сформированный программный код. Как было описано выше, при выполнении

этого кода генерируются события содержащие сведения об изменившихся данных и текущем этапе выполнения. На основании этих сведений изменяется графическое представление алгоритма.

Главное окно описанной программной системы приведено на рис. 2.

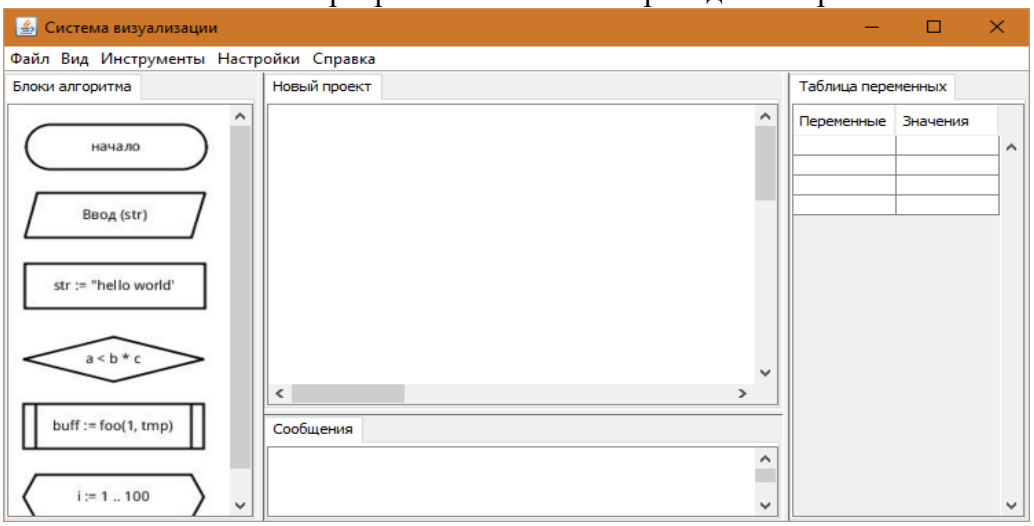


Рис. 2. Главное окно программной системы

Как видно на рис. 2, окно разделено на четыре основные области: строка меню, панель с элементами алгоритмов, основная рабочая панель, таблица переменных алгоритма и поле для вывода сообщений. Строка меню содержит элементы графического интерфейса, позволяющие управлять процессом визуализации алгоритмов и всей программной системой в целом. Панель с элементами алгоритмов содержит блоки, из которых строятся блок-схемы алгоритмов. Нужные блоки перетаскиваются в основную рабочую панель, где и строится исследуемый алгоритм. Таблица переменных позволяет задавать входные данные для алгоритма и просматривать изменения всех данных в процессе выполнения алгоритма, а также после его завершения. Во время выполнения в поле для сообщений выводятся комментарии к каждому шагу алгоритма и значимым действиям пользователя.

Представленная в данной статье система визуализации процесса выполнения алгоритма отвечает всем поставленным требованиям. Система позволяет из стандартных элементов строить блок-схемы алгоритмов и преобразовывать их в программный код. Посредством событийной модели выполняемый код связывается с графическим представлением исследуемого алгоритма и при изменении данных в программе изменяется и графическое представление. Система визуализации позволяет управлять выполняемым алгоритмом и отслеживать все изменения переменных алгоритма.

В дальнейшем планируется добавление к системе модуля построения блок-схемы алгоритма по программному коду, расширение возможностей настройки системы и добавление панелей инструментов для повышения удобства использования программы. Также планируется ввести возможность визуализации параллельных алгоритмов.

Библиографическийсписок

1. Wilson J., Aiken R. Katz I. Review of animation systems for algorithm understanding/ SIGCSE’96, Proceedings of the 1st conference on integrating technology into computer science education, Volume 28 Issue SI.
2. Brown M. Algorithm Animation. MIT Press, Cambridge, Massachussets, 1988.
3. Brown M., Sedgewick R. A system for Algorithm Animation / Computer Graphics, Proceedings of the 11th annual conference on Computer graphics and interactive techniques, July 1984.
4. Stasko J. A methodology for building Application-Specific Visualizations of Parallel Programs// Journal of Parallel and Distributed Computing. 1993, № 18.
5. Портянкин И.А. Swing. Эффектные пользовательские интерфейсы/И.А. Портянкин – СПб.:Питер, 2005. – 528с.

УДК 681.3
Н.Я. Спорыш, студент 4-го курса
А.О. Ульяненко, студент 4-го курса
Научный руководитель: И.В. Дымченко, старший преподаватель
Севастопольский государственный университет

КОНЦЕПЦИЯ ИНФОРМАЦИОННОЙ СИСТЕМЫ ДЛЯ АГРОПРОМЫШЛЕННОГО СЕКТОРА ЭКОНОМИКИ

Аннотация: В данной статье рассматривается концептуально решение по информатизации в сфере агропромышленности.
Ключевые слова: функциональная модель, информатизация, концепция, агропромышленность, мобильное приложение, веб-ресурс.
Annotation: In this article conceptual solution for informatization in the field of agro-industries is discussed.
Key words: functional model, information, concept, agro-industry, mobile app, web resource.

Введение. Информационные технологии в современном мире являются незаменимыми помощниками человека, их использование повысили производительность труда, они сделали операции, которые требовали усилий и времени простыми и быстрыми.

Информационное общество – концепция постиндустриального общества, в которой главными продуктами являются информация и знания. В таком обществе упор делается на знания и информацию, что делает людей более осознанными и развитыми, повышается уровень качества их жизни, что можно достигнуть путем широкомасштабной информатизации как общества в целом, так и отдельных субъектов различных уровней, и, в первую очередь, регионального уровня. Поэтому, 7 февраля 2008 года Президентом Российской Федерации была утверждена Стратегия развития информационного общества в Российской Федерации, являющаяся основой для подготовки и уточнения различных документов, определяющих основные цели и направления деятельности органов государственной власти, а также принципы и механизмы сотрудничества для развития информационного общества. Одним из подобных документов является Концепция информатизации Республики Крым, утвержденная Советом Министров Республики Крым 3 июня 2015 года. Она определяет основные цели по использованию информационно-коммуникационных технологий в Республике Крым, использование технологий для социально-экономического развития в тех направлениях, где применение информационно-коммуникационных технологий даст наибольшую отдачу: образование, здравоохранение, курорт и туризм, социальное обеспечение, труд и занятость, строительство, дорожное хозяйство, безопасность жизнедеятельности, транспорт, культура, энергетика, сельское хозяйство.

Цель статьи. Предложение решения для информатизации одной из отраслей агропромышленного комплекса – виноградарства, где в настоящее время, в большинстве случаев, для заполнения документации используются традиционные технологии (ручка, бумага).

Основная часть. В ходе исследования были изучены бизнес-процессы предприятия, занимающегося выращиванием винограда. Одним из основных видов работ предприятий данного типа является обработка земельных угодий, которой занимается производственная бригада, контролируемая бригадиром.

Бригадир производственной бригады в течении рабочего дня выполняет множество задач. Основные из них представлены на диаграмме декомпозиции работы “деятельность бригадира” на рисунке 1.

Из диаграммы декомпозиции видно, что в качестве инструментов вместо традиционных ручки и бумаги целесообразно использовать современные информационные технологии. Разрабатываемое решение предполагает комплексный подход, включающий разработку системы, которая состоит из трёх основных компонент: мобильного приложения для бригадира, с помощью которого будет автоматизирован процесс сбора данных, веб-ресурса для руководителя агрофирмы, который позволит осуществлять контроль за

состоянием выполненных работ и участвовать в принятии управленческих решений, а также серверного оборудования, которое обеспечит взаимодействие всех частей.

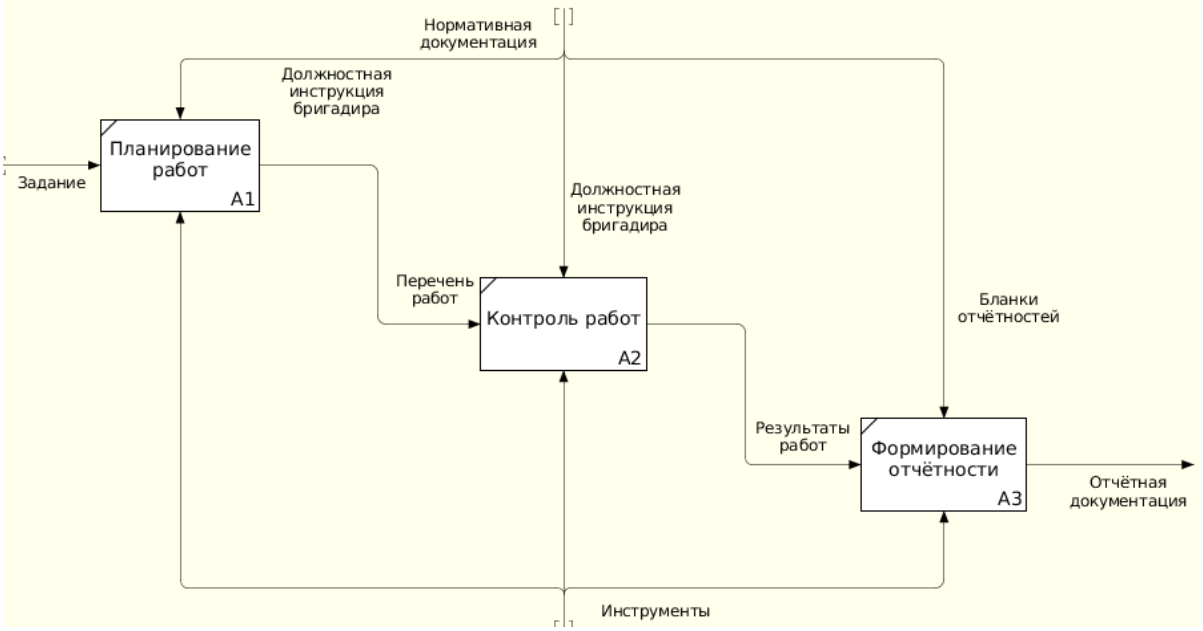


Рис. 1. Диаграмма декомпозиции работы "деятельность бригадира".

Концептуальная модель предлагаемой системы представлена на рисунке 2.

Рис. 2. Концептуальная модель системы

Так как большую часть своего рабочего дня бригадир находится непосредственно на участках, где выращивается виноград, то в качестве аппаратной платформы для установки приложения целесообразно использовать мобильное устройство - смартфон или планшет. Современные мобильные устройства обладают широким рядом функциональных возможностей, позволяют пользоваться интернетом, а также легки при транспортировке, поэтому они являются подходящим средством для работы в данных условиях.

К основным функциям приложения относятся:

1. Авторизация бригадира в системе
2. Выбор участка, на котором проводились работы
3. Указание типа работ: обрезка кустов, опрыскивание, сбор урожая
4. Указание количества выполненных работ
5. Отправка данных на сервер
6. Сохранение данных локально на устройстве, в случае неудачной отправки на сервер

Использование приложения позволит бригадиру отправлять на сервер актуальные данные об подконтрольных ему участках в режиме реального времени, находясь непосредственно на рабочем месте.

Для обработки данных предлагается использование специализированного компьютера - сервера, с предустановленным на нём программным обеспечением, для обработки запросов,

поступающих от мобильного приложения и базой данных, для хранения информации. Сервер выступает в виде промежуточного звена, которое производит обработку полученных от бригадира данных.

К основным функциям сервера относятся:

1. Обработка запросов, отправляемых мобильным приложением
2. Обработка запросов, отправляемых веб-ресурсом
3. Сохранение информации, поступающей от мобильного приложения, в базу данных
4. Выдача информации на веб-ресурс
5. Систематизация полученных данных

Для просмотра данных целесообразно использовать веб-ресурс, так как для доступа к нему нет необходимости устанавливать дополнительное программное обеспечение. В первую очередь, ресурс предназначен для сотрудников предприятия - руководителей отделов, бухгалтеров. На веб-ресурсе будет отображаться актуальная информация о текущем состоянии сельскохозяйственных угодий, принадлежащих предприятию, виде и количестве проделанных работ конкретной производственной бригадой. Для более наглядного представления информация будет систематизирована по участкам и отображаться на карте.

К основным функциям веб-ресурса относятся:

1. Авторизация пользователя
2. Получение актуальной информации от сервера
3. Отображение текущего состояния всех участков
4. Просмотр выполненных работ за определённый период конкретной бригадой
5. Отображение на карте участков, принадлежащих предприятию, с возможностью просмотра информации о их состоянии

Выводы и дальнейшее направление исследований. В данной статье, в рамках исследования для дипломного проекта, была проанализирована предметная область - виноградарство. Были выявлены процессы, подлежащие информатизации и автоматизации. Было предложено решение в виде мобильного приложения для бригадира и веб-ресурса для руководства. Дальнейшая работа над проектом предполагает реализацию и тестирование на базе предприятия «АртВин».

Библиографический список

1. "Стратегия развития информационного общества в Российской Федерации" (утв. Президентом РФ 07.02.2008 N Пр-212) http://www.consultant.ru/document/cons_doc_LAW_92004/ © 1997—2015 КонсультантПлюс
2. Концепция информатизации Республики Крым до 2018 года (утв. Советом министров Республики Крым от 03 июня 2015 года) http://rk.gov.ru/rus/info.php?id=622504
3. "Р 50.1.028-2001. Рекомендации по стандартизации. Информационные технологии поддержки жизненного цикла продукции. Методология функционального моделирования" (приняты и введены в действие Постановлением Госстандарта России от 02.07.2001 N 256-ст)

УДК 681.3
Ю.В. Увин, магистрант
Научный руководитель: В. С. Чернега, канд. техн. наук, доц.
Севастопольский государственный университет

ИССЛЕДОВАНИЕ ВЛИЯНИЯ ПАРАМЕТРОВ ВОКОДЕРА НА КАЧЕСТВО СИНТЕЗИРУЕМОЙ ИМ РЕЧИ

Аннотация: исследовано влияние параметров вокодера на основе линейного предсказания, преобразования Уолша-Адамара и косинусного преобразования на качество синтезируемой речи. Исследовано влияние параметров вокодера на величину задержки при вычислениях.
Ключевые слова: вокодер, сжатие речи, исследование параметров, синтез речи, цифровая обработка сигналов, преобразование Уолша-Адамара, косинусное преобразование.
Annotation: The research of how the parameters of the linear predictive vocoder, which uses Walsh-Hadamard transform or cosine transform, affect the speech quality was performed. The research of how the parameters of the vocoder affect the computing time of the algorithm was performed.
Key words: vocoder, speech compression, research of the parameters, speech synthesis, digital signal processing, Walsh-Hadamard transform, cosine transform.

На сегодняшний день в мире активно используется речевая коммуникация по средствам мобильных телефонов (GSM) или сети Интернет (Voice over IP). Для обслуживания как можно большего числа пользователей требуется обеспечивать низкую скорость цифрового потока (битрейт), например, 64 кбит/с для канала тональной частоты при частоте дискретизации 8 кГц. В настоящее время существуют различные системы сжатия речи (вокодеры). Работа большинства из них основана на линейном предсказании (модель авторегрессии). Сжатие речи предполагает представление цифрового речевого сигнала в виде параметров, которые передаются по каналу связи. Количество этих параметров меньше количества отсчетов цифрового речевого сигнала, за счет чего и осуществляется сжатие. На приемной стороне должен быть произведен синтез речевого сигнала на основе этих параметров [3].

Существующие алгоритмы сжатия речи имеют сложную структуру и требуют вычисления множества параметров, что требует дополнительные средства для использования этих алгоритмов, а также увеличивает задержку при вычислениях. Кроме того, современные алгоритмы сжатия речи предполагают передачу данных с высокими скоростями, что не всегда целесообразно и допустимо. В связи с этим в работе исследуется подходы к сжатию речи на основе линейного предсказания и дискретного преобразования, в частности преобразования Уолша-Адамара и косинусного преобразования. Существующие на данный момент научные работы не содержат результатов исследования зависимости величины задержки, а также скорости цифрового потока от различных параметров вокодера на основе остаточного сигнала.

Целями эксперимента является:

– проверка работоспособности алгоритма сжатия речи на основе линейного предсказания и дискретного преобразования Уолша-Адамара;

– сравнение алгоритмов сжатия речи на основе преобразования Уолша-Адамара и косинусного преобразования.

Задачи эксперимента:

– выбор критериев сравнения для алгоритмов сжатия речи;

– моделирование работы алгоритма и оценка параметров качества восстанавливаемых после сжатия речевых сигналов.

Методика проведения эксперимента:

1) оцифровка речевого сигнала, записываемого через микрофон, и сохранение его в звуковом файле;

2) считывание отсчётов сигнала из звукового файла и их кодирование (сжатие), состоящее в определении параметров сигнала;

3) восстановление (синтез) сигнала по определённым ранее параметрам;

4) применение алгоритма сжатия речи с различными задаваемыми параметрами; сравнение получаемых синтезированных речевых сигналов между собой.

Для сравнения речевых сигналов между собой применяются следующие характеристики [5]:

– коэффициент сжатия;
– отношение сигнал/шум (ОСШ);
– пиковое отношение сигнал/шум (ПОСШ);
– среднеквадратическая ошибка;
– сохранённая энергия сигнала;
– субъективная экспертная оценка.

В процессе исследований определялись:

– зависимость указанных характеристик от количества коэффициентов преобразования, а также от количества коэффициентов линейного предсказания;

– зависимость времени выполнения преобразования от количества коэффициентов преобразования, а также от количества коэффициентов линейного предсказания;

– субъективная оценка качества речи [1].

Для решения поставленной задачи рассматривалась модель сжатия речи на основе линейного предсказания. Эта модель требует определения коэффициентов линейного предсказания для сигнала, а также определения периода основного тона этого сигнала. С помощью этих коэффициентов и периода основного тона может быть осуществлен синтез речевого сигнала, другими словами, сигнал может быть восстановлен по своим параметрам [4]. Описанная модель основывается на принципе авторегрессии:

$$X_t = c + \sum_{i=1}^{p} a_i \cdot X_{t-i} + \varepsilon_t, \qquad (1)$$

где X_t — значение временного ряда в текущий момент времени; c — постоянная; a_i — коэффициенты авторегрессии; X_{t-i} — значение временного ряда в момент $(t - i)$; ε_t — белый шум.

Синтез (воспроизведение) речевого сигнала выполняется на основе коэффициентов линейного предсказания. Использование лишь одной модели линейного предсказания не обеспечивает приемлемое качество речи. Для повышения качества может быть использован остаточный сигнал [4], который вычисляется на основе коэффициентов линейного предсказания и исходного речевого сигнала. Остаточный сигнал также должен быть сжат. В процессе исследования рассматривалось сжатие остаточного сигнала с использованием дискретного преобразования Уолша-Адамара или дискретного косинусного преобразования. Сжатие остаточного сигнала производится путем передачи по каналу связи лишь части коэффициентов преобразования, поскольку большая часть энергии сигнала концентрируется в нескольких первых коэффициентах.

Ниже приведены расчетные соотношения для параметров, по которым производится сравнение речевых сигналов [5]. Коэффициент сжатия:

$$K_{сж} = \frac{L_o}{L_c}, \qquad (2)$$

где L_o — размер исходного речевого сигнала; L_c — размер сжатого сигнала, определяется в зависимости от количества бит, необходимых для его хранения и передачи.

Отношение сигнал/шум:

$$\text{ОСШ} = 10 \cdot \log_{10}\left(\frac{\sum_{i=1}^{N} x(i)^2}{\sum_{i=1}^{N} \left[x(i) - x'(i)\right]^2} \right), \qquad (3)$$

где $x(i)$ — исходный речевой сигнал; $x'(i)$ — синтезированный речевой сигнал; N — количество отсчётов в сигнале.

Чем выше отношение сигнал/шум, тем выше качество сигнала.

Пиковое отношение сигнал/шум:

$$\text{ПОСШ} = 10 \cdot \log_{10}\left(\frac{N \cdot (\max_X)^2}{\sum_{i=1}^{N}[x(i) - x'(i)]^2}\right), \quad (4)$$

где $(\max_X)^2$ — квадрат максимального абсолютного значения исходного сигнала. ПОСШ показывает отношение пикового уровня сигнала к шуму.

Нормированная среднеквадратическая ошибка:

$$\|e\| = \sqrt{\frac{\sum_{i=1}^{N}[x(i) - x'(i)]^2}{\sum_{i=1}^{N}[x(i) - \mu_x]^2}}, \quad (5)$$

где μ_x — среднее значение амплитуды исходного сигнала.

Сохранённая энергия сигнала:

$$E_c = \frac{\sum_{i=1}^{N} x'(i)^2}{\sum_{i=1}^{N} x(i)^2} \cdot 100\%, \quad (6)$$

Сохранённая энергия сигнала показывает, какое количество энергии сохраняется в синтезированном сигнале по сравнению с исходным сигналом.

На рисунке 1 показана модель вокодера для проведения эксперимента в системе MATLAB. На рисунке 1 под N_{LPC} понимается количество использованных в процессе вычисления коэффициентов линейного предсказания, а под N_{Tr} — количество использованных коэффициентов дискретного преобразования. С помощью этих двух параметров можно регулировать битрейт (скорость цифрового потока).

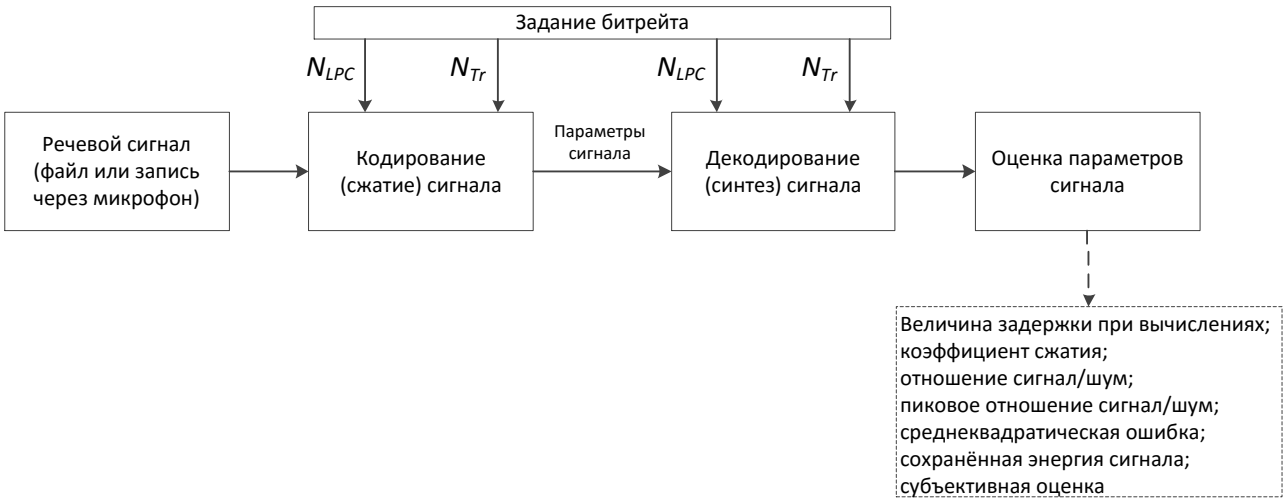


Рис. 1. Схема модели вокодера для проведения эксперимента

Входные параметры для проведения эксперимента:

– частота дискретизации: 16 кГц;
– размер кадра: 32 мс (512 отсчетов);

– количество коэффициентов линейного предсказания (N_{LPC}) при варьировании количества коэффициентов преобразования (N_{Tr}) от 10 до 200: N_{LPC} = 15;

– N_{Tr} = 100 при варьировании N_{LPC} от 10 до 200.

– речевой сигнал длительностью 9 секунд, в котором говорится о теории вероятностей.

Примечание: шаг при варьировании N_{LPC} или N_{Tr} непостоянен. Поэтому использовались следующие значения: 10, 25, 50, 70, 100, 120, 140, 150, 170, 180, 200.

В результате эксперимента было выявлено, что количество используемых коэффициентов дискретного преобразования не влияет на время выполнения этого преобразования. Это связано с тем, что при использовании неполного количества коэффициентов преобразования, неиспользуемые коэффициенты заполняются нулями. Данное действие выполняется, поскольку при обратном преобразовании необходимо, чтобы количество коэффициентов было равно количеству отсчётов сигнала в кадре.

Исследование также показало, что выполнение преобразования Уолша-Адамара занимает большее количество времени по сравнению с косинусным преобразованием. Это очевидно, поскольку преобразование Уолша-Адамара требует выполнения большего количества действий.

Оценивая коэффициент сжатия, было выявлено, что чем он больше, тем хуже качество синтезированного речевого сигнала. И наоборот — чем ниже коэффициент сжатия, тем лучше качество синтезированного речевого сигнала.

В результате проведения эксперимента объективные параметры (формулы 3 — 6) показали, что косинусное преобразование превосходит преобразование Уолша-Адамара по качеству синтезируемой речи. Кроме того, было замечено, что при изменении N_{LPC} и постоянном N_{Tr} объективные параметры для синтезированного речевого сигнала, полученного с использованием косинусного преобразования, практически не изменяются. При уменьшении же N_{Tr} и постоянном значении N_{LPC} было показано, что объективные параметры также уменьшаются. Другими словами, с увеличением степени сжатия происходит падение качества синтезируемого сигнала, согласно объективным параметрам.

Предыдущее исследование [2] проводилось для оценки субъективного качества речи в зависимости от количества коэффициентов косинусного преобразования и преобразования Уолша-Адамара (рисунок 2).

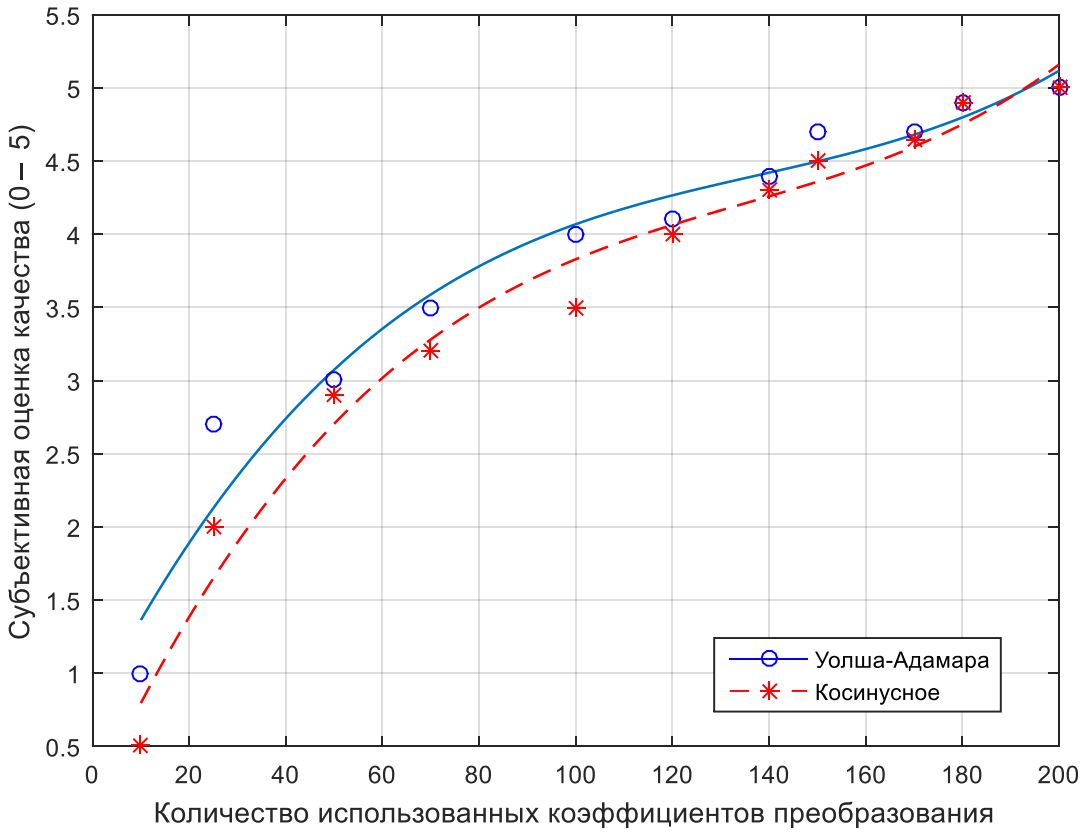


Рис. 2. Результаты субъективной оценки качества речи

На графике (рисунок 2) точки соответствуют осредненному значению оценки качества речи, полученной от трех экспертов. Эксперимент проводился таким образом, что оценке подвергались речевые сообщения, синтезированные с различным количеством коэффициентов дискретного преобразования.

Исследование показало, что преобразование Уолша-Адамара позволяет достичь лучшего результата при меньшем количестве используемых коэффициентов преобразования. Кроме того, как и в случае с оценкой сигналов с помощью объективных параметров, субъективная оценка показывает, что при увеличении степени сжатия, качество синтезируемого речевого сигнала падает.

Таким образом, имеется разница в проведении опытов на основании объективных и субъективных оценок. Однако качество речевых сигналов обычно оценивается субъективно.

Библиографический список

1. ГОСТ Р 51061-97. Системы низкоскоростной передачи речи по цифровым каналам. Параметры качества речи и методы измерений. — Москва: Изд-во стандартов, 1997. — 20 с.
2. Увин Ю. В., Чернега В. С. Система сжатия речевых сообщений для мобильной связи / Современные проблемы радиотехники и телекоммуникаций «РТ - 2015» : материалы 11-й междунар. молодежной науч.-техн. конф., Севастополь, 16 — 20 ноября 2015 г. / Севастоп. гос. ун-т; под ред. А. А. Савочкина. — Севастополь : Изд-во СевГУ, 2015. — С. 75
3. Шелухин О. И. Цифровая обработка и передача речи / О. И. Шелухин, Н. Ф. Лукьянцев — Москва.: Радио и связь, 2000. — 456 с.
4. Chu Wai C. Speech coding algorithms: Foundation and evolution of standardized coders [Электронный ресурс] / Wai C. Chu — USA: John Wiley & Sons, Inc, 2003 г. — 558 с. — Режим до-ступа: http://dsp-book.narod.ru/ChuSCA.pdf.
5. Smita Vatsa, Dr. O. P. Sahu Speech Compression Using Discrete Wavelet Transform and Discrete Cosine Transform [Электронный ресурс] / International Journal of Engineering Research & Technology (IJERT) Vol. 1 Issue 5, July — 2012. Режим доступа: http://goo.gl/ukPRmy.

УДК 004.7
И.А. Фетискин, магистрант гр. ИВТ/м-22о
Научный руководитель: Сапожников Н. Е. д.т.н., профессор
Севастопольский государственный университет

ИССЛЕДОВАНИЕ МЕТОДОВ И СИСТЕМ НАГРУЗОЧНОГО ТЕСТИРОВАНИЯ СЕРВЕРНОГО ОБОРУДОВАНИЯ В РАСПРЕДЕЛЕННЫХ СИСТЕМАХ

Аннотация: Статья посвящена актуальной на сегодняшний день проблеме несоответствия между ожидаемой и реальной производительностью серверного оборудования. При детальном рассмотрении существующих инструментов для нагрузочного тестирования выясняется, что оптимальные инструменты для решения поставленной задачи сложно подобрать.
Ключевые слова: нагрузочное тестирование, производительность, серверное оборудование
Annotation: Article is devoted to a mismatch problem actual today between the expected and actual productivity of server hardware. By detail reviewing of the existing tools for the load testing it is clarified that optimum for the solution of an objective it is difficult to pick up.
Key words:load testing, productivity, server hardware.

Введение

В век информационных технологий аппаратное и программное обеспечение просто обязано бесперебойно работать под колоссальными нагрузками. Любого рода проблемы, связанные с плохой производительностью, могут стать причиной отказа всей системы. В связи с этим, проведение качественного нагрузочного тестирования должно стать обязательным для обеспечения стабильности работы системы в целом [1].

Нагрузочное тестирование или тестирование производительности – это эмуляция работы на сервере многих пользователей сразу, такого рода тестирование максимально приближено к реальности и показывает реальную производительность устройства [2].

Цель работы

Провести исследование несоответствия между ожидаемой и реальной производительностью серверного оборудования, а также выявить слабые места и возможные проблемы при нагрузочном тестировании. Подобрать оптимальную конфигурацию серверного оборудования.

Описание ситуации в проблемной области

Возникают такие ситуации у клиентов, когда они хотят установить на сервер новые программные продукты или обновиться до более свежей версии программы. И здесь их подстерегают риски связанные с тем, как поведет себя система в целом, хватит ли ресурсов и мощностей у сервера для бесперебойной работы, и способна ли система к регенерации после сбоя.

Существующие разработки, их недостатки

Программное обеспечение HP LoadRunner помогает:

- тестировать различные приложения, включая самые современные мобильные и интернет - технологии, приложения ERP/CRM и множество старых систем;
- выполнять крупномасштабное тестирование с использованием минимального набора оборудования, включая сочетание физических и виртуальных сред, в том числе любую общедоступную облачную инфраструктуру;
- выявлять узкие места в инфраструктуре, влияющие на общую производительность, с использованием средств расширенного мониторинга и анализа, а также обеспечить соответствие новых или обновленных приложений требованиям предприятия в отношении производительности.

На рис.1 приведено окно программного продукта HP LoadRunner

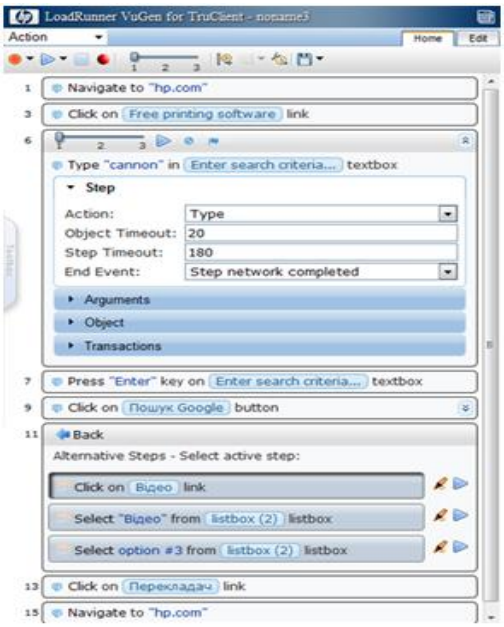


Рис.1. Окно программного продукта HP LoadRunner

К недостаткам данного программного обеспечения можно отнести: ресурсоемкость; платность использования.

Инструмент JMeter для проведения нагрузочного тестирования, разрабатываемый Apache Jakarta Project. Основными возможностями которого является:

- кроссплатформенность, инструмент написан на Java;
- гибкость, использования множества протоколов, не только веб-сервер, но и базы;
- управление через консоль и gui интерфейс;
- использование напрямую логов веб-сервера Apache и Nginx в качестве сценария с возможностью варьирования нагрузки по этим профилям.

На рис.2 приведено окно программного инструмента JMeter

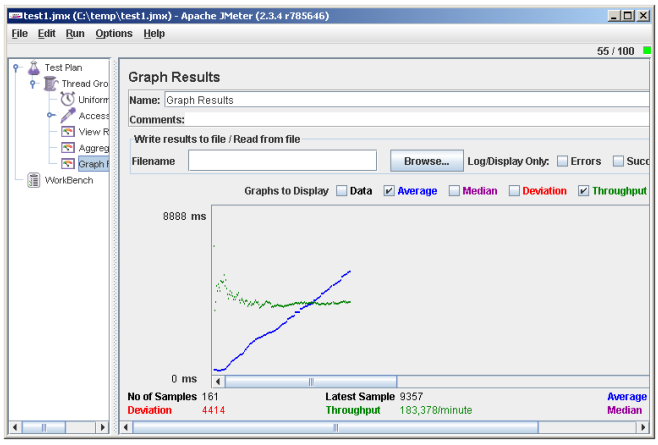


Рис.2. Окно программного инструмента JMeter

К минусам данного продукта можно отнести:

- ресурсоемкость;
- падение на длительных и тяжелых текстах по разным причинам;
- стабильность работы зависит от окружения и конфигурации сервера.

Инструмент для нагрузочного тестирования WAPT, к достоинствам данного продукта можно отнести:

- гибкость, большое количество настроек и тестов;

- эмуляция медленных каналов соединений пользователей;
- подключение модулей;
- запись сценариев тестирования прямо из браузера, как с десктопного, так и с мобильного;
- генерация различных графиков тестирования с помощью скриптов.

На рис.3 приведено окно программного инструмента WAPT

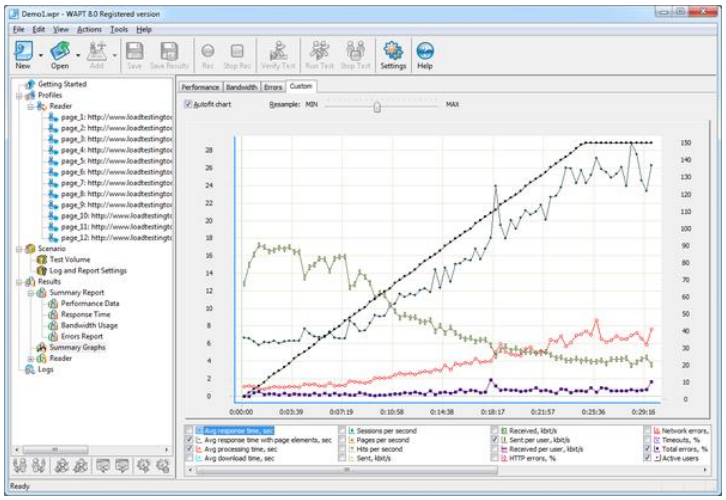


Рис.3. Окно программного инструмента WAPT

К недостаткам инструмента можно отнести: доступность только для Windows; платность.

Из приведенных выше инструментов, видны их недочеты. Ели у программного продукта высокие возможности, то он либо платный, либо ресурсоемкий.

Предлагаемые методики тестирования

Основными видами нагрузочного тестирования являются: тестирование производительности (performance testing), стрессовое тестирование (stress testing), объемное тестирование (volume testing), тестирование стабильности или надежности (stability / reliability testing).

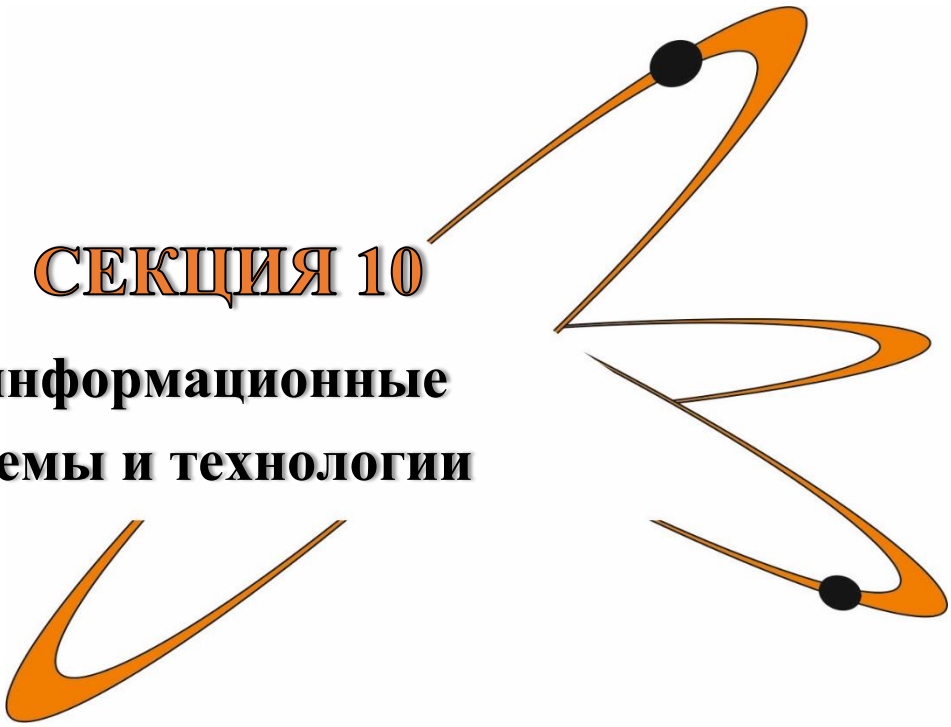
В результате применения предложенной методики нагрузочного тестирования можно получить следующие результаты:

- параметры и характеристики производительности, а также более четкое представления о работе системы в целом;
- определить предельный объем данных системы с сохранением приемлемой производительности;
- узнать предельное количество пользователей групп системы с сохранением приемлемой производительности;
- определить ресурсоёмкие операции или сценарии для дальнейшего профилирования системы.

Библиографический список

1. Савин Р. Тестирование Дот Ком, или Пособие по жестокому обра- щению с багами в интернет-стартапах. — М.: Дело, 2007. — 312 с.

2. Канер Сем и др. Тестирование программного обеспечения. Фундаментальные концепции менеджмента бизнес-приложений: Пер. с англ./Сем Канер, Джек Фолк, Енг Кек Нгуен . – К.: Издательство “ДиаСофт”,2001. – 544 с.



СЕКЦИЯ 10

Геоинформационные системы и технологии

УДК 681.3
З.Б.Аблякимова, студент 4-го курса
Научный руководитель: В.А. Строганов, старший преподаватель
Севастопольский государственный университет

ПОДСИСТЕМА ХРАНЕНИЯ И АНАЛИЗА ДАННЫХ СИСТЕМЫ ПОИСКА УТЕЧЕК ГОРОДСКИХ ВОДОПРОВОДНЫХ СЕТЕЙ

Аннотация: данная статья посвящена автоматизированной информационной системе предназначенной для обнаружения утечек подземных трубопроводов, позволяющей оперативно обнаружить факт утечки и установить место ее образования. Разработанная система не только значительно сократить время реакции аварийных служб и, как следствие значительно уменьшить экологический ущерб от разлива перекачиваемых продуктов, но и свести к минимуму время вынужденного простоя трубопровода.
Ключевые слова: утечки трубопровода, геоинформационная система, геосервис, серверная подсистема.
Annotation:
This article deals with the automated information system designed to detect leaks of underground pipelines, which allows quickly to find the leak at her place of education. The developed system not only significantly reduce the response time of emergency services and as a result significantly reduce the environmental damage from the spill pumped product, but also to minimize the period of forced inactivity of the pipeline.
Key words: pipeline leakage, geoinformation system, geoservice, server subsystem.

Обнаружение мест утечек воды из подземных трубопроводов было и остается серьезной проблемой, стоящей перед предприятиями коммунальных и аварийных служб. Не говоря уже об эксплуатационных затратах на земляные, восстановительные и другие работы, перебои в подаче воды и тепла доставляют массу неудобств потребителям. Все это обуславливает острую потребность в точном и оперативном определении мест утечек, локализации земляных работ и быстром восстановлении водопроводных и тепловых сетей. В настоящее время существует ряд методов для поиска утечек, к ним относятся гидравлический, акустический, корреляционный, тепловизионный и другие методы, однако единого, пригодного для работы в любых условиях способа еще не разработано.Наиболее перспективным представляется метод поиска утечек на основе анализа акустических сигналов, считываемых либо с поверхности земли над местом пролегания трубопровода, либо непосредственно со стенки трубопровода [1].

Целью настоящей работы является создание универсальной системы мониторинга скрытых утечек трубопроводной сети, которая представляет собой взаимодействующие друг с другом мобильное приложение и серверную подсистему. Работа серверной подсистемы будет заключаться в анализе полученных данных от мобильного устройства, определении состояния трубопровода, за счет сравнения через определенные периоды времени акустических портретов трубопровода. В случае обнаружения аномалий, будут создаваться задания для сотрудников ремонтных бригад. Кроме того данное приложение хранит информацию о местоположении точек измерений, получает и сохраняет результаты измерений, находит точные координаты повреждений трубопроводов. Разрабатываемая система позволит обнаружить аварию на ранней стадии, вследствие чего уменьшатся затраты на восстановление аварийного участка трубопровода.

При разработке программного продукта были использованы следующие технологии:

В качестве языка программирования для написания серверной части был выбран высокопроизводительный, объектно-ориентированный PHP Framework на базе готовых компонентов – Yii. Он позволяет максимально применить концепцию повторного использования кода и может существенно ускорить процесс веб-разработки. Yii использует шаблон проектирования MVC (Model-View-Controller), он является полнофункциональным фреймворком предоставляющим множество проверенных и готовых к использованию функций: построение запросов с помощью ActiveRecord для реляционных и NoSQL баз данных, поддерживает RESTful API, многоуровневое кэширование и прочие возможности.

Для написания клиентской части WEB-приложения был использован JavaScript-фреймворк с открытым исходным кодом – AngularJS. AngularJS – это производительный и

гибкий open source фреймворк для создания веб приложений, который сопровождается корпорацией Google. Фреймворк предназначен для построения single-page веб приложений и является набором JavaScript функций для организации кода на стороне клиента. В основе Angular JS лежит шаблон проектирования Model View Controller, что дает ряд преимуществ при разработке и тестировании приложения.

Кроме перечисленного, для достижения адаптивного дизайна сайта был использован CSS Framework от Twitter Bootstrap. Bootstrap позволяет сэкономить время и усилия, используя шаблоны дизайна и классы, и сконцентрироваться на других разработках. Написанные с его помощью приложения масштабируются на разные устройства и разрешения экрана без каких-либо изменений в разметке.

Яндекс.Карты — поисково-информационная картографическая служба Яндекса. API предоставляет доступ ко всему содержимому Яндекс.Карт — сотням подробных схем городов, спутниковым снимкам, а также к данным Справочника и поиску по организациям. С помощью API Яндекс.Карт возможно создать интерактивную карту, показать на ней любые данные и реализовать геосервисы любой сложности.

Разработанная система имеет удобный и интуитивно понятный пользовательский интерфейс, что облегчает ее обучение и использование. Система осуществляет аутентификацию пользователей, что предотвращает несанкционированный доступ к информационным ресурсам сервера. Кроме того, происходит валидация введенных данных, что также предотвращает ввод некорректной информации, что в дальнейшем могло бы привести к дополнительным трудностям. За счет использования Яндекс.Карт, а именно геолокации, наглядно продемонстрированы точки измерения, возможность добавления и удаления этих точек(см. рис. 1), а также возможность добраться до любой из них проложив маршрут. На сайте содержится вся информация о состоянии, координатах, исполнителях и прочей информации любой из точек(см. рис. 2, 3).

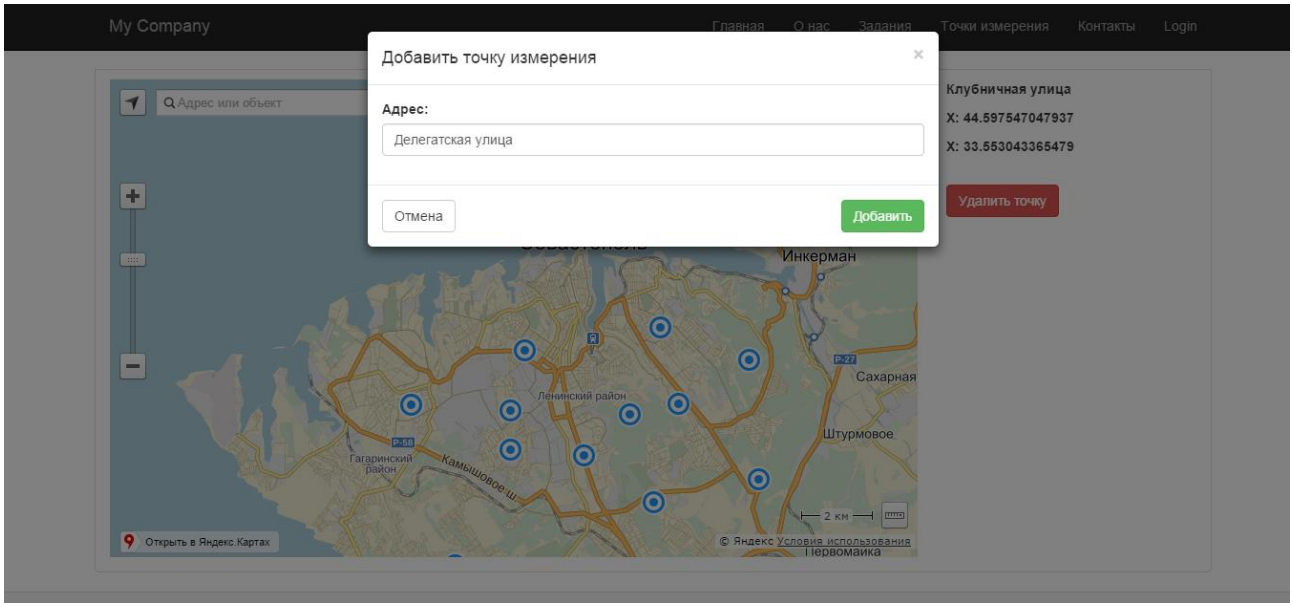


Рис 1. Окно добавления и удаления точек утечек

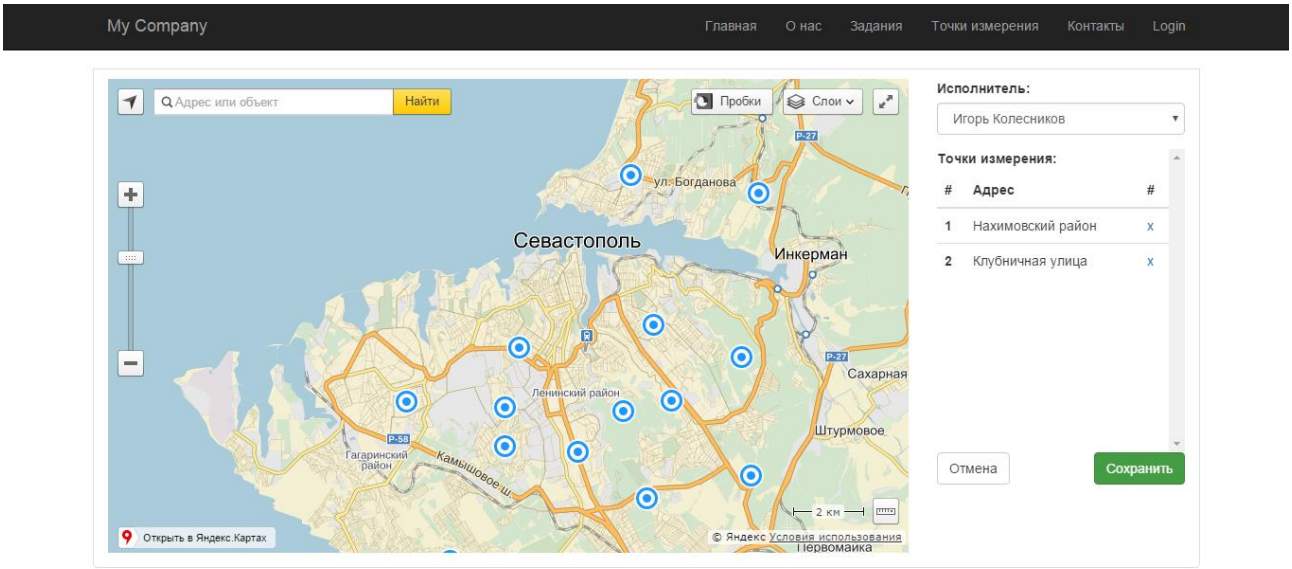


Рис 2. Окно назначения заданий ремонтным группам

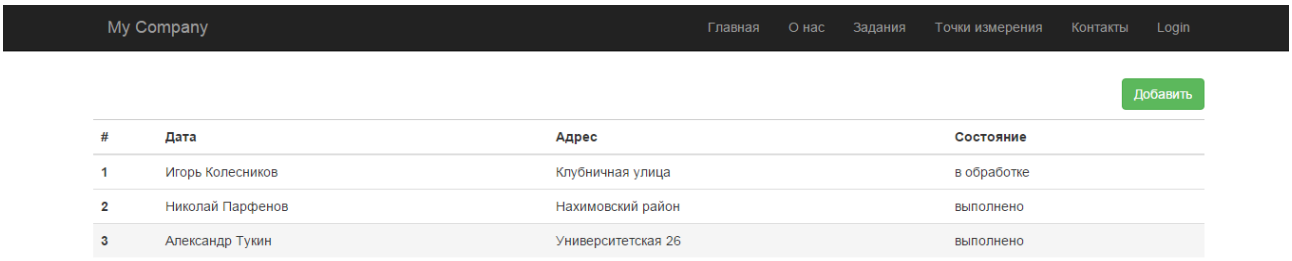


Рис 3. Окно информации обо всех заявках и их состоянии

Таким образом, разработанная система предоставляет пользователю информацию о точном местонахождении предполагаемой утечки, анализирует состояние трубопроводов, посылая необходимую информацию взаимодействующей с ней подсистемой (мобильному приложению), предоставляет набор инструментов по отображению сигнала и его спектра.

Разработанная система обладает следующими функциями:

- авторизация пользователя в системе;
- получение данных для анализа и дальнейшей обработки;
- отображение мест утечек на картах в режиме реального времени;
- добавление новых точек и соответствующей информации о них;
- отправка заданий ремонтным бригадам;
- отображение сигнала, его спектра, шумов и частотных характеристик.

Данная система способствует заметному улучшению и автоматизации процесса поиска, учета и устранения утечек в сети трубопровода. Использование представленной системы позволит обнаружить аварию на ранней стадии, а также поможет избежать значительных трат материальных средств, и обеспечит оперативную работу ремонтных групп.

Исходя из вышесказанного, можно заключить, что разработка и внедрение данного программного продукта во многом облегчит работу коммунальных и аварийных служб, а также служб ответственных за водоснабжение и водоотведение.

Библиографический список

1. Строганов В.А. Локализация утечек подземных трубопроводов на основе сопоставления акустических портретов/В.А. Строганов//«Восточно-Европейский журнал передовых технологий», 2014.- № 11(70). - С. 49 — 52.

УДК 681.3
А.В. Иванов, студент 4-го курса
Научный руководитель: И.В. Дымченко, старший преподаватель
Севастопольский государственный университет

ВЫБОР МЕТОДА КОРРЕКЦИИ КООРДИНАТ ДЛЯ РЕШЕНИЯ ЗАДАЧ НАВИГАЦИИ

Аннотация: в статье рассматриваются методы Real Time Kinematic, Precise Point Positioning для коррекции данных спутникового позиционирования с целью применения их для определения местоположения в подсистеме Web-ГИС СевГУ «Помощник студента».
Ключевые слова: система точного позиционирования, дифференциальная коррекция, кинематика реального времени, система дифференциальной коррекции.
Annotation: the Author reviews Real Time Kinematic and Precise Point Positioning methods to correct satellite positioning data in order to locate user position coordinates in "Web-GIS SevGU" subsystem called "Student Helper".
Key words: Precise Positioning system, Precise Point Positioning (PPP), Real Time Kinematic (RTK), GNSS Augmentation.

Введение. Глобальные системы спутникового позиционирования получили широкое распространение для решения разнообразных задач: сбор и обновление геоинформации, создание координатной основы карт, точное управление транспортом, прибрежная разведка месторождений и другие. Однако, в силу задержки распространения сигналов в ионосфере и нижних слоях атмосферы, расхождения между фактическим и расчетным положением спутника, расхождения шкал времени различных спутников, погрешности определения расстояния до спутника, отсутствия сигнала от потенциального доступного расстояния до спутника и др., точность позиционирования значительно снижается.

Таким образом, выбор метода коррекции и уточнения, полученных данных позиционирования является актуальным.

Цель статьи. Описание и выбор метода коррекции данных местоположения, полученных при помощи систем глобального позиционирования GPS/ГЛОНАСС, в рамках реализации проекта Web-ГИС СевГУ – интерактивной геоинформационной системы Севастопольского государственного университета.

Основная часть. Разрабатываемая «Web-ГИС СевГУ» предназначена для обеспечения геоинформационной поддержки университета, в том числе осуществления навигации по территории университета, включает в себя ряд подсистем: геопортал, хранилище данных, мобильные веб-приложения.

Подсистема «Помощник студента», разработчиком которой является автор статьи, представляет собой мобильное приложение для навигации по корпусу университета.

Для того, чтобы сделать приложение максимально простым в применении и иметь возможность использовать его в качестве полноценного навигатора, необходимо решить задачу получения данных о местоположении с точностью до нескольких сантиметров в реальном времени.

Использование GPS (англ. Global positioning system – глобальная система позиционирования), без коррекции полученных данных, дает точность от 30-х до 50-ти метров, что, не соответствует требованиям к решаемой задаче.

Для достижения точности менее десяти сантиметров спутниковая навигация должна учитывать сложности физики радиоволн и характеристики сигналов в пространстве. Точность позиционирования прямо пропорциональна точности рассчитанного расстояния между спутником и приемником и зависит от точности расчета положения спутников и погрешностей их часов.

Для вычисления неизвестных или целочисленных погрешностей, связанных с фазовыми значениями, используются два метода:

− дифференциальной коррекции с помощью опорной станции – RTK (Real Time Kinematic);

− получение новой и более точной информации о положении спутников и погрешности их часов – PPP (Precise Point Positioning) [1].

В основе дифференциальной коррекции лежит следующий принцип: измерения выполняются одновременно двумя или более приемниками, точные координаты которых заранее известны. Их называют базовыми или референцными станциями (Reference Receiver). Другой приемник является подвижным, его координаты, разумеется, заранее неизвестны (Roving Receiver) (рис.1). Для увеличения точности позиционирования по известным и полученным координатам вычисляется ошибка измерения, на основе которой находятся поправки для подвижного приемника.

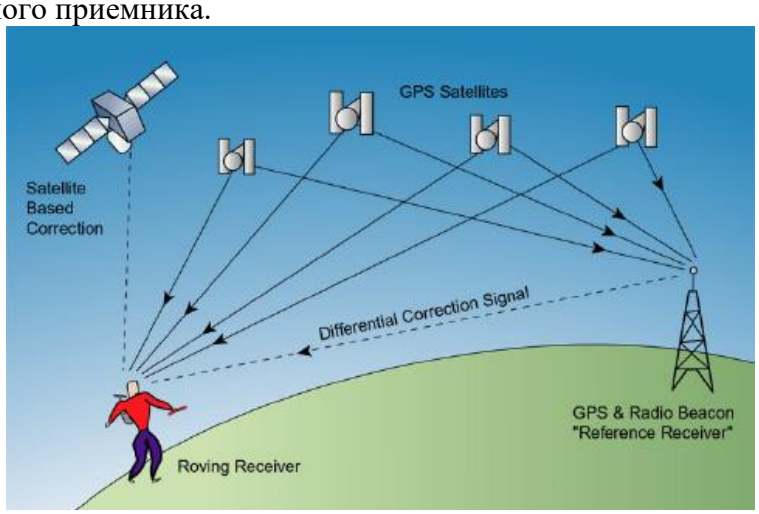


Рис 1. Принцип работы дифференциальной коррекции
(Источник http://www.directionsmag.com/images/articles/GPS_articles/realtime_diff_GPS.jpg)

Существует несколько способов дифференциальной коррекции.

Первый основан на том, что полученные псевдодальности (искаженная погрешностями дальность от объекта наблюдения до спутника) сравниваются с заранее вычисленными расстояниями и определяются их разности. Их называют дифференциальными поправками. Мобильная станция, получив дифференциальные поправки, исправляет измеренные псевдодальности, получая точные координаты.

Второй способ заключается в использовании референцной станции для вычисления разности между полученными координатами и определенными в автономном режиме, с помощью которых исправляются координаты на подвижном приемнике. При этом важно, чтобы приемник и референцная станция получали данные с одних спутников [2].

Развитием дифференциальных систем коррекции являются кинематические дифференциальные системы реального времени (RTK – Real Time Kinematic), получившие широкое распространение получили в геодезии. Принцип работы заключается в наличии от 2-х до 7-ми референцных станций, которые постоянно обмениваются данными с помощью специального программного обеспечения, вычисляющего координаты на основе собранной статистики, псевдодальностей и истинных координат.

Основная сложность заключается в необходимости большого количества станций и обеспечении подключения референцных станций и приемника к одним спутникам.

Альтернативой дифференциальной коррекции выступает метод получения новой и более точной информации о положении спутников и погрешности их часов (PPP).

Основные преимущества:

- независимость;
- обработка данных с помощью программного обеспечения;
- позволяет экономить время и ресурсы, затрачиваемые на передачу данных между референцными стациями и приемником;
- глобальное, а не местное применение.

Определение местоположения с помощью PPP базируется на обработке следующей системы уравнений без учета влияния ионосферы.

$$P_{IF} = \frac{f_1^2 \cdot P(L1) - f_2^2 \cdot P(L2)}{f_1^2 - f_2^2} = \rho - cdT + d_{trop} \quad (1)$$

$$\Phi_{IF} = \frac{f_1^2 \cdot \Phi(L1) - f_2^2 \cdot \Phi(L2)}{f_1^2 - f_2^2} = \rho - cdT + d_{trop} + \frac{cf_1N_1 - cf_2N_2}{f_1^2 - f_2^2} \quad (2)$$

где f_1 и f_2 – частоты GPS L_1 и L_2, $P(L_i)$ и $\Phi(L_i)$ – кодовые и фазовые наблюдения, ρ – действительное расстояние между спутником и приемником, c – скорость света, d_{trop} – задержка сигнала в тропосфере, N_i – фазовая неоднозначность в $\Phi(L_i)$, dT – погрешность часов.

Для компенсации основных погрешностей, возникающих при абсолютных фазовых GPS измерениях, в данном методе используются точные значения эфемерид и поправок часов спутников, информация о задержке спутникового сигнала в ионосфере и тропосфере и др., что видно из уравнений 1 и 2. [3]

Такую информацию в виде отдельных файлов формируют в международных сервисных центрах обработки данных ГНСС наблюдений (GPS и ГЛОНАСС) и предоставляют пользователям из различных стран через специализированные Интернет-ресурсы.

В [3] приводится иллюстрация точности статического (рис.2) и кинематического (рис.3) PPP. В первом случае измерения производились в течении 24 часов на станции с заранее известными координатами.

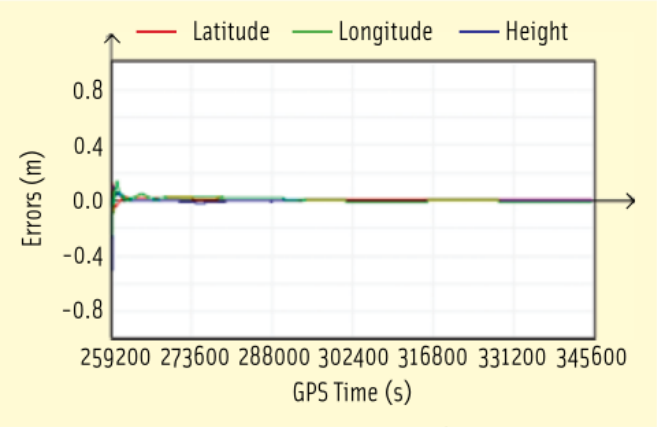


Рис 2. Ошибка статического PPP

График на рисунке 3 показывает ошибку в сравнении с RTK-решениями для кинематического PPP позиционирования самолёта.

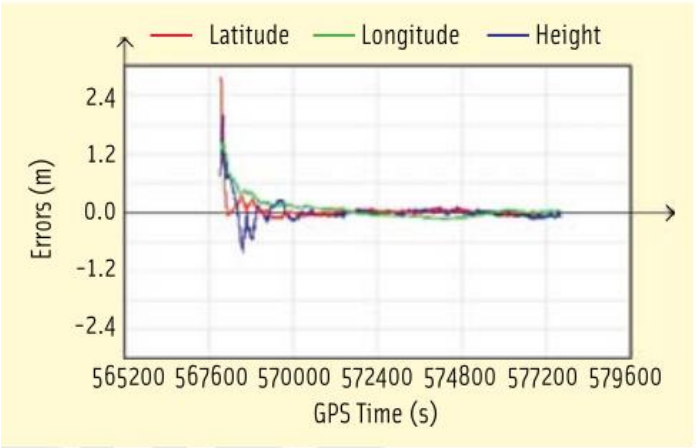


Рис 3. Ошибка кинематического PPP

Если максимально упростить описание принципов работы вышеописанных технологий, получатся следующие предложения:

− RTK: «Сообщите мне точные координаты референцной станции и получите точные координаты GPS-приемника»;

− PPP: «Сообщите мне точные данные о навигационных спутниках и получите точные координаты GPS-приемника» [3].

Оба метода обладают приблизительно одинаковой точностью, но и их совместное использование тоже возможно.

Существует так же ряд отличий:

− RTK требует дополнительных затрат на оборудование и позволяет охватывать только те территории, на которых присутствуют референцные станции.

− PPP требует полной информации о положении спутников в реальном времени, из которых не все находятся в свободном доступе.

Выводы и дальнейшее направление исследований. Для решения задачи получения точных координат, в подсистеме «Помощник студента» целесообразно применить метод Real Time Kinematic, который обеспечит точность до 1-2 сантиметров.

Удовлетворение требований к точности путем создания сети референцных станций (или хотя бы одной, в радиусе действия которой, будет находиться университет) дает возможность расширить функционал мобильного приложения и определять так же третью координату – высоту с достаточно высокой точностью в реальном времени.

Подобные решения, расширенные голосовым управлением и голосовыми подсказками, позволят использовать мобильные устройства для обеспечения навигации для людей с ограниченными возможностями.

Библиографический список

1. One-Centimeter Accuracy with PPP [Электронный ресурс] URL: http://www.insidegnss.com/node/2977 (Дата обращения: 07.03.16)
2. Серапинас Б.Б. Глобальные системы позиционирования: Учеб. изд/ Серапинас Б.Б. – М.: ИКФ «Каталог», 2002, - 106 с. с. 56-60
3. Lionel J. Garin’s “Precise Point Positioning and Its Challenges, Aided-GNSS and Signal Tracking” [Электронный ресурс] URL: http://www.insidegnss.com/auto/NovDec06GNSSSolutions.pdf (Дата обращения: 07.03.16)

УДК 656.021.8
А.М. Кричевский, студент 4-го курса
С.А. Дрозд, студент 4-го курса
Научный руководитель: В.А. Строганов, старший преподаватель
Севастопольский государственный университет

ПОДСИСТЕМА СБОРА И ОТОБРАЖЕНИЯ ДАННЫХ И ПОДСИСТЕМА ОПТИМИЗАЦИИ МАРШРУТОВ И ВЕДЕНИЯ СТАТИСТИКИ СИСТЕМЫ АНАЛИЗА И УПРАВЛЕНИЯ ДЛЯ ГОРОДСКОГО МАРШРУТНОГО ТРАНСПОРТА

Аннотация:
Рассмотрена подсистема сбора и отображения данных в реальном времени, являющаяся частью проекта, направленного на решение проблемы оптимальной организации работы общественного транспорта. Разработана система, позволяющая эффективно решать задачи визуального представления данных, получаемых от других подсистем, манипуляции данными, реализованы системные функции и средства активации взаимодействия с другими подсистемами проекта.
Рассмотрена подсистема оптимизации маршрутов и ведения статистики, являющаяся частью проекта, направленного на решение проблемы оптимальной организации работы общественного транспорта. Разработана система, позволяющая эффективно решать задачи пересчета расписаний движения и соответствующей коррекции скорости транспортных средств на основе данных о текущем их положении, получаемых от других подсистем обработки данных.
Ключевые слова:
Городской маршрутный транспорт, система мониторинга, геоинформационная система, оптимизация, многокритериальный анализ, принятие решений.
Annotation:
The paper addresses the problem of optimal city route traffic organization. Real time system for data collection and visualization was investigated. As a result the problems of real time data obtaining, data visualization, data manipulation were solved. Considered subsystem of route optimization and statistics, which is part of a project aimed to solving the problem of optimal organization of public transport. A system that allows you to effectively solve the problem of conversion timetables and appropriate speed of vehicles on the basis of the correction of the current position data received from other subsystems, data processing.
Keywords: City route transport, monitoring system, geoinformation system, optimization, multi-criteria analysis, decision-making.

Проживая в Севастополе и не имея личного транспорта, каждый обязательно столкнется с необходимостью пользования услугами городского маршрутного транспорта, а следовательно и не сможет не заметить разнообразные проблемы ему присущие. На данный момент, к наиболее значительным проблемам можно отнести:

- нехватку транспортных средств на определенных маршрутах;
- плохое техническое состояние значительной части транспортных средств;
- не оптимальное распределение количества транспортных единиц по времени;
- невозможность оперативно среагировать на любые внешние факторы (например, поломка другого транспортного средства).

Разберем все эти проблемы по отдельности и попробуем найти решение.

Нехватка транспортных средств на определенных маршрутах, вероятно, вызвана как раз тем, что перевозчики, отвечающие за них, не могут объективно оценить масштаб спроса по маршруту в виду отсутствия источника актуальной информации. Именно это позволило бы ответственным лицам точно рассчитывать необходимое количество транспортных средств. Однако, при этом возникает новая проблема: не оптимальное распределение количества транспортных единиц по времени. Как известно, существует, так называемый, «час пик», который в течение дня возникает не единожды и не всегда совпадает среди маршрутов. Здесь перевозчик вступает в неоправданный экономический риск, пуская транспортные средства по принципу «удовлетворить пик спроса», так как в определенное время, спрос будет мал для такого количества работающих единиц.

Как следствие рассмотренных двух проблем, касающихся нехватки транспортных средств на маршрутах и их временного распределения, можно сделать вывод, что необходима такая информационная система, которая могла бы предоставить реальные данные о

загруженности транспортных средств с распределением по времени. Также была бы полезна информация, касающаяся количества людей на остановках в течение времени.

Перейдем к следующей проблеме: невозможность оперативно реагировать на внешние факторы. Для понимания, предположим следующую ситуацию: транспортное средство вынуждено было сойти с маршрута по тем или иным причинам. Возникает проблема оперативного извещения участников системы, как следствие, другие водители не могут скорректировать свое движение, и возникает интервал до появления следующего маршрутного транспортного средства. Это делает сервис непредсказуемым и неудобным для конечного потребителя.

Исходя из описания вышеназванной проблемы, можно выдвинуть еще одно требование к информационной системе: необходимо обеспечить возможность уведомления всех зависящих лиц, а так же предоставить полезную информацию, способную помочь в устранении возникших неудобств.

На основе анализа описанной ранее проблемы, связанной с реакцией на внешние факторы, можно допустить, что возникновение подобных проблем можно предупреждать, и, как следствие, — уменьшить количество рискованных ситуаций. Непосредственно на это может повлиять проблема плохого технического состояния транспортных средств, что уже было ранее упомянуто в списке проблем. Тогда нужно, чтобы информационная система каким-либо образом могла предупреждать аварийную ситуацию, а значит — соответствующим образом информировать ответственное лицо.

На основе вышесказанного и перейдя к конкретной ситуации, можно выдвинуть требование о накоплении информации по схождению с маршрута, в частности по причине поломки, на основе которой можно делать вывод о критическом износе. Также полезной функцией было бы указывать, сколько какое транспортное средство проехало километров, в результате чего, на основании технического паспорта, можно предварить большинство поломок.

Подводя итог проведенного анализа, можно выделить следующие ключевые требования к системе: информационная система должна предоставлять объективную и оперативную информацию о загруженности маршрутов с учетом распределения по времени, возможность обмена информационными сообщениями между системой и клиентом, где в роли клиента может выступать оператор информационной системы или водитель транспортного средства, возможность реагировать на схождение транспортного средства с маршрута с указанием причины.

Перед началом разработки подсистемы сбора и анализа данных, в результате начального проектирования, возник вопрос выбора технологической платформы. Для данной подсистемы конечными пользователями будут являться представители перевозчика (операторы информационной системы) и пользователи маршрутных транспортных средств (пассажиры). Из чего следует, что она должна быть максимально кроссплатформенна и не должна быть труднодоступна. Также система должна уметь отображать данные в реальном времени и общаться посредством стандартизированных сообщений со связанными подсистемами. На основе заданных требований было решено разработать такую систему в виде веб-приложения.

Были использованы следующие технологии:

Язык программирования Python –интерпретируемый, компактный и удобный язык,сдинамической типизацией, предоставляющей необходимую гибкость и скорость разработки. Этот язык широко используется в разработке объектно-ориентированных веб-приложений, обладает строгой динамической типизацией и собственным сборщиком мусора.

Веб-фреймворк Flask – технология предоставляет собой микрофреймворк, также отличающейсвоей гибкостью и масштабируемостью, обладает большим и развивающимся сообществом, большой базой готовы[решений для связывания необходимых технологий призванных укорить разработку.

Язык программирования Ruby – динамический, интерпретируемый, компактный и удобный язык, разработанный японским инженером ЮкихироМацумото, изначально рассчитанный на разработку объектно-ориентированных приложений, обладает строгой динамической типизацией и собственным сборщиком мусора.

Веб-фреймворк Ruby on Rails – технология предоставляет жесткий и продуманный каркас для разработки серверных приложений любой сложности на языке ruby, обладает большим и развивающимся сообществом, большой базой решений для связывания необходимых технологий.

Веб-фреймворк Angular JS – технология для разработки динамического пользовательского интерфейса на основе языка программирования JavaScript. К ее главным преимуществам можно отнести наличие двустороннего связывания данных, что освобождает сервер от необходимости работать с шаблонами, а так же большое и развивающееся сообщество, наполненное полезной информацией.

Библиотека тестирования RSpec – удобное, обладающее «человекопонятным» синтаксисомрешение, позволяющее применить в разработке проекта принципы разработки через тестирование, тем самым документируя ее и значительно упрощая отладку. Также, такой подход позволяет обойти подробное проектирование системы, так как сам по сути и декларирует ее поведение.

Картографический сервис Яндекс.Карты — гибкая, многофункциональная разработка одной из ведущих российских компанийпозволяет реализовать все необходимые функции связанные с позиционирование и отображением соответствующих данных на карте города.

Итогом разработки становится интерактивное веб-приложение, представляющее из себя географическую информационную систему, которая предоставляет конечному пользователю в реальном времени информацию о местонахождении конкретных транспортных средств, оценивает время их прибытия и количество находящихся внутри пассажиров. Для перевозчика доступен отдельный интерфейс с возможностью просмотра собранной статистики и фильтрацией данных, позволяет осуществлять общение с водителями транспортных средств, посредством отправки и приема сообщений. Модуль статистики предоставляет в удобном виде информацию о распределении пассажиров каждого конкретного маршрута во времени, времени и причине схода с маршрута каждого транспортного средства. Прием и отправка сообщений же может происходить не только по инициативе участников диалога, но и в случае необходимости отправить автоматический отчет о происходящем.

Как следствие, разработанная информационная система реализует следующие основные функции:

- возможность регистрации пользователя в системе;
- получение от мобильных подсистем информации о местоположении транспортных средств;
- предоставление интерфейса просмотра карт маршрутов;
- отображение на картах в реальном времени положения транспортных средств на маршрутах;
- фильтр для отображаемых маршрутов на карте;
- возможность авторизации перевозчика;
- отправка сообщений водителям транспортных средств от перевозчика;
- возможность просмотра информации о конкретном транспортном средстве;
- хранение и обработка информации об оплате проезда пассажиром.
- оптимизация интервалов движения транспорта на основе нескольких параметров
- сбор и хранение статистических данных о перевозках

Такому набору функций можно поставить в соответствие концептуальную схему базы данных, изображенную на рисунке 1.

Рис. 1 Концептуальная схема базы данных

На основе полученных возможностей, перевозчик способен решить все рассмотренные ранее в статье проблемы, пассажир получит не только повышение качества обслуживания, но и систему, которая поможет ему более точно спланировать свой маршрут, оценив загруженность и ориентировочное время прибытия транспортного средства.

К возможностям развития данной информационной системы можно отнести сбор и анализ данных об аварийных ситуациях, связанных с транспортными средствами, выделение опасных участков маршрутов, определение необходимости водителю пройти дополнительный инструктаж.

УДК 681.3
А.А. Малиновский, студент 4-го курса,
А.В. Серба, студентка 4-го курса
Научный руководитель: В.А. Строганов, старший преподаватель
Севастопольский государственный университет

КРОССПЛАТФОРМЕННОЕ МОБИЛЬНОЕ РЕШЕНИЕ«VERTIGO»ДЛЯ ВОДИТЕЛЕЙ И ПАССАЖИРОВ В СИСТЕМЕ АНАЛИЗА И УПРАВЛЕНИЯ ГОРОДСКОГО МАРШРУТНОГО ТРАНСПОРТА

Аннотация: В данной статье рассмотрены проблемы загруженности и низкого качества услуг общественного транспорта города Севастополя. Оценены слабые места работающей системы общественного транспорта и представлены методы решения проблем. Описана концепция программного решения в целом и описан функционал специального кросс-платформенного приложения для водителя и для пассажира, которое нацелено на повышения эффективности работы и улучшения качества услуг, предоставляемых пассажирским автотранспортом.
Ключевые слова: общественный автотранспорт, мобильное приложение, веб-ориентированные технологии,Microsoft UWP, WinJS, NFC, трекинг транспорта.
Annotation: This article deals with the problems of congestion and poor quality of public transport services in the city of Sevastopol. Estimated operating system vulnerabilities and provides solutions to problems. It describes the concept of software solutions in general and describes the functionality of a special cross-platform applications for driver and passenger, which aims to increase efficiency and improve the quality of services provided by road passenger
Key words: public transport, mobile application, web - based technologies, Microsoft UWP, WinJS, NFC, tracking transport.

Каждое утро и вечер десятки тысяч людей пользуются услугами общественного транспорта в городе Севастополь. По данным, представленным в октябре 2015 Евгением Бабичем, начальником управления транспорта Правительства Севастополя, за восемь месяцев услугами государственного транспорта в Севастополе воспользовались 49 миллионов пассажиров: автомобильным транспортом – 9 миллионов 138 тысяч человек, электротранспортом – 34 миллиона человек, морским транспортом – пять миллионов человек.

На сегодняшний день в городе работают 68 маршрутов пассажирского автотранспорта, четыре маршрута морского транспорта и 15 троллейбусных маршрутов. Перевозят пассажиров 13 автокомпаний, эксплуатирующих порядка 1150 единиц машин, одно предприятие электротранспорта, выпускающее ежедневно на линию 95 из 145 троллейбусов, и два предприятия морского пассажирского транспорта, из которых государственное унитарное предприятие «Севморпорт» выпускает на линию ежедневно семь катеров, а компания «Севастопольские транспортные системы» – два парома [1].

При всём внушительном размера автопарков перевозчиков в часы пик на отдельных участках невозможно перевезти всех желающих [2]. Отсутствие четкого расписания хождения автотранспорта по своим маршрутам связанно с высокой загруженностью городских дорог, по которым движется общественный транспорт, и времени, проведенного транспортом на остановках, зависящего от загруженности автобуса и количества расплачивающихся. Такие нечеткие временные промежутки между остановками сложно однозначно оценить, поэтому сейчас водители общественного автотранспорта используют две меры времени в своих графиках маршрута – это общее время от начала до конца маршрута и время после начала движения впереди стоящего в очереди автомобиля коллег. Это довольно простое решение и на участках с большой интенсивностью движения, где не водителю несложно соблюдать скоростной режим для маршрута. Но такая схема не учитывает затяжных маршрутов, на преодоление которых от начала до конца требуется более часа.

В условиях пробок в часы пик и перегруженности автобусы малой вместимости не способны обеспечить спрос уже на середине маршрута, следовательно, ожидание подходящего транспорта для пассажира может затянуться на неопределенное время. Также можно на ряде маршрутов в определенных участках можно наблюдать обратный эффект, автобусы почти пустые проходят половину своего маршрута. Отсутствие актуальных и точных статистических данных по общественному транспорту не позволяют точно оценить и оптимизировать графики маршрутов и оптимальное количество единиц транспорта на них.

К перечисленным недостаткам устоявшейся автотранспортной системы также можно отнести способ оплаты за проезд. Использование в обиходе металлических монет существенного номинала всегда доставляло неудобства как пассажирам, так и водителю во время расчета. Местная особенность расплачиваться при выходе создает задержку потока пассажиров на остановках. Также существует ряд проблем, связанных с теневым оборотом средств и отсутствием системы принудительного обилечивания.

Еще существует множество субъективных проблем, которые, как и выше перечисленные, можно решить с помощью нового подхода к управлению автотранспортом, анализа статистических данных, оптимизации маршрутов и графиков следования, получения актуальной информации о транспорте пассажирами на остановках и обратная связь с ним.

Система анализа и управления городского маршрутного транспорта, основанная на кроссплатформенном решении «Vertigo-Aurigae» для водителя и «Vertigo-Peregrinus» для пассажира, направлена на решение выделенных раннее проблем.

К прикладному программному решению «Vertigo-Aurigae» можно применить следующие требования:

- Обеспечить дешевое аппаратное решение, реализующее возможность наблюдения в реальном времени перемещения транспорта с использованием систем геопозиционирования;

-Осуществлять стабильную передачу данных на сервер используя мобильный интернет, покрывающий существующие маршруты;

- Ограничить доступ к приложению неавторизированным лицам; фиксировать время начала и окончания работы водителя на маршруте;

- Осуществлять обмен данным в рамках системы бесконтактного билетирования с использованием технологии NFC[3];

- Обеспечить сбор статических данных о пассажирном потоке; пересылать сообщения о внештатных ситуациях (ДТП, поломках и т.п) ответственным лицам;

- Взаимодействовать с корректирующими данными от диспетчерской службы в рамках системы; снабжать водителя актуальной информацией о его личном графике и сообщать о его изменениях;

- Отображать рекомендуемые скоростные режимы и интервал движения;

К «Vertigo-Peregrinus», как приложению для пассажиров, можно применить следующие требования:

- Трекинг доступного общественного транспорта в реальном времени по запрашиваемому маршруту;

- Реализация трансфертных финансовых операций и пополнения NFC-метки с возможностью пополнения различными интернет кошельками.

Для реализации сформированных требований подходит приложение на основе платформы UWP, которая является частью новой операционной системы от Microsoft Windows 10. UWP - это расширение для исполнительной оболочки Windows. Универсальное приложение, созданные с помощью UWP-приложения, не требует декларирования и адаптировано для определенной операционной системы. Такое приложение может быть нацелено как на одно устройство, так и на семейство устройств, таких как ПК, смартфоны, планшеты, микрокомпьютеры для интернета вещей и игровые приставки XboxOne, используя UniversalWindowsPlatformBridge. Это расширение позволяет автоматически использовать возможности, которые доступны для конкретного устройства во время выполнения. Подобное приложение легко поддерживать и адаптировать под целевое устройство без существенных изменений кода. Использование веб-ориентированных платформ WinJS-Angular.js позволяет относительно быстро и адаптивно, под все возможные размеров экранов, реализовать интерфейс и логику приложения. Использование в написании языков C# и javascript с препроцессором TypeScript позволит сделать код надежным и уберечь разработчика от множества ошибок на начальных этапах написания ПО и в дальнейшем позволят легко поддерживать и развивать программный продукт. Приложение «Vertigo» так же будет

возможно развертывать на облачном хранилище и использовать его через браузер. Адаптирование под разные устройства затронет всего около 10% программного кода.

В результате приложение «Vertigo» для водителя повысит качества услуг водителя общественного транспорта без кондуктора; освободит водителя от рутины с мелочью и позволит проводить быструю оплату; позволит осуществлять контроль общественного транспорта; контролировать маршруты; разгружать общественный транспорт в час пик за счет оптимизации управления транспортом на маршрутах; реализовывать гибкий контроль маршрутного такси; позволить определять сложные «красные» участки маршрутов, что в будущем позволит решить проблемы перегрузок городских дорог Севастополя.

Библиографический список

1. SevStar. Схема городского транспорта. СевСтар. [Электронный ресурс] 13 Январь 2016 г. [Цитировано: 9 Март 2016 г.] http://people.sevstar.net/novosti/225-sevstar-darit-sevastopoltsam-novuyu-transportnuyu-skhemu.

2. ООО "ИНФОРМ-МЕДИА". Утренняя проблема транспорта в Севастополе. INформер. [Электронный ресурс] 2 Декабрь 2015 г. [Цитировано: 9 Март 2016 г.] http://ruinformer.com/page/utrennjaja-problema-transporta-v-sevastopole.

3. Е.Зобнин. Используем NFC для автоматизации. ХАКЕР. [Электронный ресурс] 30 Январь 2015 г. [Цитировано: 9 Март 2016 г.] https://xakep.ru/2015/01/30/nfc-for-automation/.

УДК 681.3
А.А.Сукачев, студент 4-го курса
Научный руководитель: В.А. Строганов, старший преподаватель
Севастопольский государственный университет

ПОДСИСТЕМА СБОРА И ОТОБРАЖЕНИЯ ДАННЫХ СИСТЕМЫ ПОИСКА УТЕЧЕК ПОДЗЕМНЫХ ТРУБОПРОВОДОВ

Аннотация:
Статья посвящена подсистеме сбора и отображения данных в режиме реального времени, являющейся частью системы, решающей проблемы поиска утечек подземных трубопроводов. Разработанная система существенно облегчает сложный и дорогостоящий процесс определения точного места утечек, и автоматизирует работу ремонтных групп.
Ключевые слова: Утечки трубопровода, геоинформационная система, мобильное приложение.
Annotation:
The article is devoted to a subsystem of gathering and displaying data in real time, which is a part of a system that solves the problem of finding leaks of underground pipelines. Developed system simplifies complex and costly process of determining the exact location of leaks, and automates maintenance groups.
Key words:pipeline leakage, geoinformation system, mobile application.

Каждый сталкивался с тем, что из-за протечек в трубопроводе продолжительное время в домах нет воды и тепла. Это связано, в первую очередь, с тем, что на точное определение и восстановление неполадок в трубопроводах уходит неоправданно большое количество времени. Это обуславливает острую потребность в точном и оперативном определении мест утечек, что позволило бы локализовать землянные работы и быстро восстанавливать водопроводные сети.

Для эффективной организации работы по устранению утечек необходимо применять автоматизированные программные решения. Разрабатываемая система должна решать задачи создания и учета заявок на поиск утечек, хранения измеренных сигналов (акустических портретов трубопровода), хранения информации о местоположении точек измерений, получения и сохранения результатов измерений, модификации статусов заявок. Целью нашей работы было создание именно такой мобильной подсистемы, предназначенные для использования сотрудниками бригад, осуществляющих измерения параметров трубопровода.

Было принято решение в качестве главной операционной системы для мобильного приложения использовать Android версии 4.0.3 – 4.0.4 Ice Cream Sandwich, чтобы обеспечить максимальную работоспособность на всех устройствах (97% всех устройств по данным статистики на 2016 год). Выбор ОС Android так же объясняется тем, что большая часть населения среднего и выше возраста, проживающих на территории РФ и СНГ, использует именно эту операционную систему на своих мобильных устройствах.

При разработке программного продукта были использованы следующие технологии:

- в качестве языка программирования был выбран Java. Этот язык разработала компания Sun Microsystems (которую приобрела компания Oracle). Особеностью языка является то, что приложение, написанное на Java транслируются в специальный байт-код, из чего следует, что оно может сохранять работоспособность на любой виртуальной Java-машине, вне зависимости от архитектуры компьютера. Java часто используется для разработики мобильных приложений, использующих операционную систему Android. Все, что необходимо для этого – среда Android Studio (в которой и разрабатывалась система) и плагин Android Development Tool.

- библиотеки: GSON, RETROFIT. GSON – это Java-библиотека, которая используется для сериализации и десериализации Java-объектов из JSON и обратно. Такая задача возникла, так как необходимо общаться с серверной подсистемой посредством API. RETROFIT – это простое и красивое решения для взаимодействия приложения с API. Эта библиотека помогает сделать код чище и проще. Можно использовать как синхронные, так и асинхронные запросы, добавлять тело запроса, манипулировать URL-адресами или заголовками, добавлять параметры запроса. Обе эти библиотеки отлично сочетаются и помогают в разы ускорить разработку и избавить разработчика от рутинной работы.

- Google maps – картографический сервис компании Google, который предоставляет собой спутниковые интерактивные карты планеты Земля. Карты Google является лидером среди таких сервисов, так как обладает большим количеством разнообразных дополнительных сервисов и инструментов.

Тестирование — это один из важных этапов разработки программного продукта. Для разработки тестов была выбрана библиотека Robolectric. Это библиотека, которая запускает тесты для Android-приложения на локальной JVM. Это существенно ускоряет процесс написания и выполнения тестов, так как нет необходимости ждать, пока загрузится и установится apk-файл, пока запустится приложение на устройстве. Этот инструмент отлично подходит для unit-тестирования. Для наглядного отображения покрытия кода тестами существует JaCoCo. Результаты отображаются в красивой форме в виде html-отчета.

Разработанная система имеет удобный и интуитивно понятный интерфейс, что облегчает обучение и использование данной системы у ремонтных групп. Система предусматривает аутентификацию пользователей, что предотвращает несанкционированный доступ к информационным ресурсам сервера. В системе использован механизм геолокации: ремонтные группы могут легко добраться до места предполагаемой утечки в быстрые сроки, имея отмеченные точки на карте. Не используя дорогостоящее оборудование, имеется возможность достаточно точно и относительно легко записать необходимый сигнал (акустический протрет трубопровода) и отправить в другую подсистему для дальнейшего анализа.

В конечном итоге получено мобильное приложение, которое предоставляет пользователю информацию о точном местонахождении предполагаемой утечки в трубопроводе, набор инструментов по отображению сигнала и его спектра, а также возможность записи акустического сигнала и отправки его на сервер для последующего анализа.

Интерефейс пользователя мобильного приложения показан на рисунке 1.

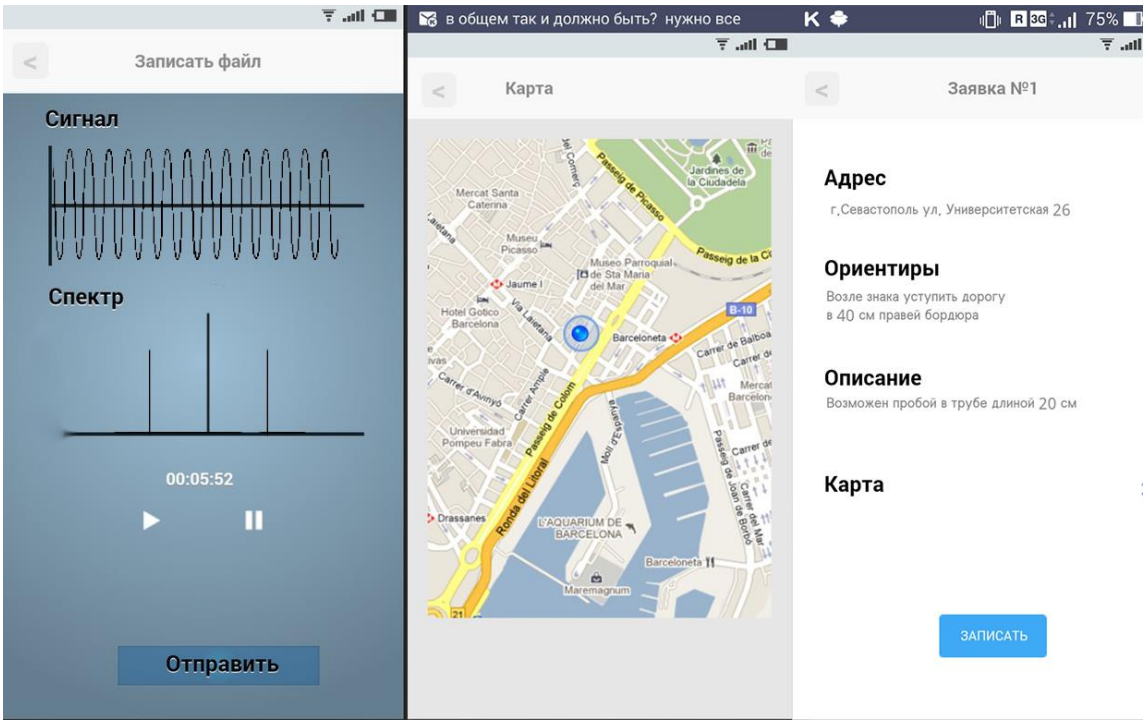


Рис 1. Основные окна приложения

Разработанная система обладает следующими функциями:

- авторизация пользователя в системе;
- получение списка текущих заявок;

- отображение мест предполагаемых утечек на картах в режиме реального времени;
- запись файлов и отправка на сервер для дальнейшей обработки;
- отображение сигнала и спектра сигнала.

Данная система способствует заметному улучшению и автоматизации процесса поиска и устранения утечек в сети трубопровода, контролированию работы ремонтных групп.

Обобщив вышесказанное, можно сказать, что данная система обладает рядом особенностей:

- высокое быстродействие;
- вывод всех результатов работы в понятной форме на экран устройства;
- интуитивно понятный и удобный интерфейс;
- работа в режиме реального времени (мобильный интернет / WiFi);
- высокая точность.

Сборник статей студенческой научно-технической конференции
Издано в авторской редакции

Ответственность за содержание статей несут авторы

Ответственный за выпуск: О.В.Ченгарь

Для создания электронного издания использовано:
Microsoft Word 2013, Adobe Acrobat Pro
Подписано к использованию 4.04.2016 г.
Уч.-изд. л. 22,36. Объем данных 8,9 Мб

ФГАОУ ВО «СевГУ», кафедра ИТиКС
299053, Севастополь, ул. Университетская, 33.
Тел.: (8692) 43-50-08
E-mail: itiks@sevsu.ru