

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 – 7 апреля 2017 г.

УДК 004

Научный редактор:

канд. техн. наук, доцент кафедры информационных технологий и компьютерных систем (СевГУ) **Е.Н. Мащенко**

Редакционная коллегия:

профессор (СевГУ), д-р техн. наук **А.В. Скатков**,
профессор (СевГУ), д-р техн. наук **Ю.К. Апраксин**,
профессор (СевГУ), д-р техн. наук **Ю.Е. Обжерин**,
канд. техн. наук, доцент (СевГУ) **А.А. Брюховецкий**,
канд. техн. наук, доцент (КФУ) **Ю.В. Сосновский**,
канд. техн. наук, доцент (СевГУ) **Д.Ю. Воронин**,
канд. техн. наук, доцент (СевГУ) **В.И. Шевченко**,
канд. техн. наук, доцент (СевГУ) **Е.Н. Мащенко**,
канд. техн. наук, доцент (СевГУ) **О.В. Ченгарь**

Ответственный за выпуск:

канд. техн. наук, доцент кафедры информационных технологий и компьютерных систем (СевГУ) **О.В. Ченгарь**

Мир компьютерных технологий: Сборник статей студенческой научно-технической конференции, г.Севастополь, 04 – 07 апреля 2017 г/ М-во образования и науки РФ, Севастопольский государственный университет; науч. ред. Е.Н. Мащенко – г. Севастополь: СевГУ, 2017. – 203 с. Режим доступа: <http://lib.sevsu.ru:8080/xmlui/handle/123456789/7985> — Загл. с титул. экрана.

В сборник входят статьи по материалам докладов студентов и магистрантов–участников студенческой научно-технической конференции (4-7 апреля 2017г.), 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

© Коллектив авторов, 2017

© СевГУ, 2017

ОГЛАВЛЕНИЕ

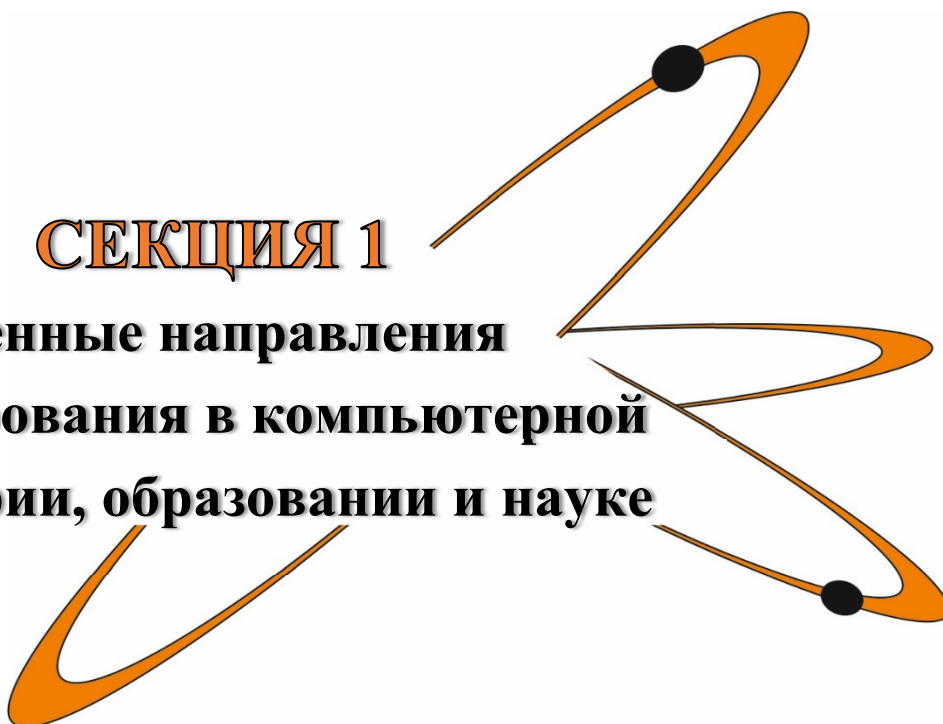
СЕКЦИЯ 1. СОВРЕМЕННЫЕ НАПРАВЛЕНИЯ МОДЕЛИРОВАНИЯ В КОМПЬЮТЕРНОЙ ИНЖЕНЕРИИ, ОБРАЗОВАНИИ И НАУКЕ	6
<i>Белая К.В., Шевченко В.И.</i> ДИВЕРСИФИКАЦИЯ МОДЕЛЕЙ ПОВЕДЕНИЯ СОТРУДНИКОВ УДАЛЕННОГО ОФИСА	7
<i>Гылка А.В., Скаковская А.Н.</i> МОДЕЛИРОВАНИЕ ТРЕХМЕРНЫХ ОБЪЕКТОВ В AUTODESK 3DS MAX.....	17
<i>Полетаева В.Д., Сысоев С.С., Воронин Д.Ю.</i> ИМИТАЦИОННЫЙ СТЕНД ПОДДЕРЖКИ ПРИНЯТИЯ РЕШЕНИЙ РАСПРЕДЕЛЕНИЯ ЗАДАЧ В РАСПРЕДЕЛЕННЫХ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМАХ	19
<i>Рудоченко С.А., Скаковская А.Н.</i> РАЗРАБОТКА ВИЗУАЛЬНЫХ ЭФФЕКТОВ СРЕДСТВАМИ КОМПЬЮТЕРНОЙ ГРАФИКИ	22
<i>Рыдзывыло К.Г., Шевченко В.И.</i> АЛГОРИТМЫ ЭФФЕКТИВНОГО РАЗМЕЩЕНИЯ ФАЙЛОВ В РАСПРЕДЕЛЕННЫХ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМАХ.....	24
<i>Сафонов М.Д., Привалов М.В.</i> АЛГОРИТМЫ И ИНСТРУМЕНТАЛЬНЫЕ СРЕДСТВА ПОСТРОЕНИЯ И ВИЗУАЛИЗАЦИИ 3D-ПОВЕРХНОСТЕЙ ЧЕЛЮСТНО-ЛИЦЕВОЙ ОБЛАСТИ ПО КОМПЬЮТЕРНЫМ ТОМОГРАММАМ.....	30
<i>Чалый А.И., Заморёнов М.В.</i> СИСТЕМА АВТОМАТИЗАЦИИ ПОСТРОЕНИЯ МАТЕМАТИЧЕСКИХ МОДЕЛЕЙ.....	35
 СЕКЦИЯ 2. РОБОТОТЕХНИКА. ТЕХНОЛОГИИ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА	39
<i>Архипова А.А., Безуглая А.Е.</i> ПРОГРАММНО-АППАРАТНЫЙ КОМПЛЕКС СИСТЕМЫ ПОДДЕРЖАНИЯ ЖИЗНЕДЕЯТЕЛЬНОСТИ РАСТЕНИЙ КАК СРЕДСТВО ОБУЧЕНИЯ РОБОТОТЕХНИКЕ	40
<i>Казанцев М.А., Привалов М.В.</i> ПАРАЛЛЕЛЬНАЯ СЕГМЕНТАЦИЯ ТРЕХМЕРНЫХ ИЗОБРАЖЕНИЙ НА ОСНОВЕ АЛГОРИТМА РАЗБИЕНИЯ ГРАФА С ПРИМЕНЕНИЕМ РОЕВОГО ИНТЕЛЛЕКТА.....	43
 СЕКЦИЯ 3. ИНТЕЛЛЕКТУАЛЬНЫЙ АНАЛИЗ ДАННЫХ. ТЕХНОЛОГИИ BIG DATA. СИСТЕМЫ ПОДДЕРЖКИ ПРИНЯТИЯ РЕШЕНИЙ	48
<i>Вилинский А.А., Козлова Е.В.</i> ПРОГРАММНАЯ СИСТЕМА ПОДДЕРЖКИ ПРИНЯТИЯ РЕШЕНИЙ О ВЫБОРЕ БРЕНД-ПРОЕКТА	49
<i>Измайлов Р.Ю., Мащенко Е.Н.</i> КОНЦЕПЦИЯ ОТКРЫТЫХ ДАННЫХ.....	52
<i>Криворучко Ю.Э., Чернега В.С.</i> МОДЕЛИРОВАНИЕ ПРОБЛЕМНО-ОРИЕНТИРОВАННЫХ ТЕКСТОВЫХ СООБЩЕНИЙ.....	55
<i>Смирнов И.В., Крячко А.В., Шевченко В.И.</i> КОНЦЕПТУАЛЬНАЯ МОДЕЛЬ ОРГАНИЗАЦИИ ПРОЦЕССОВ ОБРАБОТКИ БОЛЬШИХ ДАННЫХ В ПРОЕКТЕ C2S НА ОСНОВЕ ТЕХНОЛОГИИ SPRING XD	58
<i>Струг С.С., Васяева Т.А.</i> ЭВОЛЮЦИОННЫЙ ПОДХОД К ПОСТРОЕНИЮ ПРОДУКЦИОННОЙ МОДЕЛИ ДЛЯ ЭКСПЕРТНЫХ СИСТЕМ	63

<i>Черненко О.С., Строганов В.А.</i> ПРИМЕНЕНИЕ TF-IDF АЛГОРИТМА В РЕКОМЕНДАТЕЛЬНЫХ СИСТЕМАХ ГОСУДАРСТВЕННЫХ ЗАКУПОК.....	66
<i>Шулянский Д.В., Хмелевой С.В.</i> ОПРЕДЕЛЕНИЕ СХОЖЕСТИ СЛОВ ПРИ ПОМОЩИ ОРФОГРАФИЧЕСКОГО И ФОНЕТИЧЕСКОГО АНАЛИЗА В СИСТЕМАХ АВТОМАТИЧЕСКОГО ТЕСТИРОВАНИЯ ЗНАНИЙ.....	68
СЕКЦИЯ 4. ОБЛАЧНЫЕ И ГРИД-ТЕХНОЛОГИИ	74
<i>Лисов М.С., Брюховецкий А.А.</i> ИССЛЕДОВАНИЕ ДИНАМИКИ ХАРАКТЕРИСТИК ТРАФИКА В ОБЛАЧНОЙ ВЫЧИСЛИТЕЛЬНОЙ СРЕДЕ	75
<i>Макаров И.В., Ченгарь О.В.</i> АНАЛИЗ МЕТОДОВ РАСПРЕДЕЛЕНИЯ РЕСУРСОВ В GRID СИСТЕМЕ	77
<i>Пасик О.А., Мащенко Е.Н.</i> АНАЛИЗ СТРАТЕГИЙ ВЫБОРА ПРОВАЙДЕРА ОБЛАЧНЫХ IAAS УСЛУГ.....	82
СЕКЦИЯ 5. ПРОЕКТИРОВАНИЕ АППАРАТНЫХ СРЕДСТВ ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ	90
<i>Бойченко В.А., Моисеев Д.В.</i> ОЦЕНКА ТОЧНОСТИ И БЫСТРОДЕЙСТВИЯ ПРИ ВЕРОЯТНОСТНОЙ ФОРМЕ ПРЕДСТАВЛЕНИЯ ИНФОРМАЦИИ	91
<i>Бойченко В.А., Моисеев Д.В.</i> ПРИМЕНЕНИЕ СПЕЦИАЛИЗИРОВАННЫХ ВЕРОЯТНОСТНЫХ ПРОЦЕССОРОВ В ИНФОРМАЦИОННО-ИЗМЕРИТЕЛЬНЫХ СИСТЕМАХ	95
СЕКЦИЯ 6. КОМПЬЮТЕРНЫЕ СЕТИ	98
<i>Бернацкий А.Д., Доронина Ю.В., Дымченко И.В.</i> СИСТЕМА ПОДДЕРЖКИ ПРИНЯТИЯ РЕШЕНИЙ ПО ОПТИМИЗАЦИИ ПРОГРАММНО-АППАРАТНОЙ СРЕДЫ	99
<i>Мозговая В.В., Мащенко Е.Н.</i> АНАЛИЗ ПРОБЛЕМ ПРАКТИЧЕСКОЙ РЕАЛИЗАЦИИ БЕСПРОВОДНЫХ СЕНСОРНЫХ СЕТЕЙ	103
<i>Сивков С.А., Апраксин Ю.К.</i> ИССЛЕДОВАНИЕ НЕСОСТОЯТЕЛЬНОСТИ СЕТЕВЫХ ПРОТОКОЛОВ НА ОСНОВЕ КОНЕЧНО-АВТОМАТНОЙ МОДЕЛИ	108
<i>Сукачев А.А., Кудрявченко И.В.</i> АНАЛИЗ САМООРГАНИЗУЮЩИХСЯ MESH СЕТЕЙ.....	114
СЕКЦИЯ 7. WEB – РАЗРАБОТКА	116
<i>Анисимова Д.В., Дергачев Н.А., Чернега В.С., Дрозин А.Ю.</i> МОБИЛЬНОЕ ПРИЛОЖЕНИЕ ДЛЯ ПОСЛЕОПЕРАЦИОННЫХ ПАЦИЕНТОВ	117
<i>Батраченко А.И., Строганов В.А.</i> WEB-ОРИЕНТИРОВАННАЯ СИСТЕМА ПОСТРОЕНИЯ И АНАЛИЗА РАСПИСАНИЯ ЗАНЯТИЙ.....	119
<i>Спорыш Д.Н., Строганов В.А.</i> ИНФОРМАЦИОННАЯ СИСТЕМА ДЛЯ МЕДИЦИНСКОГО ПЕРСОНАЛА И ПАЦИЕНТОВ.....	121
<i>Сысоев С.С., Овчинников А.Л.</i> СИСТЕМА КОНТРОЛЯ УСПЕВАЕМОСТИ И ПОСЕЩАЕМОСТИ СТУДЕНТОВ	124
<i>Шушпан П.Е., Строганов В.А.</i> WEB-ОРИЕНТИРОВАННАЯ СИСТЕМА ЭЛЕКТРОННОГО ЖУРНАЛА ПРЕПОДАВАТЕЛЯ	127

СЕКЦИЯ 8. ИНФОРМАЦИОННАЯ БЕЗОПАСНОСТЬ	129
<i>Вихров Р.Г., Фисун С.Н.</i> ИССЛЕДОВАНИЕ БЫСТРОДЕЙСТВИЯ СИММЕТРИЧНЫХ АЛГОРИТМОВ БЛОЧНОГО ШИФРОВАНИЯ	130
<i>Рязанов А.С., Мащенко Е.Н.</i> ИССЛЕДОВАНИЕ ХАРАКТЕРИСТИК КАЧЕСТВА СИСТЕМЫ ОБНАРУЖЕНИЯ ВТОРЖЕНИЙ В ОБЛАЧНОЙ ИНФРАСТРУКТУРЕ....	135
 СЕКЦИЯ 9. СИСТЕМНОЕ И ПРИКЛАДНОЕ ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ	141
<i>Безпалый И.А., Абрамов Т.А.</i> КОНЦЕПТУАЛЬНАЯ МОДЕЛЬ ИГРЫ «ОСКАЛОР»..	142
<i>Воронецкий Ю.О., Журба С.С., Абрамов Т.А.</i> РАЗРАБОТКА АВТОМАТИЗИРОВАННОЙ СИСТЕМЫ «УМНЫЙ ДОМ».....	152
<i>Голованев В.В., Шлемко А., Фисун С.Н.</i> РАЗРАБОТКА ЛЕКСИЧЕСКОГО И СИНТАКСИЧЕСКОГО АНАЛИЗАТОРОВ ПРИ ПРОЕКТИРОВАНИИ КОМПИЛЯТОРА	155
<i>Гримута А.В., Шевченко В.И.</i> СИСТЕМА АВТОМАТИЗИРОВАННОГО ДОКУМЕНТООБОРОТА КАФЕДРЫ НА ПЛАТФОРМЕ «1С: ПРЕДПРИЯТИЕ».....	159
<i>Дзюба С.Н., Фисун С.Н.</i> ПРОГРАММНАЯ СИСТЕМА ДЛЯ АВТОМАТИЗАЦИИ РАЗРАБОТКИ СИНТАКСИЧЕСКИХ АНАЛИЗАТОРОВ	164
<i>Какаев А.В., Шевченко В.И.</i> АВТОМАТИЗИРОВАННОЕ ТЕСТИРОВАНИЕ ПО ОБЗОР ИНСТРУМЕНТОВ НЕПРЕРЫВНОЙ ИНТЕГРАЦИИ	168
<i>Лысянский М.В., Строганов В.А.</i> МОБИЛЬНОЕ ПРИЛОЖЕНИЕ ДЛЯ МЕДИЦИНСКОГО ПЕРСОНАЛА БОЛЬНИЦЫ	172
<i>Пашкова Ю.И., Коваленко А.С., Лебедева М.А.</i> АВТОМАТИЗИРОВАННАЯ СИСТЕМА УЧЕТА И УПРАВЛЕНИЯ ВОДОМЕРНЫМИ ПОДКЛЮЧЕНИЯМИ ГУПС «ВОДОКАНАЛ»	175
<i>Поддубный А.Ю., Шумейко И.П.</i> РАЗРАБОТКА ТЕХНОЛОГИИ ДЛЯ ФОРМИРОВАНИЯ НАУЧНОЙ ОТЧЕТНОСТИ ИЗ РАЗРОЗНЕННЫХ ИСТОЧНИКОВ СБОРА ДАННЫХ	179
<i>Райков С.С., Пармузин А.О., Чернега В.С.</i> ОРГАНИЗАЦИЯ БЕСПРОВОДНОГО КАНАЛА СВЯЗИ ДЛЯ УПРАВЛЕНИЯ МЕДИЦИНСКИМ ОБОРУДОВАНИЕМ.....	184
<i>Салидинов А.И., Шаталова Ю.Г.</i> РАЗРАБОТКА СИСТЕМЫ РЕПЛИКАЦИИ ДЛЯ РАСПРЕДЕЛЕННОЙ БАЗЫ ДАННЫХ АВТОМАГАЗИНА	187
<i>Флора Л.В., Заморёнов М.В.</i> ПРОГРАММНО-АППАРАТНЫЙ КОМПЛЕКС СИМУЛЯТОРА СПОРТИВНОЙ СТРЕЛЬБЫ	189
 СЕКЦИЯ 10. ГЕОИНФОРМАЦИОННЫЕ СИСТЕМЫ И ТЕХНОЛОГИИ	192
<i>Гальшина Е.А., Дымченко И.В.</i> ПРИМЕНЕНИЕ ГЕОИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ ПРИ РЕШЕНИИ ПРОБЛЕМ ОРГАНИЗАЦИИ СБОРА, ХРАНЕНИЯ И ПРЕДСТАВЛЕНИЯ ДАННЫХ АРХЕОЛОГИЧЕСКИХ ИССЛЕДОВАНИЙ	193
<i>Доронина Е.Б., Доронина Ю.В.</i> ОРГАНИЗАЦИОННОЕ УПРАВЛЕНИЕ ПРИ ПРОЕКТИРОВАНИИ ГЕОИНФОРМАЦИОННЫХ СИСТЕМ НА ОСНОВЕ SADT И RUP-ТЕХНОЛОГИЙ	199

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***ДИВЕРСИФИКАЦИЯ МОДЕЛЕЙ ПОВЕДЕНИЯ СОТРУДНИКОВ УДАЛЕННОГО ОФИСА**

Аннотация: В работе представлено несколько подходов к описанию и реализации имитационной модели процессов взаимодействия сотрудников удаленного офиса. Для описания концептуальной модели использована методология структурного анализа и проектирования SADT. Приведены две имитационные модели процессов взаимодействия менеджера и сотрудников удаленного офиса при выполнении функциональных задач, использующих агентное и дискретно-событийное моделирование.

Ключевые слова: методология структурного анализа и проектирования, диаграмма IDEF0, имитационная модель, агентное моделирование, дискретно-событийное моделирование.

Annotation: The paper presents several approaches to the description and implementation of the simulation model of interaction processes for employees of a remote office. To describe the conceptual model, the methodology of structural analysis and design of SADT is used. Two simulation models of interaction processes between the manager and employees of the remote office are presented when performing functional tasks that use agent and discrete-event modeling.

Key words: methodology of structural analysis and design, diagram IDEF0, simulation model, agent modeling, discrete-event modeling.

Введение

На сегодняшний день, в сфере корпоративного менеджмента, особенно крупных IT-компаний, большое внимание уделяется такому, сравнительно новому явлению, как распределённые офисы [1]. Дистанционное управление персоналом представляет собой управление людьми на расстоянии с помощью современных информационных технологий. Функции, предоставляемые корпоративной информационной системой в базовом варианте, дают возможность вести оперативное управление средствами видеоконференций и общее управление на основе информации, передаваемой в базу данных центрального офиса из регионов. Дистанционный менеджмент сокращает расходы на переезды до 50% и дает выигрыш в производительности. При этом сотрудники руководящего звена сталкиваются с рядом проблем, а именно: перенос уже апробированных на сотрудниках обычных офисов стратегий кадрового управления на удаленные организационные структуры, либо узкая ориентация на дистанционное управление персоналом. В то же время дистанционный менеджмент имеет высокий риск, если стратегия корпорации не совершенствовалась и требует инвестиций в технологии, а также в обучение команды.

Необходимо внедрение на практике новых подходов, консолидирующих решения для дистанционного управления персоналом, дистанционного управления филиалами для территориально распределенных компаний и дистанционного обучения. При внедрении таких инновационных решений любая компания не обходится без этапа разработки и тестирования организационных решений на моделях.

В настоящее время для моделирования и анализа бизнес-процессов используются различные подходы и средства. В первую очередь следует отметить методологии описания бизнес-процессов IDEF, ARIS, BPMN, которые позволяют получить наглядные графические нотации для исследуемых решений [2]. При этом в программных продуктах, предназначенных для описания бизнес-процессов, нередко присутствуют модули, обеспечивающие проведение имитационного эксперимента на построенных событийных моделях (BPWin, Arena, ARIS Simulation).

1. Постановка задачи

В статье рассматривается диверсный подход к жизненному циклу разработки и внедрения модели управления процессами поведения сотрудников удаленного офиса. Целью работы является описание базовой концепции работы распределенного офиса и на ее основе разработки комплекса имитационных моделей. Предложено использовать два подхода – дискретно-событийное и агентное моделирование, что позволило повысить адаптивность и унификацию моделей к реальным объектам.

2. Концептуальная модель процессов взаимодействия сотрудников удаленного офиса

Распределенную систему коммуникации сотрудников можно отнести к классу сложных систем и необходимо рассматривать на различных уровнях абстракции в виде иерархии. Одним из эффективных подходов к анализу сложных систем с точки зрения их проектирования является системный анализ. При выполнении системного анализа будем придерживаться методологии SADT (Structured Analysis and Design Technique – структурный анализ и проектирование) [3, 4], которая представляет собой совокупность методов, правил и процедур, предназначенных для построения функциональной модели предметной области, которая отображает функциональную структуру, производимые функции и действия, а также связи между ними.

Сущность подхода SADT заключается в декомпозиции (разбиении) системы на автоматизируемые функции, которые в свою очередь делятся на подфункции, функциональные задачи и так далее. Процесс декомпозиции продолжается вплоть до определения конкретных процедур. По этой методологии субъектом моделирования служит исследуемая система. Однако моделируемая система никогда не существует изолированно и всегда связана с окружающей средой. Причем зачастую трудно сказать, где кончается система и начинается среда. По этой причине в SADT подчеркивается необходимость точного определения границ системы [4].

В рамках методологии SADT рассмотрим исследуемую систему с использованием подхода построения IDEF0 диаграмм. Целью методологии IDEF0 является построение функциональной схемы исследуемой системы, описывающей все необходимые процессы с точностью, достаточной для однозначного моделирования деятельности системы. В основе методологии лежат четыре основных понятия: функциональный блок, интерфейсная дуга, декомпозиция, глоссарий. Моделируемая система проектируется с целью улучшения качества решения задач удаленными сотрудниками. Система представлена с точки зрения руководителя удаленного офиса (рис.1).

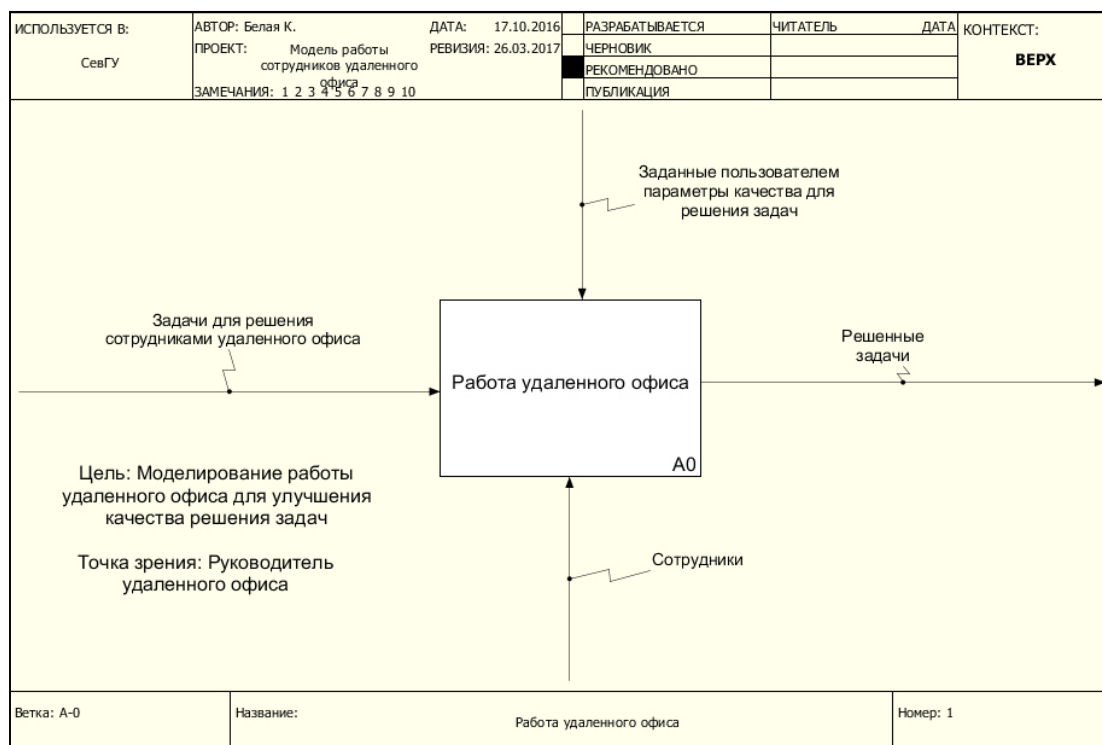


Рис. 1 – Контекстная диаграмма исследуемой системы

На рис. 1 представлена контекстная диаграмма верхнего уровня, где стрелками указаны входные и выходные данные, а также механизмы управления и параметры. Далее в соответствии с принципами системного анализа необходимо выполнить декомпозицию исходной системы на подсистемы и отдельные модули. Декомпозиция позволяет постепенно и структурировано представлять модель системы в виде иерархической структуры отдельных диаграмм, что делает ее менее перегруженной и легко интерпретируемой (рис.2).

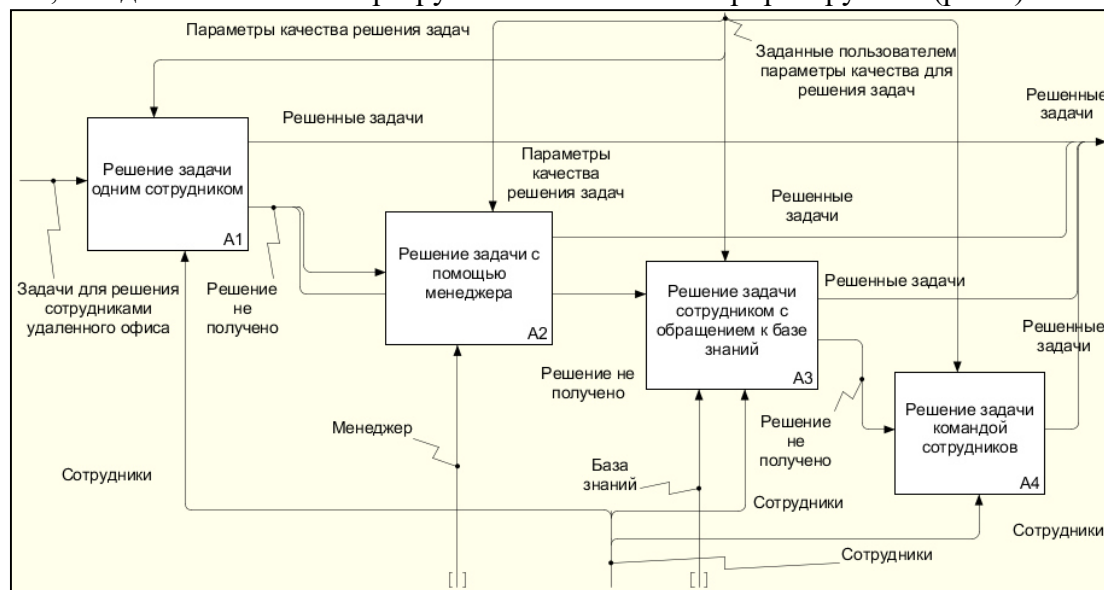


Рис. 2 – Декомпозиция исследуемой системы

Полученная декомпозиция системы не является окончательной. Каждый сценарий поведения сотрудника можно представить в более широком смысле.

Исходную систему можно декомпонировать на следующие структурные модули:

- Решение задачи одним сотрудником.
- Решение задачи с помощью менеджера.
- Решение задачи сотрудником с обращением к базе знаний.
- Решение задачи с помощью других сотрудников (командное решение).

Далее последовательно рассматриваются диаграммы структурно-функциональной декомпозиции по каждому из направлений.

Для сценария процесса «Решение задачи одним сотрудником» на вход подсистемы поступают задачи для сотрудника, на выходе имеем либо готовые решенные задачи, либо информацию о том, что задачи не решены. Механизмом управления в данном случае выступает сотрудник удаленного офиса. (рис.3).

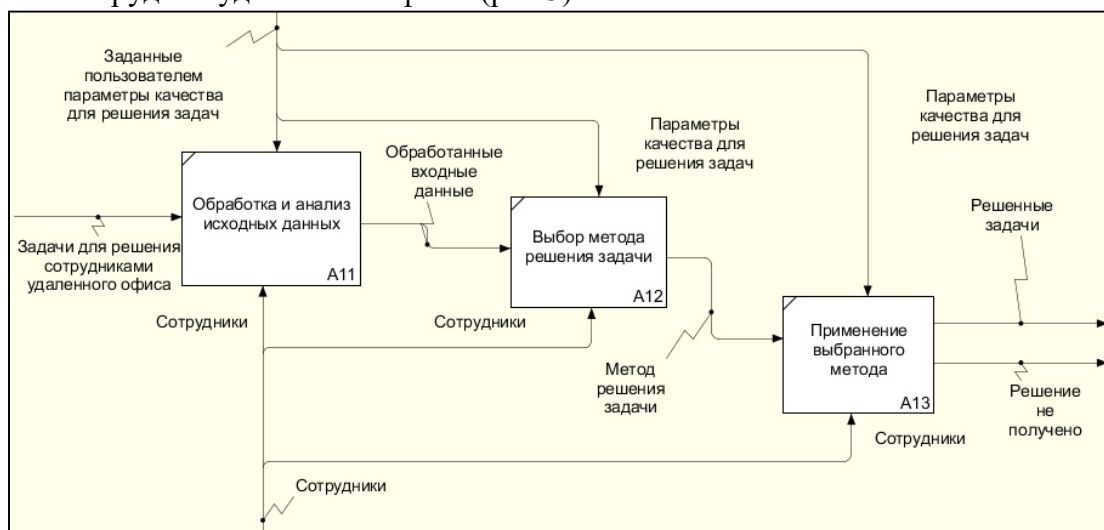


Рис. 3 – Результат декомпозиции сценария A1

Для сценария процесса «Решение задачи с помощью менеджера» на вход подсистемы поступают нерешенные задачи от сотрудника удаленного офиса, на выходе подсистемы – пул решенных задач. Механизмом управления в данном случае выступает менеджер отдела (рис.4).

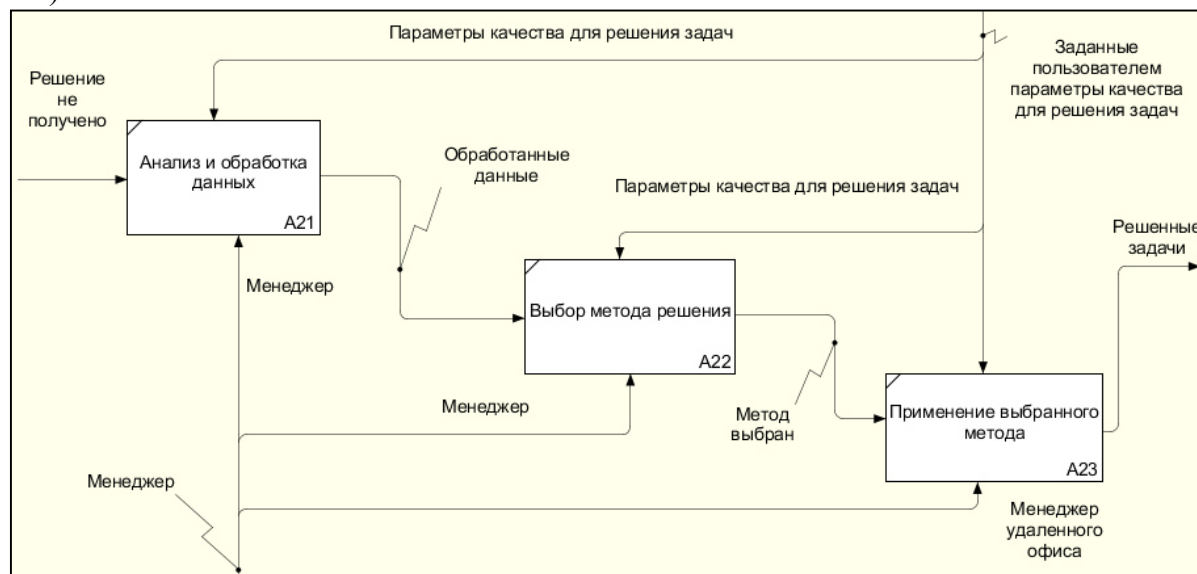


Рис. 4 – Результат декомпозиции сценария A2

Для сценария процесса «Решение задачи сотрудником с обращением к базе знаний» на вход подсистемы поступают задачи для сотрудника удаленного офиса, если сотрудник не может решить задачу (не достаточный уровень компетенций), то он обращается к базе знаний за необходимой информацией, затем пытается снова решить задачу. На выходе — либо решенные, либо нерешенные задачи (рис.5).

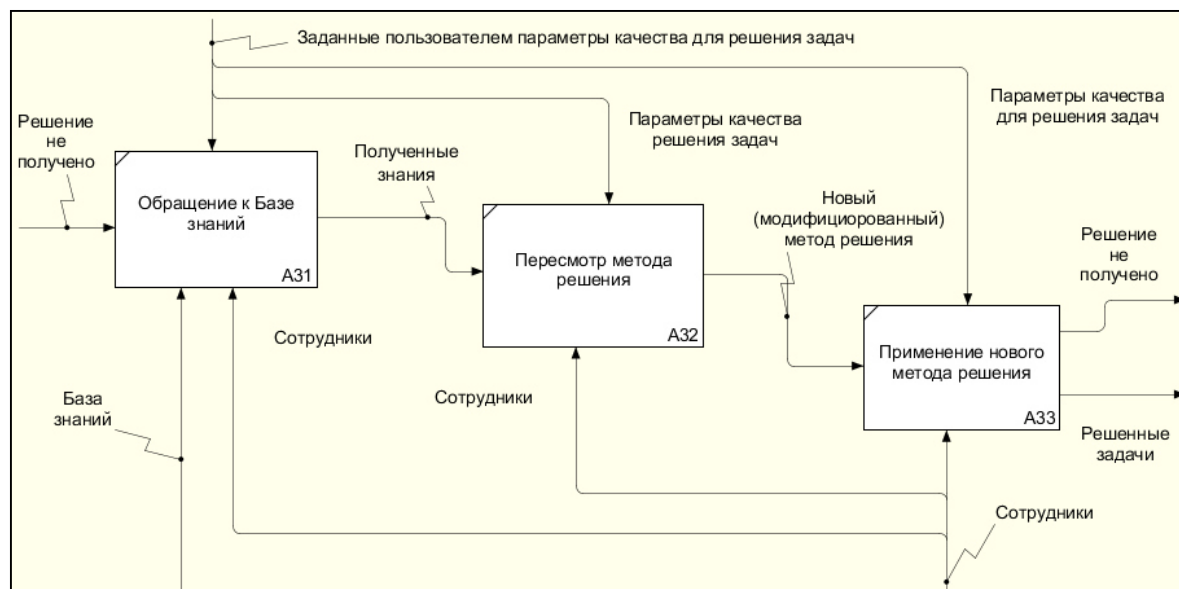


Рис. 5 – Результат декомпозиции сценария 3

Для сценария процесса «Решение задачи с помощью других сотрудников» на вход подсистемы поступают задачи для сотрудника удаленного офиса, на выходе – обработанные задачи, которые были решены с привлечением других сотрудников. В качестве механизмов воздействия используется дополнительные сотрудники, которые компетентны в данной предметной области (рис.6), таким образом данный сценарий иллюстрирует процесс принятия коллективного решения.

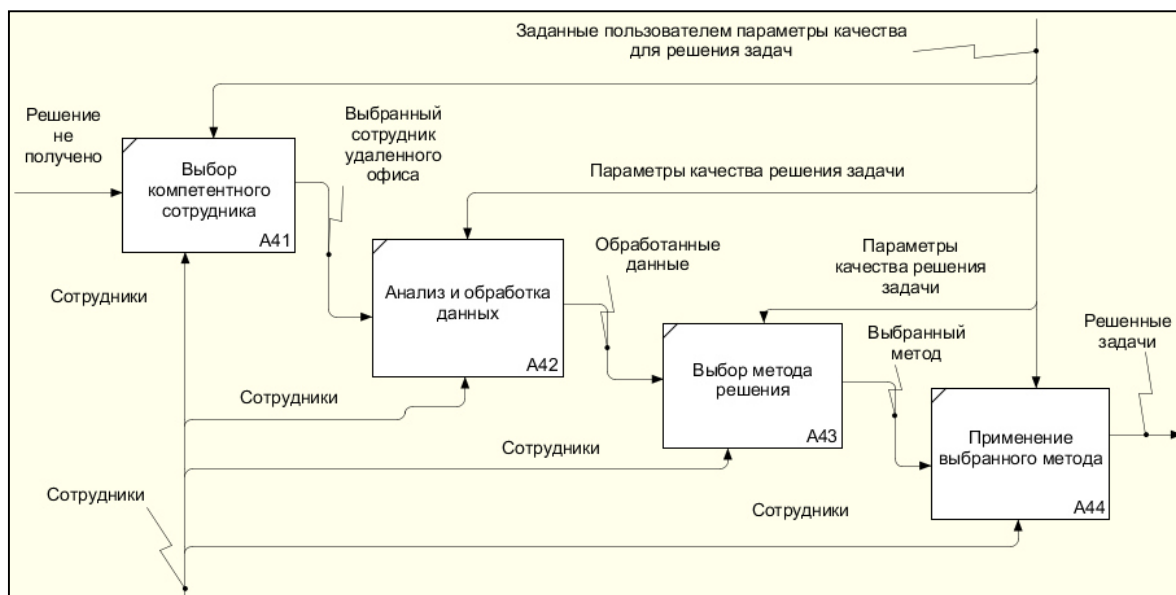


Рис. 6 – Результат декомпозиции сценария A4

Представленные схемы исследуемой системы, описывают все необходимые процессы с точностью, достаточной для однозначного моделирования дальнейшей деятельности, что полностью соответствует целям методологии IDEF0.

3. Дискретно-событийная модель процессов взаимодействия сотрудников удаленного офиса

На основе концептуальной модели в среде моделирования AnyLogic была разработана имитационная модель исследуемой системы. Модель базируется на дискретно-событийном подходе. Особенностью подхода является представление функционирования системы в виде хронологической последовательности событий. Каждое событие происходит в определенный момент времени и знаменует собой изменение состояния системы. Ограниченность подхода в жесткой структурной организации, т. е. в нашем случае проектировщик заранее учитывает сколько будет в модели сотрудников-менеджеров, сотрудников исполнителей и других организационных единиц и на основе этой информации строит структуру модели. Это существенно снижает возможности унификации модели. Например, на рис. 7 приведена модель для 3 сотрудников.

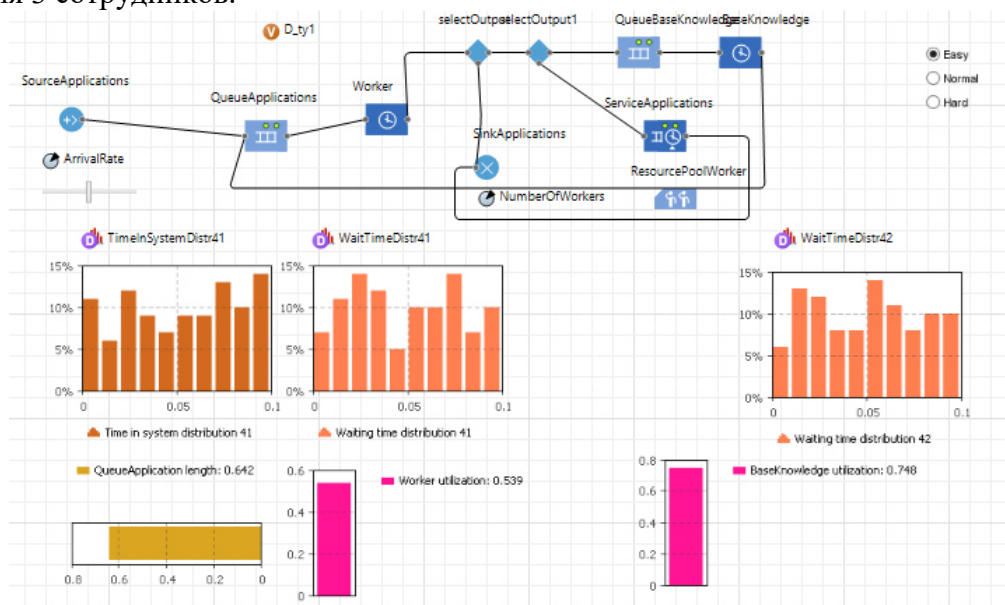


Рис. 7 – Имитационная модель в среде AnyLogic 7 (Дискретно-событийный подход)

Последовательность событий при моделировании процессов обработки задачи включает следующие этапы:

- задачи для решения поступают на из блока *SourceApplications* (генератор запросов) в очередь *QueueApplications* на обработку к сотруднику удаленного офиса;
- обработка задач сотрудником имитируется блоком *Worker*;
- если сотрудник не может решить задачу (ситуация имитируется блоками *SelectOutput*), то он обращается к базе знаний *BaseKnowledge* и затем снова возвращается к пересмотру метода решения;
- если сотрудник после обращения к базе знаний так и не решил задачу, то она поступает к двум другим сотрудникам на обработку, они представлены в виде ресурса *ResourcePoolWorker*, а сам блок обработки задачи дополнительной группой сотрудников – *ServiceApplications*.

В модели выделено три уровня сложности задач: easy, normal и hard. Также, в качестве информационных параметров, на рабочее поле модели вынесены диаграммы изменения времени ожидания задания в системе *TimeInSystemDistr* и времени нахождения задания в системе до полного ее решения *WaitTimeDistr*.

При исследовании модели было рассмотрено три сценария: поступающие задачи на обработку низкого, среднего и высокого уровня трудоемкости. На рис. 8–10 представлены промежуточные результаты моделирования исследуемой системы, время моделирования 8 часов (полный рабочий день).

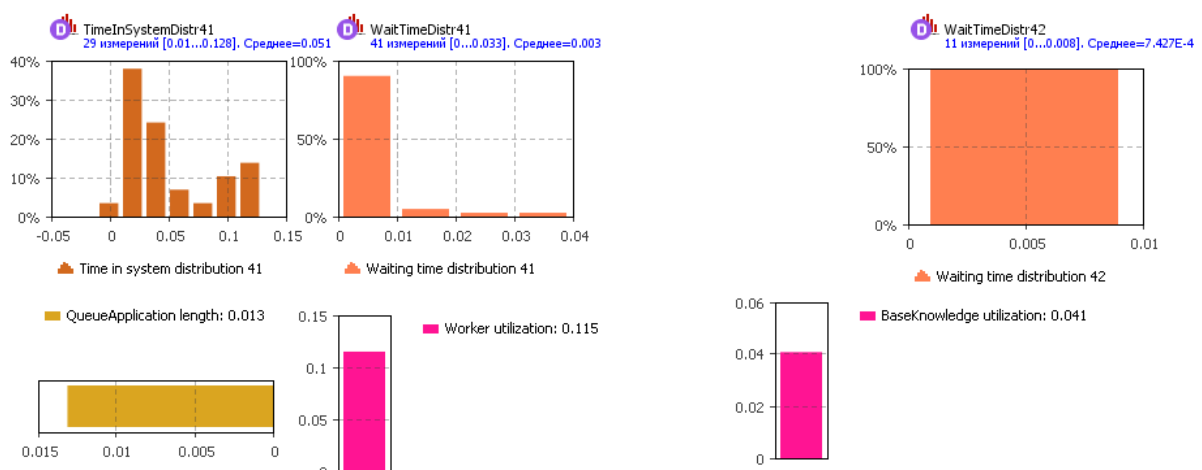


Рис. 8 – Моделирование системы для задач уровня Easy

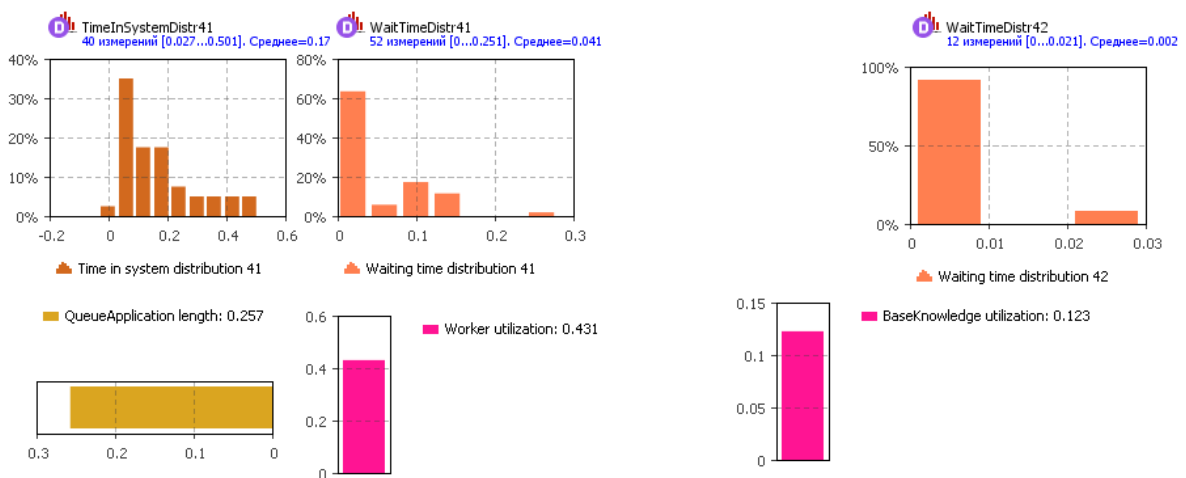


Рис. 9 – Моделирование системы для задач уровня Normal

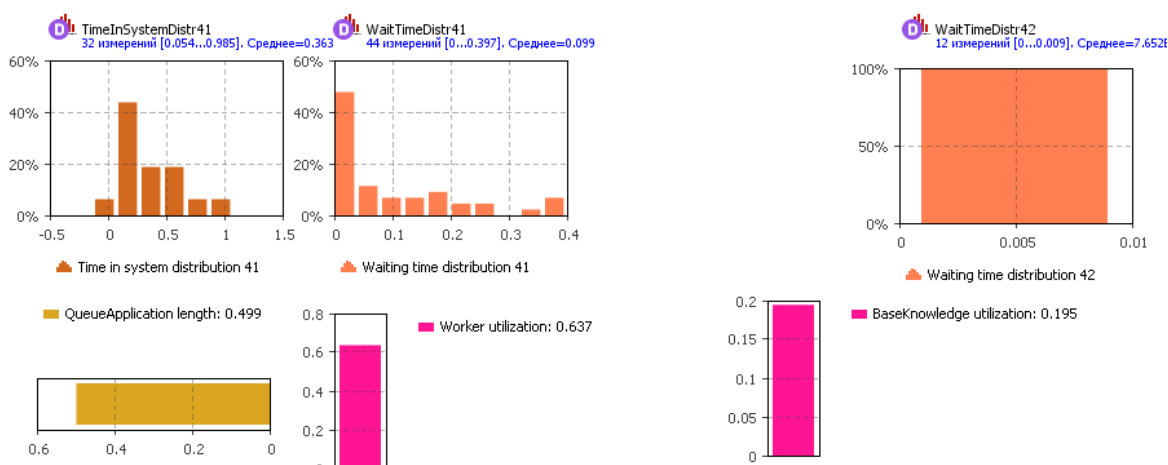


Рис. 10 – Моделирование системы для задач уровня Hard

Исследование модели показало, что наибольшая загрузка работников удаленного офиса наблюдается при решении задач высокого уровня, она составляет около 63.7%. При этом все задачи, поступившие на обработку, успешно решены. Всего за период исследования было отправлено 32 задачи, 30 задач решил работник сам, из них 10 он решил, обращаясь к базе знаний, а 2 задачи были решены другими сотрудниками. Меньше всего работник загружен при решении задач низкого уровня, его загрузка составила 11.5%.

4. Агентная модель процессов взаимодействия сотрудников удаленного офиса

Исследуемую систему можно также представить в виде агентной модели. Агентное моделирование – метод имитационного моделирования, исследующий поведение децентрализованных агентов и то, как это поведение определяет поведение всей системы в целом. При разработке агентной модели, разработчик вводит параметры агентов (это могут быть люди, компании, активы, проекты, транспортные средства, города, животные и т.д.), определяет их поведение, помещает их в некую окружающую среду, устанавливает возможные связи, после чего запускает моделирование. Индивидуальное поведение каждого агента образует глобальное поведение моделируемой системы.

Обобщенная модель взаимодействия может быть представлена в виде кортежа:

$$M = \langle SA, SE, SP, DI, SRA \rangle,$$

где SA – множество агентов-сотрудников, участвующих в обработке задач; SE – множество коммуникативных сред распределенных офисов, где происходит взаимодействие сотрудников; SP – множество исследуемых проблемных областей (например: уровень технической и технологической компетенции сотрудников; уровень коммуникационной активности); DI – множество сценариев диалога между участниками взаимодействия; SRA – множество отношений показывающих возможность осуществления процесса взаимодействия между парой агентов-сотрудников.

На основании накопленной информации определяются количественные параметры, необходимые для оценки эффективности работы каждого сотрудника, такие как: количество сообщений от одного сотрудника к другому; количество оформленных в базе данных электронных документов отчетов; количество переданных сообщений, за период исследований. На основе этих данных производится оценка эффективности системы. При реализации рассматриваемой концепции для обеспечения достаточного уровня адекватности модели обязательным условием является выполнение следующих ограничений и допущений:

- модель отражает только те направления деятельности сотрудника, которые касаются непосредственно процесса выполнения им функциональных задач;
- набор базовых правил функционирования (в том числе связанных с предметной областью) даётся агенту на этапе разработки, что позволяет избавить лицо, принимающее решение от этапа первичного обучения интеллектуального агента.

На рис. 11 представлена модель, построенная на основе многоподходной концепции, однако основное внимание в ней уделено агентному представлению сотрудников-исполнителей исследуемой системы [5]. Модель эмулирует взаимодействие интеллектуальных сущностей: менеджер, сотрудник удаленного офиса, база знаний и распределенная программная система (РПС). Базовый алгоритм взаимодействия сотрудников удаленного офиса заключается в следующем:

- менеджер головного офиса посылает запросы на решение функциональных задач с определенными параметрами (предельное время решения, трудоемкость, требуемые компетенции исполнителей);
- РПС распределяет задачи по сотрудникам в соответствии с их компетенциями;
- запросы на решение функциональных задач обрабатываются сотрудниками удаленного офиса в соответствии с распределением РПС;
- не обработанные функциональные задачи ожидают обработки в буфере ожидания;
- за несвоевременную обработку в системе удаленного офиса начисляются штрафные баллы.

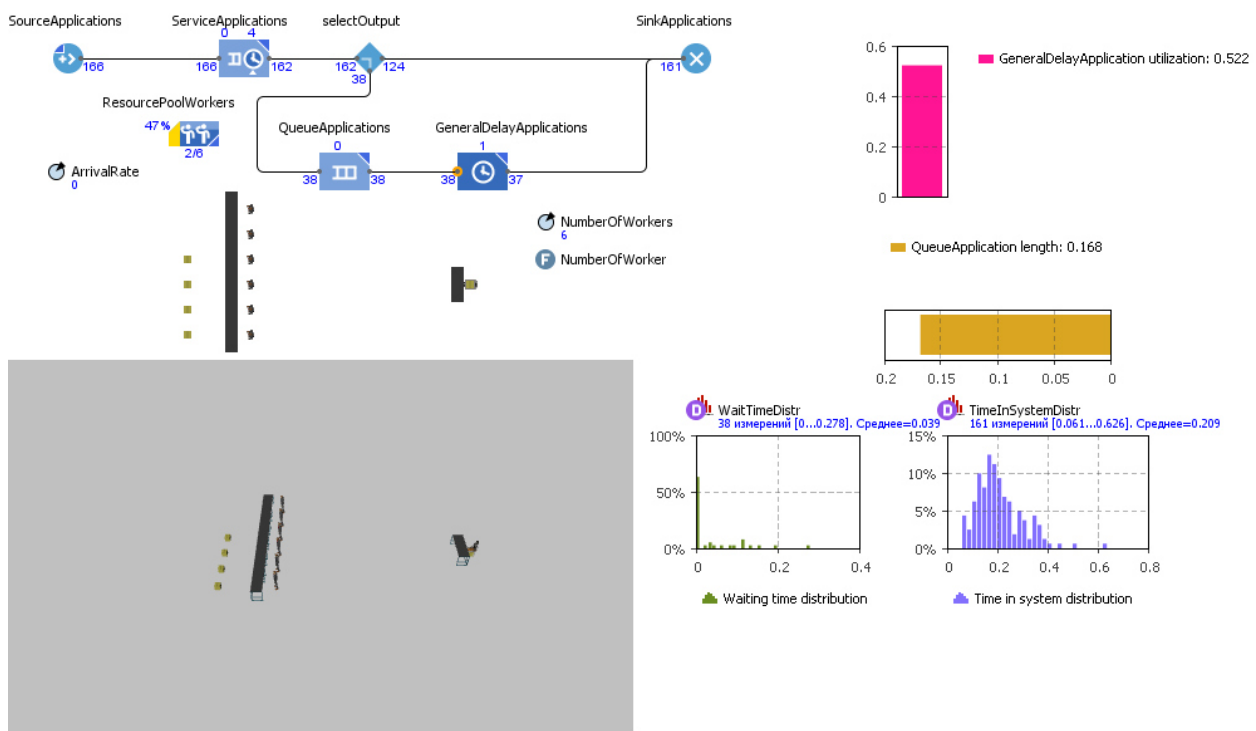


Рис. 11 – Моделирование системы с использованием интеллектуальных агентов

В схеме рис.11 присутствуют следующие блоки и обозначения: *SourceApplications* – источник заданий для сотрудников офиса (запросы от менеджера); *ServiceApplications* – буфер для хранения заданий, поступающих из источника; *ResourcePoolWorkers* – сотрудники офиса, обрабатывающие функциональные задания; *QueueApplications* – буфер для хранения не решенных заданий; *GeneralDelayApplications* – главный сотрудник офиса (менеджер), который обрабатывает задания, не решенные сотрудниками офиса; *NumberOfWorkers* – параметр, отвечающий за количество работников (участвует в оптимизации); *NumberOfWorker* – функция, отвечающая за издержки заработной платы сотрудников (участвует в оптимизации); *WaitTimeDistr* – время нахождения функциональной задачи в системе до полного ее решения; *TimeInSystemDistr* – время ожидания начала обработки для задачи в системе.

Основными агентами в системе являются агенты сотрудники удаленного офиса и агенты функциональные задачи, поступающие в систему. Сценарий поведения агентов может быть описан логическими схемами (*Statechart*), включающими функциональные блоки и связи между ними. Логические схемы для интеллектуальных агентов представлены на рис. 12.

Агент типа *Сотрудник удаленного офиса* имеет 2 состояния, т.е. работник может быть в состоянии «свободен» или быть в состоянии «занят» при обработке запроса на решение задачи. Также предусмотрен параметр «svoboden», для передачи информации о состоянии сотрудника другим интеллектуальным агентам удаленного офиса.

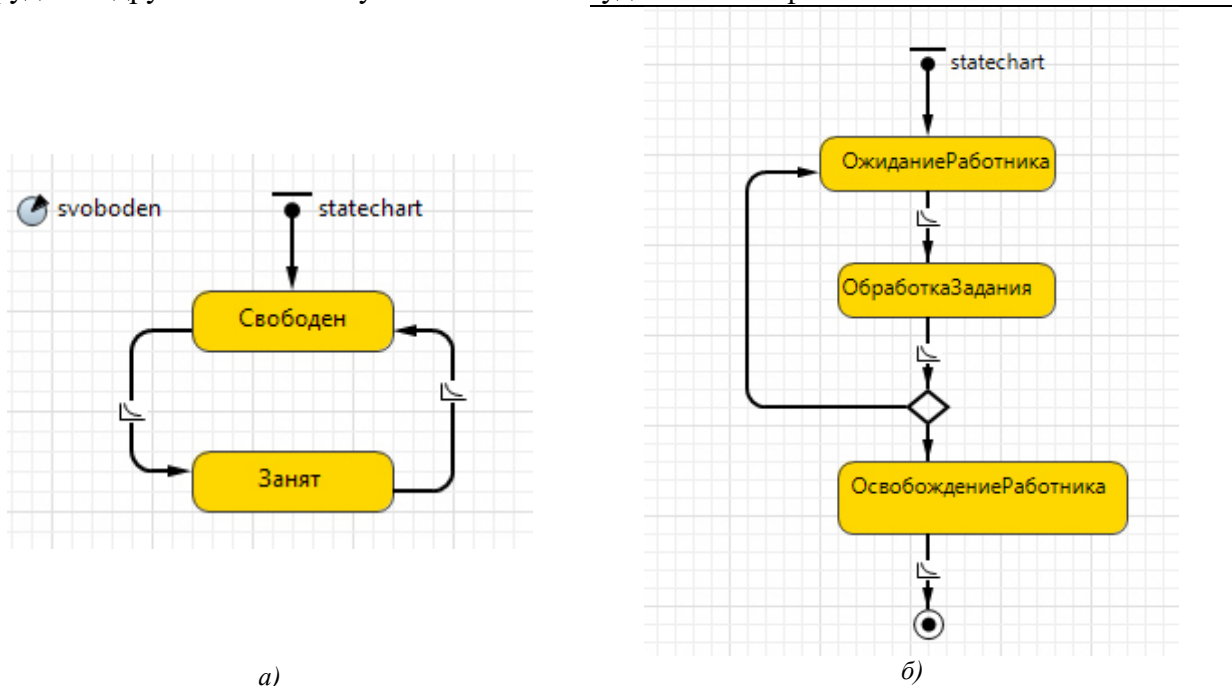


Рис. 12 – Логические схемы для интеллектуальных агентов: а) агент «Сотрудник удаленного офиса»; б) агент «Запрос»

Сценарий действий для агентов типа *Запросы* 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.

С целью определения эффективного состава удаленного офиса при котором с одной стороны минимизируются издержки на содержание сотрудников (зарплата) и с другой стороны повышается качество решенных функциональных задач (минимум времени решения пакета задач, поступающих от менеджера) проведен оптимизационные эксперименты. На рисунках 13 и 14 приведены их результаты.

RemoteOffice : Optimization

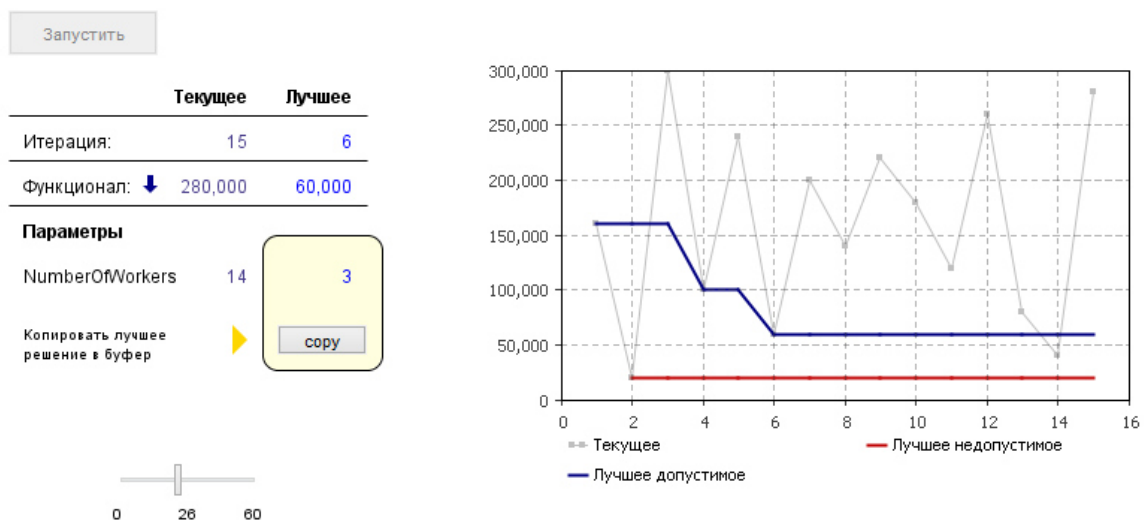


Рисунок 13 – Результаты оптимизационного эксперимента

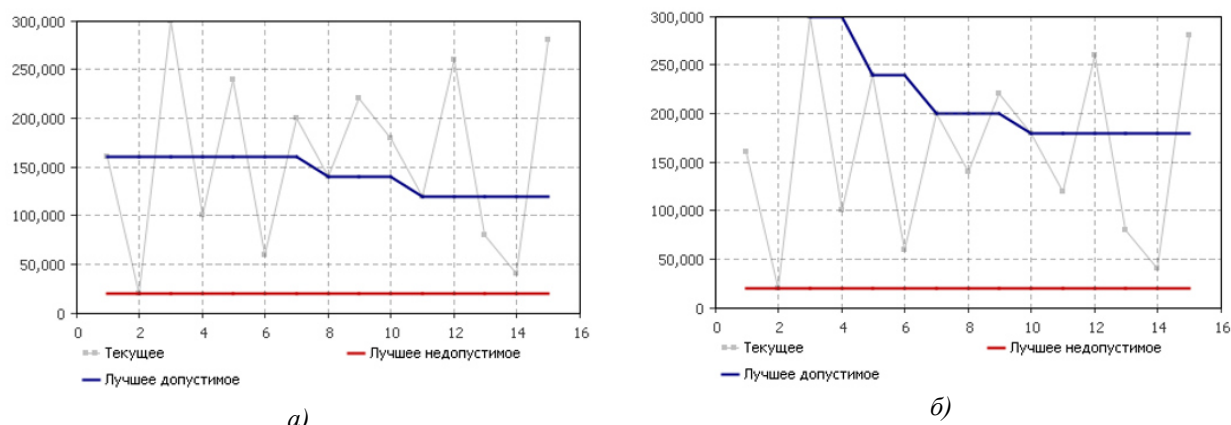


Рисунок 14 – Результаты оптимизационного эксперимента: а) 44 задания в час требуется 6 сотрудников; б) при интенсивности заданий 60 в час, требуется 9 сотрудников

При интенсивности заявок 26 в час, оптимальное количество работников для качественного выполнения заданий равно 3 (менеджер удаленного офиса не входит в это число). При увеличении интенсивности поступления запросов соответственно увеличивается необходимое число сотрудников удаленного офиса для их обработки. Настраивая временные параметры модели под конкретные свойства функциональных задач результаты таких экспериментов могут быть использованы при определении управляющим звеном компании эффективного кадрового состава и набора функциональных задач.

Выводы

В статье рассмотрен диверсный подход к моделированию процессов взаимодействия сотрудников удаленного офиса. А именно: на этапе описания концептуальной модели процессов взаимодействия сотрудников использована методология структурно-функционального анализа на базе стандартов IDEF0; на этапе построения имитационной модели использованы два подхода – дискретно-событийный и агентный. Это позволило построить имитационные модели учитывающие последовательность этапов обработки функциональных задач при жестко заданной структурной организации удаленного офиса и оптимизационные модели эффективной структурной организации удаленного офиса.

Перспективой дальнейших исследований является разработка комплекса имитационных моделей с использованием оптимитационно-имитационного подхода и построение на их основе системы поддержке принятия решений, основанной на базе знаний интеллектуальных агентов.

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. Гусятников В.Н., Безруков А.И. Стандартизация и разработка программных систем. – М: Финансы и статистика, 2010.
3. Марка Д., МакГоуэн К. Методология структурного анализа и проектирования SADT Structured Analysis & Design Technique. Часть 1. – М: Мета Технология, 1993. – 85 с.
4. Методические указания для выполнения лабораторных работ на тему «Структурный подход к моделированию критических систем» по дисциплине «Моделирование критических процессов» для магистров дневной и заочной форм обучения по направлению 09.04.01 «Информатика и вычислительная техника»/ Сост.: В.И. Шевченко – Севастополь: 2016. – 40 с.
5. Белая К.В., Шевченко В.И. Использование мультиагентного подхода для моделирования взаимодействия сотрудников распределенных офисов //Материалы всероссийской научно-технической конференции молодых ученых, аспирантов и студентов «ИСУМ – 2016», Севастополь 19-21 мая 2016 г.–Севастополь: Изд-во СевГУ, 2016.– С.328-332

УДК 004.94

А.В. Гылка, студент 3-го курса

Научный руководитель: А.Н. Скаковская, канд. техн. наук., доц.

Севастопольский государственный университет

МОДЕЛИРОВАНИЕ ТРЕХМЕРНЫХ ОБЪЕКТОВ В AUTODESK 3DS MAX

Аннотация: Рассматривается трехмерное моделирование для проектирования интерьера дома с помощью программного пакета Autodesk 3ds Max и создание визуальных эффектов с использованием скрипта Fastrender.

Ключевые слова: – 3D моделирование, анимация, визуальные эффекты, плагин.

Annotation: Three-dimensional modeling for interior design of the house is considered using the Autodesk 3ds Max software package and creating visual effects using the Fastrender script.

Key words: 3D modeling, animation, visual effects, plugins.

Сегодня Autodesk 3ds Max является самым современным средством решения графических задач и один из самых мощных интерактивных средств компьютерной графики. [1-4]. 3ds Max и специализированные приложения на его основе нашли широкое применение в машиностроении, строительстве, архитектуре и других отраслях промышленности. На сегодняшний день 3D визуализация архитектурных объектов является достаточно прибыльным бизнесом. Так, расценки на архитектурный проект колеблются от \$100 до \$10,000. Архитектурное моделирование в 3ds Max пользуется повышенным спросом, так как дает заказчику возможность оценить внешний вид здания до его постройки, в отличие от того же AutoCAD.

В работе выполнен анализ технологических возможностей системы автоматизированного проектирования в сфере 3D-моделирования. Компания Autodesk сделала свою продукцию достаточно простой в использовании, а главное доступной: разработаны бесплатные лицензионные версии для обучения в домашних условиях. Ранние версии 3ds Max в основном базировались на создании чертежей моделей, позже появился набор инструментов для объемного проектирования.

Данная программа имеет большое практическое значение: использование 3D-модели позволяет изучить объект еще до того, как он создан. Трехмерную модель можно осмотреть, вращать, измерить ее свойства. Полученная информация позволяет провести анализ будущего объекта (его размещения, габариты и т.д.).

В работе выполняется достаточно актуальная задача: создание архитектурного проекта дома. Такой проект дает возможность рассмотреть детали, которые практически невозможно рассмотреть на обычном чертеже (рисунок 1).

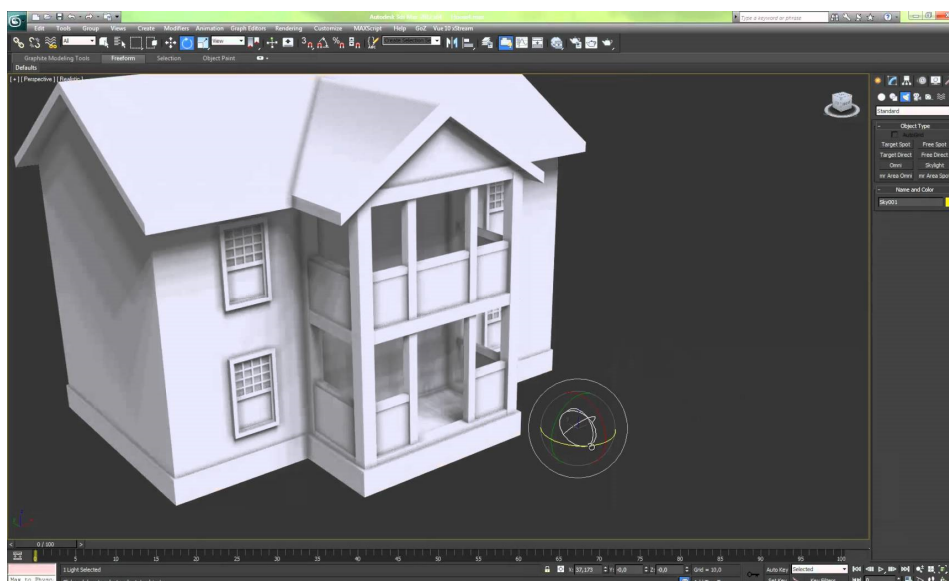


Рис. 1. Изображение трехмерной модели дома

Трехмерное моделирование - относительно трудоемкий процесс. Главная задача, решаемая системой трехмерного моделирования - это значительное сокращение времени построения моделей. Система 3ds Max содержит все необходимые инструменты для визуализации, такие как освещение, материалы, текстуры, рендеринг. Данные инструменты удобны для использования и удовлетворяют потребности большинства пользователей.

В процессе 3D моделирования пользователь часто сталкивается с такими проблемами как недопонимание между заказчиком и исполнителем, несоответствие конечного результата с моделью и т.п. Одной из главных проблем является неспособность заказчика правильно выразить свои требования исполнителю.

Недостатком проектирования в 3ds Max является то, что 3D программы требуют большой объем оперативной памяти и памяти на жестком диске. Также такие программы требуют много времени на обработку и вычисление данных, что означает – чем больше файл, тем больше времени уйдет на его обработку (рендеринг). Рендеринг может занять, как несколько минут, так и несколько часов. Суть рендеринга в том, чтобы передать естественную форму 3D модели и выполнить все наложенные на нее эффекты. К примеру, круглый стеклянный стол в 3D измерении не будет прозрачным, это будет фигура с наложенным на нее эффектом. При этом рендеринг выполняет эффект стеклянной и прозрачной поверхности, при этом создавая тень на других объектах в зависимости от расположения источника света.

Для получения конечного результата необходимо выполнить все действия над объектами и запустить рендеринг. Если рендеринг не даст ожидаемый результат, то нужно изменить наложенные эффекты и заново запустить его. Так повторять цикл до получения нужного результата.

Описанный способ очень трудоемкий - требует много ресурсов и времени. Для решения этой проблемы предлагается установить дополнительный скрипт, который в разы ускорит рендеринг, при этом сохранит все нужные детали и позволит визуально оценить работу. Для этого в своей работе я использую скрипт Fast Render. Его главной задачей является тестовый рендер перед выполнением полного рендера со всеми возможными деталями. Таким образом, Fastrender позволяет экономить время разработчика, давая возможность изменить модель заранее и ускорить сроки сдачи работы. Так же скрипт не нагружает процессор и использует относительно мало оперативной памяти.

Библиографический список

1. Харьковский А.В. 3ds Max 2013 /А.В. Харьковский // Лучший самоучитель. – М., 2012. – 480с.
2. Дымный А.А. Неорганическое моделирование в 3ds Max: видео урок / А.А. Дымный. – 2013.
3. Бондаренко М.Ю., Бондаренко С. В., 3ds Max 2008 за 26 уроков. Издательский дом «Диалектика», 2008. - 304 с.
4. Кулагин Б.Ю, Яцюк О.Г. 3ds Max в дизайне среды. – С.: «БХВ-Петербург», 2008. – 976с.

УДК 004.942

В.Д. Полетаева, студентка 4-го курса

С.С. Сысоев, студент 4-го курса

Научный руководитель: Д.Ю. Воронин, канд. техн. наук, доц.

Севастопольский государственный университет

ИМИТАЦИОННЫЙ СТЕНД ПОДДЕРЖКИ ПРИНЯТИЯ РЕШЕНИЙ РАСПРЕДЕЛЕНИЯ ЗАДАЧ В РАСПРЕДЕЛЕННЫХ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМАХ

Аннотация: в данной статье рассматривается распределенные вычислительные системы как способ решения проблемы директивного срока для задач, связанных с нахождением скорости поверхностного течения с помощью метода спутниковой альтиметрии.

Ключевые слова: распределённые вычислительные системы, скорость течения, директивный срок, спутниковая альтиметрия

Annotation: The paper deals with distributed computing system as the solution with the target completion date of the surface current velocity by means of satellite altimetry.

Key words: Distributed computing system, Surface current velocity, target completion date, satellite altimetry.

Проблема расчёта скорости поверхностного течения воды очень актуальна в современных реалиях мира. Территория Российской Федерации омывается 12 морями Атлантического, Северного Ледовитого и Тихого океанов, поэтому на долю транспортировки грузов морскими путями выпадает более 50% [1]. Следовательно, имея информацию о поверхностных скоростях течения, имеется возможность уменьшить стоимость грузоперевозки и, как следствие, повысить рентабельность такого вида транспортировки в целом.

Существует несколько методологий расчёта поверхностных скоростей течения, в таблице 1 подробнее рассмотрим метод дрейфующих буёв и метод спутниковой альтиметрии [2].

Таблица 1. Сравнение методов расчёта скорости поверхностного течения

Метод	Дрейфующих буёв	Спутниковой альтиметрии
Отличительные особенности	Существенное количество дрейфующих буёв собирают информацию, перемещаясь по акватории исследуемой области.	Источником информации в спутниковом методе является серия фотографий водной поверхности сделанных в ИК-диапазоне. На них отслеживаются движения масс воды, которые можно идентифицировать по различной температуре. Сопоставление нескольких таких снимков, произведённых с небольшим временным интервалом (до суток), позволяет отследить направление движения водных масс и, как следствие, рассчитать скорость поверхностного течения.
Достоинства	Полученная информация не требует постобработки.	Обширная территория изучения акватория при относительно малых финансовых затратах
Недостатки	Относительно малый объём данных представляемый таким устройством.	Метод предполагает выполнение сложных и трудоёмких математических вычислений. Погодные условия являются важным фактором для корректной работы метода.

Очевидно, для того, чтобы получить результаты методом спутниковой альтиметрии, требуется совершить ряд действий: преобразование фотографии в матрицу температур воды, провести расчёт по нескольким парам снимков из одной серии, провести верификацию и валидацию полученных векторов скорости с помощью математических равенств и информации с плавающих буёв (если таковая имеется).

Для того чтобы не нарушать директивный срок исполнения задачи расчёта скорости поверхностного течения спутниковым методом, требуется иметь вычислительные мощности. Например, эффективным способом уменьшения времени расчёта является использование распределённых вычислительных систем (РВС) - набор независимых компьютеров, представляющий пользователям единой объединённой системой, и направленный на решение определённой общей задачи [3]. Главными преимуществами таких систем является возможность использования разнородных ресурсов для вычисления специфических задач, масштабируемость и устойчивость к сбоям.

Ввиду того, что проводить натурные эксперименты не рентабельно, проведём теоретические исследования с помощью имитационной модели РВС, которая представлена на рисунке 1.

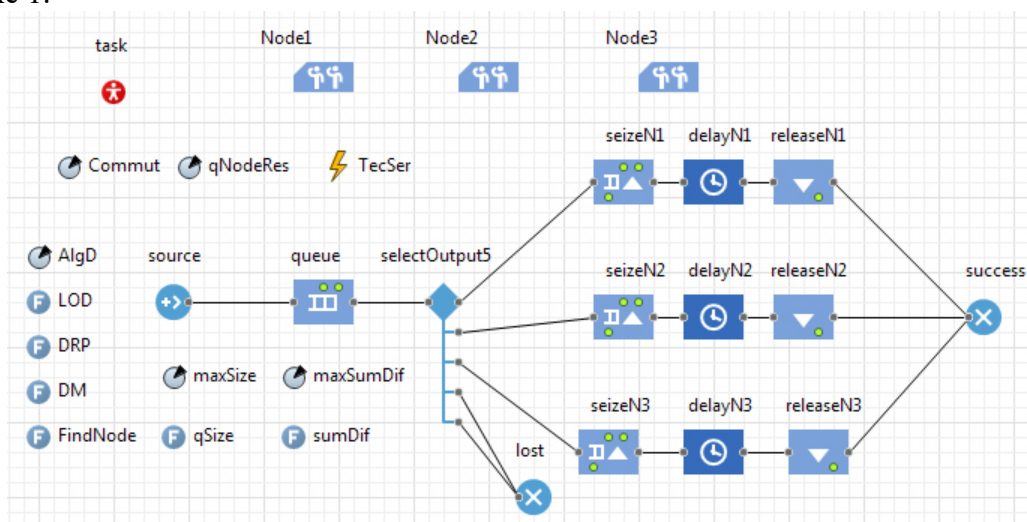


Рисунок 1. Имитационная модель РВС

На вход такой модели поступает пакет, который состоит из подзадач, таких как сложение, умножение, дифференцирование, интегрирование и т.д. Каждая из подзадач имеет следующие атрибуты:

- Матрицу связей, которая отображает всех предшествующие подзадачи, без которых невозможно выполнение текущей подзадачи.
- Сложность вычислительных работ (например, сложение имеет самую низкую сложность, а дифференцирование – самую высокую).
- Приоритет.
- Предварительную оценку требуемой мощности.
- Директивный срок всего пакета.

Сценарий обработки пакета следующий:

1. Поступление пакета задач на вход системы, предварительная оценка времени исполнения задачи, выставление приоритетов.
2. Поступление очередной подзадачи в очередь на выполнение.
3. Проверка директивного срока. Если он нарушается, то снятие всех подзадач с системы. Начисление штрафа.
4. Работа алгоритма диспетчеризации для корректного распределения нагрузки на узлы РВС.
5. Поступление подзадачи на узел системы, выполнение подзадачи. На выходе производится проверка директивного срока аналогично п.3. Если срок не нарушается переход к п.2. Если подзадача была конечной и директивный срок не нарушен, начисляется вознаграждение.

При использовании подобной РВС при вычислении скорости поверхностного течения методом спутниковой альтиметрии распределенные вычисления позволят сократить реальное время расчёта задачи не менее чем в 2 раза. Следовательно, этот способ расчета можно будет использовать с большей эффективностью в различных критических ситуациях.

Библиографический список.

1. Морской транспорт России. Источник: <http://helpiks.org/4-57254.html>
2. Бабий В.И., Погребной А.Е., Ратнер Ю.Б., Станичный С.В. Возможности восстановления поля скорости поверхностных течений по спутниковым данным (на примере Чёрного моря)//Сб.науч.трудов/ Системы контроля окружающей среды. – Севастополь: «ЭКОСИ-Гидрофизика»б 2002. – с.254-261
3. Таненбаум Э., Ван Стен М. Распределенные системы. Принципы и парадигмы. - СПб.: Питер, 2003. — 877 с.;

УДК 004.928

С.А. Рудоченко, студент 4-го курса**Научный руководитель: А.Н. Скаковская, канд. техн. наук., доц.***Севастопольский государственный университет***РАЗРАБОТКА ВИЗУАЛЬНЫХ ЭФФЕКТОВ СРЕДСТВАМИ КОМПЬЮТЕРНОЙ ГРАФИКИ***Аннотация: В программном пакете Autodesk 3ds Max созданы 3D модели бокал с водой, автомобиля и визуальные эффекты с использованием различных плагинов, Element 3D и Particular (работа с частицами).**Ключевые слова: 3D моделирование, анимация, визуальные эффекты, работа с частицами.**Annotation: In the software package Autodesk 3ds Max created 3D models of a glass with water, cars and visual effects using various plug-ins Element 3D and Particular (working with particles).**Key words: 3D modeling, animation, visual effects, work with particles.*

В наше время компьютерная графика широко применяется для детальной визуализации инженерных разработок и демонстрации результатов исследований, а также в сфере кино и рекламы, где играет чрезвычайно важную роль при создании захватывающих визуальных эффектов. Существует большое количество мощных программных продуктов для создания визуальных эффектов и 3D-анимации. Например, Autodesk Maya, Autodesk 3Ds Max, Maxon Cinema 4D, Houdini, Real Flow, Blender, Adobe After Effects. Каждый из этих инструментов имеет свои преимущества и недостатки, но основной проблемой, с которой встречается большинство специалистов VFX индустрии - это большие потери времени на рендеринг и постоянную конвертацию сцен с одного программного продукта в другой.

Детальный анализ различных методик разработки визуальных эффектов средствами компьютерной графики показал, что к процессу создания визуальных эффектов, особенно если в сценах присутствует симуляция жидкостей, паров, а также взаимодействия частиц в различных физических явлениях, привлекается большое количество программных продуктов, но не всегда с удобным решением.

Например, для создания сцены с водой и бокалом (рис.1) было применено два программных пакета: Autodesk 3ds Max и Next Limit Real Flow. В данном случае бокал и вишня были смоделированы в Autodesk 3ds Max, а жидкость - в предназначенном для симуляции жидкостей, паров и других манипуляций с частицами Next Limit Real Flow.

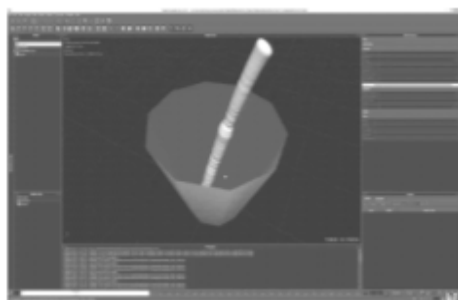


Рис. 1. Процесс создания сцены с жидкостью в программе Real flow и результат рендера в 3Ds Max соответственно

Неудобство данного метода заключается в том, что программа Next Limit Real Flow не имеет своего рендера поэтому сначала объекты, с которыми будет происходить взаимодействие веществ моделируются в пакетах трехмерного моделирования, после чего импортируются в Real Flow, затем с учетом взаимодействия с этими объектами в Real Flow моделируется поведение жидкости, пара и т.п. Третьим шагом является импорт из Real Flow снова в трехмерный пакет, где мы настраиваем материалы и рендерим финальную картинку.

В наше время существует много дополнительных плагинов, скриптов и пресетов для программы 3Ds Max, которые позволяют частично, а иногда и полностью, исключить применение других программных пакетов при разработке визуальных эффектов.

В работе рассматривается способ создания некоторых визуальных эффектов с помощью применения плагинов. Рассмотрим процесс создания визуальных эффектов. Возьмём разработку анимации компьютерной игры жанра «гонки». Имеем следующий сценарий: в полной темноте находятся два автомобиля, каждый из которых постепенно должен показать свою форму. Сначала автомобили заводятся, буксуют, едут друг на друга и сталкиваются. Основным условием является то, что автомобили не затекстурированы реалистично.

Начинать процесс создания визуальных эффектов необходимо с разработки аниматика (рис.2). Аниматика - это упрощенная версия анимации, отражающая основные кадры из видео. Аниматика помогает понять, как та или иная сцена будет выглядеть в движении и почувствовать хронометраж сцен. Этот этап является очень важным, поскольку именно во время него легким способом можно избежать серьезных поправок на последующих этапах.

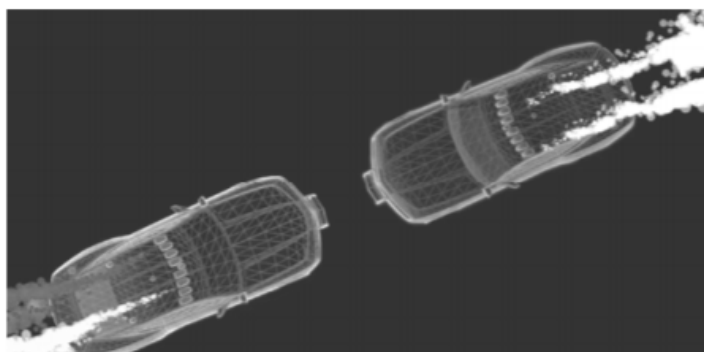


Рис. 2. Процесс аниматика

После утверждения аниматика необходимо перейти к моделированию объектов, которые будут участвовать в сцене. В нашем видео взаимодействуют два автомобиля. Есть возможность использовать готовые модели, но все равно их необходимо приготовить к анимации в пакете 3D моделирования. Мы использовали низко-полигонные модели автомобилей и редактировали их в Maxon Cinema 4D. Далее количество полигонов у моделей увеличили и разделили каждый из автомобилей на корпус и колеса, поскольку именно эти части будут задействованы в анимации. На этом работа в трехмерном пакете завершится. Следующим шагом является импорт данной модели в среду Adobe After Effects. Для каждого из обособленных элементов автомобиля избран так называемый Аух-канал (Auchannel), что позволяет анимировать колеса и корпус. Все выбранные Аух-каналы являются дочерними элементами в группу. Таким образом, перемещение автомобиля в пространстве организуется изменением позиции всей группы, а анимация оборотов колес и изменения углов наклона корпуса – с помощью Аух-каналов.

Следующий этап текстурирования был осуществлен в 3ds Max, в котором установлены дополнительные плагины. Таким образом автомобиль имеет и линии построения, и подсвечивания участков, помогает лучше отразить его форму. Также была смоделирована работа с частицами, поскольку буксования, выхлоп и движение автомобиля сопровождается пылью или газами. В нашем случае это будет пыль с частиц, что светятся. Его создали и настроили с помощью плагина Trapcode Particular.

Таким образом, для создания и визуализации моделей, требуется большие ресурсы, умение работать с различными плагинами для совместной работы с программой в 3ds Max. В работе были применены плагины Element 3D и Trapcode Particular, которые предназначены для работы с анимацией, текстурированием моделей и созданием системы частиц соответственно. Рассмотрена методика создания визуальных эффектов, которая интенсивно используется в кинематографе.

УДК 004.415

К.Г.Рыдзывыло, магистрант**В.И. Шевченко, канд. техн. наук, доцент***Федеральное государственное автономное образовательное учреждение высшего образования "Севастопольский государственный университет"***АЛГОРИТМЫ ЭФФЕКТИВНОГО РАЗМЕЩЕНИЯ ФАЙЛОВ В РАСПРЕДЕЛЕННЫХ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМАХ***Аннотация: Работа посвящена аналитическому обзору алгоритмов эффективного размещения файлов в распределенных вычислительных системах. В статье рассмотрены следующие алгоритмы: алгоритм на основе критерия суммарного объема запросов; муравьиный алгоритм и генетический алгоритм. Генетический алгоритм представлен двумя модификациями, адаптированными для вычислительной системы.**Ключевые слова: оптимальное размещение файлов, распределенные системы, муравьиный алгоритм, генетический алгоритм.**Annotation: The work is devoted to the analysis of algorithms for efficient file allocation in distributed computing systems. The following algorithms are considered in the article: the algorithm based on the criterion of the total volume of requests; Ant algorithm and genetic algorithm. The genetic algorithm is represented by two modifications adapted for the computer system.**Key words: optimal allocation of files, distributed systems, the ant algorithm, the genetic algorithm***Введение**

На сегодняшний день бизнес-приложения, системы промышленной автоматизации, аналитические информационные системы для научных исследований требуют комплексного подхода к решению задач управления конвергентными данными. При этом увеличение пропускной способности каналов передачи данных, увеличение физической емкости хранилищ не решает проблем эффективной обработки больших данных. Как отмечено в работе [1], современные системы хранения больших объемов данных, во-первых, базируются на распределенных вычислительных структурах, а во-вторых, разделяются на две группы: распределенные файловые системы; распределенные хранилища данных. Предполагается, что файловые системы хранят неструктурированную информацию, а хранилища – структурированную.

При этом особую актуальность приобретает задача эффективного размещения файлов распределенных баз данных, а так же блоков и объектных модулей для интеллектуальных хранилищ данных. Оптимальное размещение баз данных и файлов в узлах распределенных систем с учетом приведенных затрат на размещение и обслуживание позволит получить значительный экономический эффект при создании и эксплуатации больших кластерных систем [2].

В рамках статьи рассматривается базовая концептуальная модель эффективного размещения файлов по узлам конвергентной вычислительной системы, построенная на базе исследований, изложенных в [3, 4]. Задачей данной работы является аналитический обзор современных решений в данном направлении с целью выделения наиболее эффективных алгоритмов и их слабых сторон для дальнейшей модификации и адаптации к разрабатываемому программному комплексу для моделирования процессов распределения файлов. В качестве исследуемых выбраны следующие алгоритмы оптимального размещения файлов в распределенной системе: алгоритм на основе критерия суммарного объема запросов, муравьиный алгоритм и генетический алгоритм.

Концептуальная модель эффективного размещения файлов по узлам конвергентной вычислительной системы

Рассматривается модель конвергентной вычислительной системы (КВС), в которой размещаются специализированные информационные системы (ИС). Особенностью системы является однородность ИС, консолидация данных ИС в единое корпоративное хранилище, конвергентность сетей передачи данных и вычислительного оборудования на которых размещены экземпляры ИС. Для описания параметров концептуальной модели введены следующие обозначения:

$n = 1, N$ – индекс экземпляра информационной системы, входящих в КВС, N – количество экземпляров ИС. Полагаем, что в целях упрощения синхронизации в КВС, все ИС в системе имеют одинаковую структуру метаданных.

M – количество файлов, связанных с информационным обменом в экземпляре ИС.

S – число типов функциональных задач, решаемых в процессе информационного обмена в КВС.

$k = 1, K$ – число автоматизированных рабочих мест пользователей (АРМ), участвующих в процессах информационного обмена в КВС.

A_k – k -ый АРМ в структуре КВС.

f_{nm} – m -ой файл в n -ой ИС, связанный с информационным обменом в КВС.

$d_{msn}^{(1)}$ – элемент матрицы смежности $D_n^{(1)}$, для n -ой ИС, соответствующей графу логической связанности m -го файла с функциональной задачей S -го типа.

$$d_{msn}^{(1)} = \begin{cases} 1, & \text{если } f_{mn} \text{ связан с функциональной задачей } S\text{-го типа} \\ 0, & \text{иначе} \end{cases}$$

$d_{mnk}^{(2)}$ – элемент матрицы смежности $D_n^{(2)}$, для n -ой ИС, соответствующей графу размещения m -го файла в A_k :

$$d_{mnk}^{(2)} = \begin{cases} 1, & \text{если } f_{mn} \text{ размещен в } A_k \\ 0, & \text{иначе} \end{cases}$$

$D_n^{(0-2)}$ – матрица смежности, соответствующая графу размещения m -го файла в A_k в момент времени t_0 .

$dd_{mnk}^{(2)}$ – элемент матрицы $DD_n^{(2)}$, допустимости размещения m -го файла в A_k для n -ой ИС:

$$dd_{mnk}^{(2)} = \begin{cases} 1, & \text{если } f_{mn} \text{ может размещаться в } A_k \\ 0, & \text{иначе} \end{cases}$$

$l = 1, L$ – число узлов системы хранения данных в КВС.

$d_{mnl}^{(3)}$ – элемент матрицы смежности $D_n^{(3)}$, для n -ой ИС, соответствующей графу размещения файла в l -ом узле системы хранения данных:

$$d_{mnl}^{(3)} = \begin{cases} 1, & \text{если } f_{mn} \text{ размещен в } l \\ 0, & \text{иначе} \end{cases}$$

$D_n^{(0-3)}$ – матрица смежности, соответствующая графу размещения m -го файла в l -ом узле системы хранения данных в момент времени t_0 ;

$dd_{mnl}^{(3)}$ – элемент матрицы $DD_n^{(3)}$, допустимости размещения m -го файла в файла в l -ом узле хранения данных для n -ой ИС:

$$dd_{mnl}^{(3)} = \begin{cases} 1, & \text{если } f_{mn} \text{ может размещаться в } l \\ 0, & \text{иначе} \end{cases}$$

λ_{sk} – интенсивность поступления запросов, на выполнение функциональных задач (ФЗ) S -го типа из АРМ A_k .

u_{snk} – объем данных, обрабатываемых при выполнении ФЗ S -го типа в n -ой ИС в A_k .

t_{sk}^I – время генерации запроса на выполнение ФЗ S -го типа в A_k .

t_{sk}^O – время обработки запроса на выполнение ФЗ S -го типа к файлам от A_k .

t_{sk}^H – время пересылки запроса на выполнение ФЗ S -го типа к файлам от A_k .

T_{mnk}^S – общее время, необходимое для выполнения ФЗ S-го типа в A_k .

T_S^{Π} – предельное время, выделяемое на выполнения ФЗ S-го типа в КВС.

$\bar{V}$ – вектор критериев качества управления информационным обменом.

С учетом общей структуры ИС и возможности совместного доступа к файлам данных справедливо ограничение (1), показывающее, что в КВС должен присутствовать хотя бы один экземпляр файла данных m-го типа, размещаемый либо в АРМ, либо на одном из узлов обработки данных:

$$\sum_{n=1}^N \sum_{k=1}^K d_{mnk}^{(2)} + \sum_{n=1}^N \sum_{l=1}^L d_{mnl}^{(3)} \geq 1, \quad \forall m = 1, M. \quad (1)$$

Общее время T_{mnk}^S , необходимое для обработки информации, связанной с выполнением ФЗ S-го типа в КВС определяется согласно (2):

$$T_{mnk}^S = t_{smnk}^F + t_{smnk}^O + t_{smnk}^{\Pi} \quad (2)$$

при условии (3):

$$T_{mnk}^S \leq T_S^{\Pi}. \quad (3)$$

В случае, если условие (3) не выполняется, функциональная задача считается не выполненной и изменения в файлах данных не фиксируются.

Далее, рассмотрим альтернативы алгоритмов размещения файлов с учетом применимости к предложенной модели.

Алгоритм оптимального размещения файлов на основе критерия минимизации суммарного объема запросов

В работе [2] предлагается алгоритм оптимального размещения файлов на основе критерия минимизации суммарного объема запросов. Условием алгоритма также является критерий единства файла, что совпадает с требованиями, изложенными в концептуальной модели. Введены следующие обозначения: а – множество узлов информационной сети (без дифференциации по типам АРМ, система хранения и т.п.); b – множество файлов; F_i – обозначает файл i; Y_j – узел j; V_j – объем памяти узла Y_j ; v_i – объем передаваемых данных файла F_i ; λ_{ij} – интенсивность запросов узла вычислительной системы к файлу F_i ; ω_{ij} – объем данных запроса к файлу F_i ; c_{ij} – стоимостные затраты на хранение файла F_i на узле Y_j .

Двоичные переменные принимают следующие значения (4), (5):

$$X_{ijk} = \begin{cases} 1, & \text{если } j\text{-ый узел посылает запрос к } F_i \text{ в } k\text{-ом узле} \\ 0, & \text{в противном случае} \end{cases} \quad (4)$$

$$\alpha(m) = \begin{cases} 1, & m > 0 \\ 0, & \text{в противном случае} \end{cases} \quad (5)$$

где m – числовой аргумент функции α . В систему введены ряд ограничений (6 – 8).

$$\sum_{k=1}^n X_{ijk} = 1, i = \overline{1, b}, j = \overline{1, a}. \quad (6)$$

$$\sum_i v_i \alpha(p) \leq V_j, j = \overline{1, a}, \quad (7)$$

$$\text{где } p = \sum_{k=1}^a X_{ijk}.$$

$$X_{ijk} \in \{0, 1\}, j, k = \overline{1, n}, i = \overline{1, m}. \quad (8)$$

Функция размещения, которая описывает совокупную минимальную стоимость хранения файлов, будет иметь следующий вид, определяемый выражением (4):

$$\min \sum_{j=1}^a \sum_{i=1}^b c_{ij} \alpha(p). \quad (9)$$

Для решения задачи эффективного размещения файлов (4–9) предлагается использование эвристического алгоритма:

- для каждой пары i, j вычисляются значения величины $Z_{ij} = \lambda_{ij} \omega_i$;
- значения Z_{ij} упорядочиваются по убыванию;
- для очередного Z_{ij} определяются номер файла F_i и номер узла Y_j . Если файл F_i пока не размещен в Y_j узле, проверяется выполнимость ограничений (6),(7). Если поставленные ограничения выполняются, файл размещается;
- если файлы размещены в сети, вычисляются значения критерия оптимизации (9)[1].

Предложенный алгоритм, с учетом увеличения индексности системы (виды узлов по типам) может быть применен для модели (1) – (3). Однако задача в постановке (4) – (9) является NP-сложной задачей линейного программирования. На сегодняшний день для решения таких задач эффективно применяются биоинспирированные алгоритмы [5, 6].

Муравьиный алгоритм распределения файлов в распределенной вычислительной системе

Данный алгоритм появился в результате исследования поведения живых муравьев. Муравей, двигаясь по заданному маршруту, оставляет за собой след пахнущего вещества (феромона). Это вещество влияет на выбор маршрута: муравьем выбирается то направление, где уровень феромона больше. Самоорганизация муравьев (в модели – интеллектуальных агентов) обеспечивается взаимодействием ряда следующих компонентов: случайность, воспроизводимость, положительная обратная связь, отрицательная обратная связь. Основной идеей муравьиного алгоритма является накопление и использование статистических данных, собираемых интеллектуальными агентами муравьями[5].

Введем следующие обозначения:

F_{ij} – количество запросов к файлу i в узле j в единицу времени; X_{ij} – двоичная переменная, принимает значение 1, если файл i расположен в узле j , иначе $X_{ij} = 0$; V_i – объем файла i (передаваемых данных); B_j – объем узла j (вместимость узла хранения); τ_j – уровень феромона на маршруте j ; p – коэффициент испарения феромона; P_j – вероятность размещения файла в узел j ; $i = 1..m$, $j = 1..n$, где m – количество файлов, n – количество узлов.

В задаче (6) – (9) необходимо найти матрицу размещения файлов X . В данной трактовке задачи максимизируется суммарный поток запросов, при этом задача размещения файлов по узлам вычислительной системы имеет вид (10) – (13).

$$L = \sum_{i=1}^m \sum_{j=1}^n F_{ij} V_i X_{ij} = \sum_{i=1}^m \sum_{j=1}^n L_{ij} X_{ij} \rightarrow \max \quad (10)$$

При ограничениях:

$$X_{ijk} \in \{0,1\}, \sum_{j=1}^n X_{ij} = 1, i = \overline{1, m} \quad (11)$$

$$\sum_{i=1}^m V_i X_{ij} \leq B_j \quad (12)$$

Вероятность размещения файла определяется по формуле:

$$P_j = f(x) = \frac{\tau_j F_j}{\sum_{k=1}^n \tau_k F_k} \quad (13)$$

При выборе узла l для размещения файла происходит увеличение значения

$\tau_l : \tau_l = (1 - p)\tau_l + \frac{\tau_l}{d}$, значение d должно быть одного порядка с оптимальным значением целевой функции. При переходе к очередной итерации алгоритма уровень феромона

накапливается: $\tau_j = (1 - p)\tau_j + \sum_{k=1}^n \frac{\tau_k}{d}$ [6].

В работе [6] предложен следующий алгоритм:

- формируется начальный вектор $P(n)$ вероятностей размещения i -го файла в j -й узел (13);
- генерируется случайная величина g , распределенная по закону $P(n)$;
- если $j \leq g \leq j+1$ и выполняется условие (12), то i -й файл назначается в j -й узел и увеличивается значение феромона, иначе – выполняется предыдущий пункт алгоритма;
- вычисляется значение целевой функции по формуле (10).

Этот подход, применительно к модели (1) – (3) позволит существенно сократить время решения оптимизационной задачи. Однако существуют другие, не менее эффективные подходы.

Генетический алгоритм оптимизации размещения файлов в распределенной системе

Эволюционная теория утверждает, что каждый биологический вид целенаправленно развивается и изменяется для того, чтобы наиболее эффективным образом приспособиться к изменениям в окружающей среде. Основным механизмом эволюции является естественный отбор. Его суть состоит в том, что более приспособленные особи имеют больше возможностей для выживания и размножения и, следовательно, приносят больше потомства, чем плохо приспособленные особи. При этом благодаря передаче генетической информации потомки наследуют от родителей их базисные качества. Таким образом, потомки сильных индивидуумов также будут относительно хорошо приспособленными, а их доля в общей массе особей будет возрастать. После смены нескольких десятков поколений средняя приспособленность особей данного вида заметно возрастает.

В теории генетического алгоритма (ГА) используется биологическая терминология в упрощенном виде [7]: хромосома – вектор чисел; ген – бит хромосомы; популяция – совокупность особей.

Классический ГА, согласно [8, 9] включает следующие этапы:

1. Генерация начальной популяции из k особей.
2. Вычисление функции приспособленности каждой особи и отброс наименее приспособленных особей.
3. Выбор среди оставшихся особей двух родителей.
4. Применение к родительским хромосомам оператора скрещивания с вероятностью p_c ; при отсутствии скрещивания предполагается, что дочерние хромосомы равны родительским.
5. Применение к полученным дочерним хромосомам оператора мутации с вероятностью p_m ; предполагается, что в отсутствие мутации хромосомы не изменяются.
6. Итерационное повторение шагов 3–5, пока популяция вновь не будет включать k особей.
7. Повторение шагов 1–6 до тех пор, пока не будет достигнут критерий окончания процесса [9].

На сегодняшний день существует несколько альтернативных вариантов использования ГА на параллельных вычислительных системах. В работе [8] было предложено два варианта модификаций:

1. На i -м этапе получения потомков с помощью дробления на $m/2$ процессов основного потока выполнения каждый процесс проводит скрещивание двух особей, затем при вычислении функции приспособленности каждый узел вычислительной системы отыскивает значения F_i (функция приспособленности на i -м этапе) для порождённых на нем особей (файлов данных).
2. Параллельное развитие нескольких популяций на узлах распределенной вычислительной системы с последующим выбором наилучшего результата.

Второй вариант является приемлемым в случае легко вычисляемых целевых функций,

когда потери, возникающие при пересылке данных в конвергентной коммуникационной среде распределенной вычислительной системы, могут быть сравнимы со временем вычисления функции приспособленности F . Также одновременная эволюция нескольких популяций может компенсировать возможное схождение популяции к локальному экстремуму[8].

В развитии первой модификации в [9] было предложено осуществлять скрещивание и вычисление значений функции приспособленности, а, следовательно, и целевой функции, одновременно на нескольких машинах (клиентская часть алгоритма) и проверять условие сходимости на выделенной машине (серверная часть алгоритма).

Второй вариант параллельной реализации ГА предполагает одновременный запуск алгоритма на p вычислительных узлах. По окончании работы алгоритма на всех рабочих узлах параллельной ВС производится сбор данных о полученных значениях экстремума и выбор наибольшего значения функции приспособленности F [6].

Заключение

В статье предложена базовая концептуальная модель эффективного размещения файлов в узлах вычислительной системы. В развитии этого направления были рассмотрены четыре алгоритма оптимального размещения файлов, базирующихся на разных подходах. Это позволило автору сформировать концепцию для последующей разработки программной среды диверсного моделирования процессов размещения файлов в конвергентных вычислительных средах.

Библиографический список

1. Ботыгин И.А., Попов В.Н. Архитектура распределенной файловой системы // Интернет-журнал «НАУКОВЕДЕНИЕ» 2014. № 6 <http://naukovedenie.ru/PDF/137TVN614.pdf> (доступ свободный). Загл. с экрана. Яз. рус., англ. DOI: 10.15862/137TVN614
2. Зинкин С.А., Белецкий П.А. Оптимизация размещения данных по узлам информационно-вычислительной сети [Текст] // Современные тенденции технических наук: материалы II Междунар. науч. конф. (г. Уфа, май 2013 г.). – Уфа: Лето, 2013. – С. 29–31.
3. Шевченко В.И. Оптимизация управления распределением файлов в корпоративной информационной сети / Е.Н. Машенко, В.И. Шевченко // Оптимизация производственных процессов: Сб. научн. тр. – Севастополь: Изд-во СевНТУ, 2010. – Вып. 12 – С.131–135.
4. Шевченко В.И. Динамическое размещение таблиц данных при информационном обслуживании в критических системах / А.В. Цуканов, В.И. Шевченко // Научно технический журнал «Радиоэлектронные и компьютерные системы» – Харьков: изд-во НАУ «ХАИ», 2012. – Вып. 6 (58) – С. 23–27.
5. Штовба С.Д. Муравьиные алгоритмы / Научно-практический журнал ExponentaPro. –2003. – №4. С. 70 – 75.
6. Едемская Е.Н., Бельков Д.В. Муравьиный метод распределения файлов в компьютерной сети // Актуальные направления научных исследований XXI века: теория и практика – г. Воронеж: 2015. – С. 124–128.
7. Панченко Т. В. Генетические алгоритмы: учеб.- метод. пособие / под ред. Ю. Ю. Тарасевича / Аст- рахан. ун-т. Астрахань, 2007. – 87 с.
8. Eroftiev A. A., Timofeeva N. E., Savin A. N. Parallel Computing in Application to Global Optimization Problem Solving // Grid and Visualization Systems : MIPRO, 2011 Proc. of the 34th Intern. Convention. Zagreb, Croatia : DENONA, 2011. – P. 185–190.
9. Савин А.Н. , Дружинин И.В., Ерофтиев А.А. Применение генетических алгоритмов для решения задач оптимизации на параллельных и распределенных вычислительных системах // Известия саратовского университета. Новая серия. Серия: Математика. Механика. Информатика. – г. Саратов: 2013. – С. 99–109.

УДК 004.021

М.Д. Сафонов, магистрант**Научный руководитель: М.В. Привалов, канд. техн. наук, доц.***Донецкий национальный технический университет***АЛГОРИТМЫ И ИНСТРУМЕНТАЛЬНЫЕ СРЕДСТВА ПОСТРОЕНИЯ И ВИЗУАЛИЗАЦИИ 3D-ПОВЕРХНОСТЕЙ ЧЕЛЮСТНО-ЛИЦЕВОЙ ОБЛАСТИ ПО КОМПЬЮТЕРНЫМ ТОМОГРАММАМ**

Аннотация: в работе рассмотрена задача построения трехмерной модели челюстно-лицевой области (ЧЛО) на основании результатов обследования пациентов с помощью компьютерной томографии (КТ), проведен анализ подготовки данных для построения модели, был использован алгоритм марширующих кубов для построения модели на реальных данных.

Ключевые слова: трехмерная модель, поверхность, изображение, срезы, марширующие кубы

Annotation: article deals with the task related to buildup of 3D model of the maxillofacial skeleton based on the results of the survey using computed tomography, performed analysis of the preparation of data for the construction of the model, the marching cubes algorithm was used to build a model based on real data.

Key words: three-dimensional model, surface, image, slices, marching cubes

Введение

Компьютерные технологии приобретают особую роль в медицине и в частности в челюстно-лицевой хирургии (ЧЛХ). В основе этих технологий лежит компьютерная томография (КТ), которая не только является одним из важнейших методов диагностики в медицине, но и служит основой для построения трехмерных (3D) представлений исследуемых объектов.

Важной задачей в челюстно-лицевой хирургии является построение 3D модели челюстно-лицевой области (ЧЛО) для наглядного выделения области поражения, определения её параметров и построения материальной модели предоперационного шаблона.

Постановка задачи

Результаты обследования пациентов с помощью КТ записываются в DICOM (Digital Imaging and Communications in Medicine) файлы. Зачастую эти файлы представляют собой дискретные срезы (двумерные изображения). Данный формат является стандартом создания, хранения, передачи и визуализации медицинских изображений и документов обследованных пациентов. DICOM файл представляет собой сложную структуру данных, которая включающей в себя информацию о пациенте, данные изображения, данные о системе координат, связанной с изображением, данные об оборудовании, на котором проводилось исследование, описание проведенного исследования и т.д.

В зависимости от оборудования, DICOM файлы могут быть не сжатыми или сжатыми, иметь различную фотометрическую интерпретацию (MONOCHROME1, MONOCHROME2, RGB, CMYK, или YBR), иметь разное количество бит для хранения оттенков (обычно 8 или 16 бит, однако существуют сканеры, которые сохраняют в 12 или 32 битах) [1].

Таким образом, для построения 3D модели на основании массива DICOM снимков, не достаточно просто считать данные изображения. Необходимо правильно интерпретировать входную информацию и преобразовать её для последующего построения трехмерной модели.

Подготовка данных

Основой для отображения DICOM файлов, является информация о плотности тканей. Для отображения изображения необходимо значениям плотности сопоставить цвет. Это делает передаточная функция. Передаточные функции можно разделить по типу на абсолютные и относительные. Абсолютная передаточная функция строится для всех возможных плотностей, в то время как относительная передаточная функция строится на основе так называемого окна, которое указывает какой именно диапазон плотностей нужно отображать. Для КТ абсолютная передаточная функция - это шкала Хаунсфилда.

Шкала Хаунсфилда — это количественная шкала рентгеновской плотности (радиоденсивности)[2]. Шкала единиц Хаунсфилда (денситометрических показателей, англ. HU) — это шкала линейного ослабления излучения по отношению к дистиллированной воде, рентгеновская плотность которой была принята за 0 HU (при стандартных давлении и температуре). Величина HU для материала X с линейным коэффициентом ослабления μ_X вычисляется по следующей формуле:

$$\frac{\mu_X - \mu_{\text{water}}}{\mu_{\text{water}} - \mu_{\text{air}}} \times 1000, \quad (1)$$

где μ_{water} и μ_{air} — линейные коэффициенты ослабления для воды и воздуха при стандартных условиях. Данные стандарты были выбраны для практического применения в КТ живых организмов (в том числе человека), т. к. их анатомические структуры в значительной степени состоят из связанной воды.

Для веществ и органов, которые обычно встречаются в человеческом теле, вычислены следующие средние денситометрические показатели.

Таблица 1 Значения величины Хаунсфилда для тканей в человеческом теле.

Вещество	HU
Воздух	-1000
Легкие	-700
Мягкие ткани	с -300 по -100
Жир	-84
Вода	0
Кровь	с +30 по +45
Мышца	+40
Кость	с +700 (губчатая кость) по +3000 (плотная кость)

Для восстановления значения величины Хаунсфилда для данного пикселя используется следующая формула:

$$HU = \text{Pixel Intensity} * \text{Rescale Slope} + \text{Rescale Intercept}. \quad (2)$$

Pixel Intensity — это значение интенсивности пикселя. Параметры Rescale Intercept и Rescale Slope находятся в метаданных DICOM файлов по адресу (0028,1052) и (0028,1053) соответственно. Значения этих параметров зависят от устройства, которым были сняты изображения. Чаще всего они имеют следующие значения Rescale Slope = 1, Rescale Intercept = -1024. Данная формула позволяет отфильтровать изображения, удалив все пиксели, значения которых не попадают в интересующий нас диапазон. Таким образом, с помощью шкалы Хаунсфилда можно выделить все пиксели на которых отображены кости [3,4].

Построение трехмерной модели

Основой для построения поверхности является создание воксельной модели и триангуляция. Целью триангуляции является создание планарного графа по заданному набору координатных точек, который впоследствии разбивал бы объект исследования на треугольники и внешнюю поверхность. Существует много различных методов для выполнения триангуляции 3D поверхности, каждый из которых имеет свои положительные и отрицательные свойства применительно к работе с медицинскими изображениями. На основании анализа методов построения трехмерной модели в статье [5], для построения трехмерной модели был выбран алгоритм марширующих кубов. Этот алгоритм можно разбить на два этапа:

1. Разбиение области G пространства R^3 на конечное множество ячеек, поиск ячеек пересекаемых искомой поверхностью.

2. Аппроксимация поверхности в найденных ячейках.

Основными проблемами первого этапа являются:

1. Разбиение области G на ячейки.

2. Выбор ячеек, которые пересекаются с искомой поверхностью.

После разбиения области G на ячейки, значение функции, задающей поверхность, можно получить только в вершинах этих ячеек. При разбиении области, необходимо принимать во внимание, что возможны большие потери точности в тех случаях, когда, размер ячейки будет велик. В этом случае некоторые части поверхности не будут видны. Однако стоит отметить, что сильное уменьшение размера ячейки может сильно сказаться на скорости построения полигональной поверхности (зависит от используемого оборудования).

Задачей второго этапа является аппроксимация поверхности в каждой ячейке. Триангуляция является наиболее оптимальным способом аппроксимации. Имея способ триангуляции, можно уже аппроксимировать поверхность. К этому моменту уже известно количество треугольников и ребра ячеек, на которых лежат его вершины. Необходимо найти точку на ребре ячейки, в которой поверхность ее пересекает. В случае заданной таблично функции искомая точка находится с помощью линейной интерполяции двух вершин [6,7].

При построении трехмерной модели алгоритмом марширующих кубов использовалось исследование, состоящее из массива 448 DICOM файлов с размером среза 672 на 672 пикселя. Алгоритм марширующих кубов был реализован на языке программирования Java с использованием OpenGL GLSL для трехмерного рендеринга. Результатом работы алгоритма стала модель, состоящая из ≈ 5 млн. треугольников. Каждый треугольник представляет собой массив из трех точек, которые, в свою очередь, являются массивом из трех вещественных чисел представляющих координаты x , y , z пикселя.



Рис 1. Структурная схема построения 3D модели по данным DICOM файлов.

Для рендеринга на языке GLSL были реализованы вершинный и фрагментный шейдеры. Вершинный шейдер оперирует данными, связанными с вершинами треугольников, а также используется для видового и перспективного преобразования вершин. Каждая вершина умножается на матрицу MVP (матрицы Модели, Вида и Проекции), благодаря чему возможно перемещать, поворачивать и масштабировать модель. Фрагментный шейдер работает с фрагментами растрового изображения и определяет его цвет.

Массив координат вершин треугольников передается в память графического процессора (GPU) с помощью Vertex Buffer Object (VBO). VBO обеспечивает методы выгрузки данных в видеоустройство для не оперативного режима рендеринга. Это дает существенный прирост производительности, т.к. данные находятся в памяти видеоустройства, а не в оперативной памяти и поэтому она может быть отрендерена непосредственно видеоустройством. На рисунке 1 представлена укрупненная структурная схема процесса построения 3D модели по данным DICOM файлов. Полученная трехмерная модель представлена на рисунке 2.

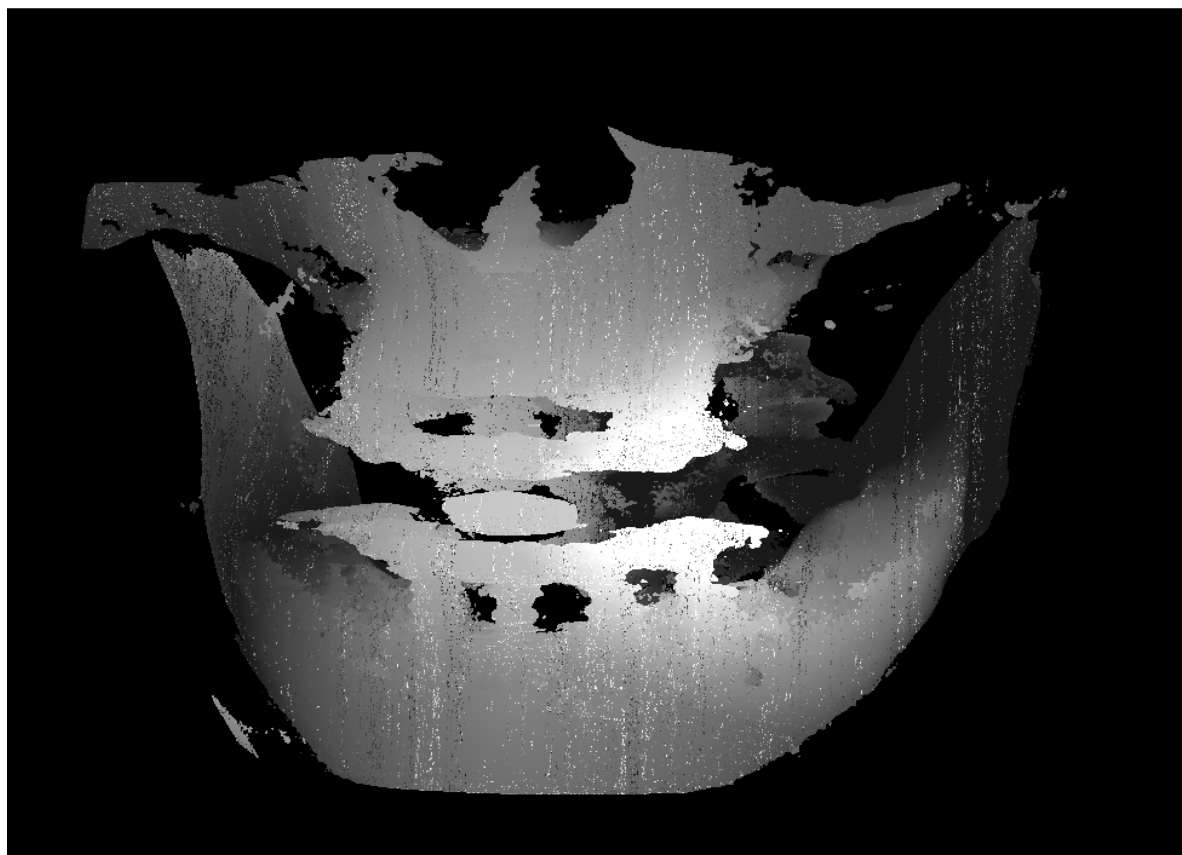


Рис 2. Трехмерная модель полученная с помощью алгоритма марширующих кубов.

Выводы

Таким образом, можно сказать, что метод марширующих кубов хорошо подходит для построения трехмерных моделей ЧЛО по данным исследований КТ. Однако, следует отметить, что результаты работы данного алгоритма могут иметь некоторые топологические неточности и генерировать очень большое количество треугольников, что в свою очередь влияет на быстродействие алгоритма. Использование параллельных и распределенных вычислений, а также методов пост-обработки поверхности, позволит решить эти проблемы [6].

Направлением дальнейших исследований является пост-обработка трехмерной модели с целью оптимизация полученной поверхности, определение и выделение области поражения, а также построение модели имплантата.

Библиографический список

1. DICOM [Электронный ресурс] // Digital Imaging and Communications in Medicine — (<http://medical.nema.org/dicom>).
2. Шкала Хаунсфилда // Википедия. [2013—2013]. Дата обновления: 14.03.2013. URL: <http://ru.wikipedia.org/?oldid=53583434> (дата обращения: 14.03.2013).
3. С. Бояровски, А. Петров. Построение трёхмерной геометрической модели по набору МРТ снимков. Санкт-Петербургский государственный университет.
4. Караваев А. С., Копысов С. П., Кузьмин И. М. Построение биомеханических конечно-элементных моделей. Труды ИМ УрО РАН «Проблемы механики и материаловедения», 2016, Ижевск.
5. М. Д. Сафонов, Д. Д. Новиков, М.В. Привалов. Методы построения поверхности по изображениям КТ для создания модели челюстно-лицевого скелета методом 3D печати.
6. А.С. Резниченко, М.В. Привалов Повышение производительности методов реконструкции поверхности за счет применения параллельных вычислений. Інформаційні управляючі системи та комп'ютерний моніторинг (ІУСКМ–2013)/Матеріали IV міжнародної науково-технічної конференції студентів, аспірантів та молодих вчених. - Донецьк, ДонНТУ — 2013, Том 1.
7. Бугров Н.В., Голубев В.И., Клименко А.С., Дижевский А.Ю., Обоймов А.С., Какауридзе Д.Г., Фролов П.В. Обзор алгоритмов триангуляции неявно заданной поверхности, MEDIAS2012 Труды Международной научной конференции, 07-14 мая 2012 г., Лимассол, Республика Кипр, Изд.ИФТИ, С.151-173, ISBN 978-5-88835-023-2

УДК 681.3

А.И. Чалый, студент 4-го курса

Научный руководитель: М.В. Заморёнов, канд. техн. наук, доц.

Севастопольский государственный университет

СИСТЕМА АВТОМАТИЗАЦИИ ПОСТРОЕНИЯ МАТЕМАТИЧЕСКИХ МОДЕЛЕЙ

Аннотация: В данной работе приводится описание автоматизированной системы построения математических моделей. Описывается алгоритм функционирования данной системы, основанный на известных из литературы методе путей и методе обхода графов в глубину. Также приводится пример работы программной системы автоматизации построения математических моделей.

Ключевые слова: математическая модель, система, автоматизация, граф.

Abstract: This article describes the automated system for building mathematical models. Describes the algorithm of this system, which is based on known from the literature methods, such as method of ways and the method of graph traversal in depth. Also contains an example of the functioning of software system for automated building mathematical models.

Keywords: mathematical model, system, automation, graph.

В современном мире достаточно актуальной является проблема прогнозирования качества и надёжности функционирования разрабатываемых систем. Как правило, подобные задачи решаются с помощью различных методов моделирования. Моделирование можно рассматривать как замещение исследуемого объекта (оригинала) его уменьшенным либо упрощённым по структуре аналогом, обеспечивающим близкое к оригиналу поведение системы в рамках некоторых допущений и приемлемых погрешностей. Моделирование выполняется с целью познания свойств оригинала путем исследования его модели, так как создание модели, как правило, несравнимо дешевле и проще создания оригинальной системы.

Одним из наиболее точных методов моделирования является математическое моделирование [1, 2]. Математическое моделирование общественных, экономических, биологических и физических явлений, объектов, систем и различных устройств является одним из важнейших средств познания природы и проектирования самых разнообразных систем и устройств. Целью математического моделирования является анализ реальных процессов математическими методами. В свою очередь, это требует формализации процесса математического моделирования, подлежащего исследованию. Модель может представлять собой математическое выражение, содержащее переменные, поведение которых аналогично поведению реальной системы. Однако, вручную рассчитывать все параметры и характеристики математических моделей зачастую бывает чрезмерно трудоёмко. Особенно в случаях, когда в системе присутствуют аспекты, влияющие на систему с определённой вероятностью. С развитием современных технологий стала доступна автоматизация процесса математического моделирования, а так же повышение точности расчётов, в том числе за счёт возможности проведения определённого количества симуляций модели и сбора статистических данных о её работе.

Целью данной работы является процесс разработки автоматизированного построения математических моделей разрабатываемых систем. На данном этапе разработки предполагается, что сведения о разрабатываемой системе будут указаны как входные данные и должны быть заранее вручную определены пользователем, а именно – количество состояний, в которых может находиться система, времена пребывания системы в каждом состоянии и вероятности переходов между ними. Результаты расчётов программы будут содержать информацию о частоте нахождения системы в каждом из состояний и общее время работы либо простоя системы.

Алгоритм работы программы основан на методе путей [3] и заключается в следующем – пользователь вводит количество различных состояний системы, переходы и вероятности переходов между состояниями. При этом каждое состояние модели должно соответствовать определённому состоянию реальной системы – рабочий этап, подготовительный этап, состояния поломки какого-либо модуля системы, состояние починки, и подобные. Таким

образом, из состояний будет составлен граф с начальной и конечной вершинами, а также множеством переходов между ними.

На первом этапе работы программы будут определены все возможные пути, включая все различные циклы, которые будут найдены с помощью метода обхода графов в глубину [4]. Обход в глубину (поиск в глубину, англ. Depth-First Search, DFS) — один из основных методов обхода графа, часто используемый для проверки связности, поиска цикла и компонент сильной связности и для топологической сортировки. Общая идея алгоритма состоит в следующем: для каждой не пройденной вершины необходимо найти все не пройденные смежные вершины и повторить поиск для них.

Но для поиска циклов в графе недостаточно отметить вершину как «посещённую» или «не посещённую». Поэтому в процессе алгоритма вершинам задают некоторые цвета:

- если вершина белая, значит, мы в ней еще не были, вершина не пройдена;
- серая – вершина проходится в текущей процедуре;
- черная – вершина пройдена, все итерации от нее завершены.

На втором этапе – для каждого построенного пути будет рассчитано время нахождения в каждом из состояний, после чего программа определит общее время нахождения модели в рабочих состояниях и в состояниях простоя.

На третьем этапе будет определено среднее время для всех случаев (путей), а также определены лучший (с наименьшим временем простоя) и худший (с наименьшим временем работы) варианты путей и их рассчитанные вероятности появления.

На рис. 1 и 2 представлены примеры графов состояний системы без цикла и с циклом.

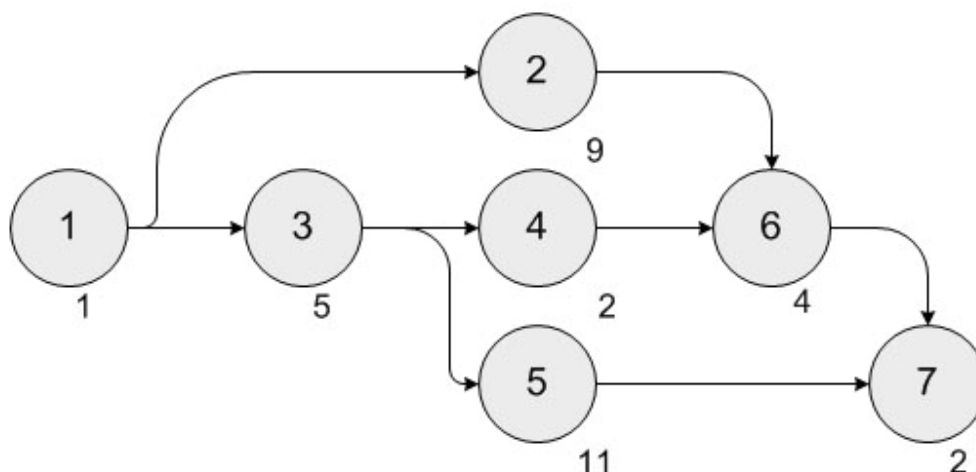


Рис. 1. Граф состояний системы без цикла

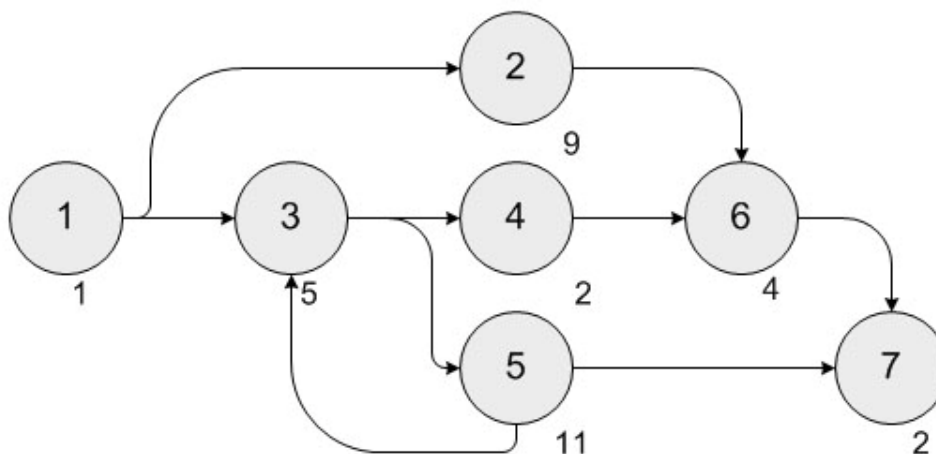


Рис. 2. Граф состояний системы с циклом

Примеры работы программы для систем представленных на рис. 1 и 2 представлены на рис. 3 и 4.

```

Текущее дерево:
Вершина: 1; время вершины: 1
Родители вершины: 0 4 6
Дочерние вершины: 2 3
Вершина: 2; время вершины: 9
Родители вершины: 1
Дочерние вершины: 6
Вершина: 3; время вершины: 5
Родители вершины: 1
Дочерние вершины: 4 5
Вершина: 4; время вершины: 2
Родители вершины: 3
Дочерние вершины: 6
Вершина: 5; время вершины: 11
Родители вершины: 3
Дочерние вершины: 7
Вершина: 6; время вершины: 4
Родители вершины: 2
Дочерние вершины: 7
Вершина: 7; время вершины: 2
Родители вершины: 5
Дочерние вершины:
//////////
Введите номер желаемого действия:
1)Добавить новую вершину
2)Вывод списка вершин текущего дерева
3)Вывод всех путей обхода текущего дерева без циклов
4)Поиск циклов текущего дерева методом обхода в глубину
0)Выход
3
Возможные пути от начальной вершины до конечных:
-1--2--6--7- Время пути: 16.
-1--3--4--6--7- Время пути: 18.
-1--3--5--7- Время пути: 16.
//////////
Введите номер желаемого действия:
1)Добавить новую вершину
2)Вывод списка вершин текущего дерева
3)Вывод всех путей обхода текущего дерева без циклов
4)Поиск циклов текущего дерева методом обхода в глубину
0)Выход
4
//////////
Введите номер желаемого действия:
1)Добавить новую вершину
2)Вывод списка вершин текущего дерева
3)Вывод всех путей обхода текущего дерева без циклов
4)Поиск циклов текущего дерева методом обхода в глубину
0)Выход

```

Рис. 3. Пример работы программы для системы, не содержащей циклы

Таким образом, данный программный продукт предоставляет возможность более точно определить вероятности нахождения разрабатываемой системы в работоспособном состоянии и в состоянии простоя за меньшее время, чем это рассчитывалось бы вручную. Это позволит получать результат на более ранних этапах разработки системы, а значит, и прогнозировать расходы на обслуживание системы также раньше.

```

Текущее дерево:
Вершина: 1; время вершины: 1
Родители вершины: 0
Дочерние вершины: 2 3

Вершина: 2; время вершины: 9
Родители вершины: 1
Дочерние вершины: 6

Вершина: 3; время вершины: 5
Родители вершины: 1 5
Дочерние вершины: 4 5

Вершина: 4; время вершины: 2
Родители вершины: 3
Дочерние вершины: 6

Вершина: 5; время вершины: 11
Родители вершины: 3
Дочерние вершины: 3 7

Вершина: 6; время вершины: 4
Родители вершины: 2 4
Дочерние вершины: 7

Вершина: 7; время вершины: 2
Родители вершины: 5 6
Дочерние вершины:
//////////
Введите номер желаемого действия:
1)Добавить новую вершину
2)Вывод списка вершин текущего дерева
3)Вывод всех путей обхода текущего дерева без циклов
4)Поиск циклов текущего дерева методом обхода в глубину
0)Выход
4
Цикл на вершине 3! (-3--5--3-)
//////////
Введите номер желаемого действия:
1)Добавить новую вершину
2)Вывод списка вершин текущего дерева
3)Вывод всех путей обхода текущего дерева без циклов
4)Поиск циклов текущего дерева методом обхода в глубину
0)Выход
3
Возможные пути от начальной вершины до конечных:
-1--2--6--7- Время пути: 16.
-1--3--4--6--7- Время пути: 18.
-1--3--5--7- Время пути: 16.
//////////

```

Рис. 4. Пример работы программы для системы, содержащей циклы

Библиографический список:

1. Математическое моделирование – Википедия - [https://ru.wikipedia.org/wiki/Математическая модель](https://ru.wikipedia.org/wiki/Математическая_модель).
2. Основные понятия математического моделирования - <http://www.studfiles.ru/preview/6339322/>
3. Моделирование процесса функционирования обслуживающего устройства с обесценивающими отказами методом путей М.В. Заморёнов, В.Я. Копп, Д.В. Заморёнова, Ю.Л. Явкун // Известия ТулГУ. Технические науки. Вып. 4. Тула: Изд-во ТулГУ, 2016. С. 225–236.
4. Использование обхода в глубину для поиска цикла – Викиконспекты - [http://neerc.ifmo.ru/wiki/index.php?title=Использование обхода в глубину для поиска цикла](http://neerc.ifmo.ru/wiki/index.php?title=Использование_обхода_в_глубину_для_поиска_цикла)

СЕКЦИЯ 2

Робототехника. Технологии искусственного интеллекта



УДК 004.032

А.А. Архипова, студентка 1-го курса магистратуры**Научный руководитель: А.Е. Безуглая, канд. техн. наук, доц.***Севастопольский государственный университет***ПРОГРАММНО-АППАРАТНЫЙ КОМПЛЕКС СИСТЕМЫ ПОДДЕРЖАНИЯ ЖИЗНЕДЕЯТЕЛЬНОСТИ РАСТЕНИЙ КАК СРЕДСТВО ОБУЧЕНИЯ РОБОТОТЕХНИКЕ**

Аннотация: в данной статье рассмотрено создание программно-аппаратного комплекса системы поддержания жизнедеятельности растений на базе микроконтроллера и ПО для обучения школьников робототехнике. Рассмотрены методы, которые должны быть применены в работе. Выявлены и обоснованы подсистемы, включенные в систему. Кроме того, описан общий процесс функционирования разрабатываемой системы и ожидаемые результаты от ее применения.

Ключевые слова: обучение робототехнике, обеспечение жизнедеятельности, микроконтроллер, датчики, обучающие программы

Annotation: in this article, the creation of a software and hardware system for the maintenance of plants on the basis of a microcontroller and software for teaching students of robotics is considered. The methods that should be applied in the work are considered. Identified and justified subsystems included in the system. In addition, the general process of functioning of the system being developed and the expected results from its application are described.

Key words: training in robotics, life support, microcontroller, sensors, training programs.

Современная система школьного образования не дает возможности подросткам обучаться робототехнике дома или в школе, в то время как создание роботов сегодня – одна из самых перспективных и интересных для детей областей развития науки и техники. В связи с этим возникает необходимость включить в образовательный процесс прикладную робототехнику, как средство развития инженерного мышления уже на момент выпуска из школы.

Одной из областей применения робототехники, достаточно простой и наглядной для осознания подростками, является биология, а именно ботаника. Считаем, что синтез робототехники и ботаники позволяет как изучить основы программирования, так и процесс жизнедеятельности растения и влияние на него изменения условий роста, а также создать конкурентные условия и провести соревнования среди учеников.

Системы автоматизированного поддержания жизни растений разрабатываются, однако, не в образовательном аспекте, а в промышленных масштабах, например тепличные системы. Однако изучение таких систем без специального образования затруднительно. Мы же предлагаем построение коробочной версии, которая могла бы быть собрана пользователем «с нуля» без какого-либо багажа знаний в данной сфере, что актуально для подростков, а главное соответствует системо-деятельностному подходу, применяемому в современной школе.

Основная цель функционирования подобных разработок – развитие у детей и молодежи навыков сборки и программирования замкнутых систем, использующих микроконтроллеры, а также предоставление возможности наблюдения и исследования работы полученной системы.

Для реализации предложенного подхода используются такие методы: синтез, анализ, индукция, абстрагирование, моделирование, эксперимент и другие.

Основные этапы реализации проекта в школах

1. Выбор и покупка оборудования для сбора системы.
2. Сборка прототипа системы (собрать и проверить на адекватность работы первый прототип, найти недочеты и исправить ошибки для того, чтобы система, которая будет собрана с учениками, была собрана верно).
3. Создание коробочной версии совместно с учениками (создание коробочной версии с учетом недостатков, выявленных при создании первого прототипа, совместно с учениками).
4. Проведение соревнований среди учеников на наибольшее время автоматического обслуживания растения.

Для начала реализации проекта в школе необходимо достаточное техническое обеспечение, чтобы каждый ученик был подготовлен по дисциплине «Информатика» как минимум на среднем уровне, изучение основ программирования в школе должно начинаться в первые годы обучения в школе, а в 6 м классе ученики уже должны иметь общие представления по ботанике. Именно в это время можно начинать реализацию предлагаемого проекта [1].

Рассмотрим общую структуру коробочной системы, которая будет собрана и запрограммирована со школьниками.

Система состоит из подсистемы контроля и подсистемы реагирования (рис. 1).



Рис. 1 Структурная схема системы

Подсистема контроля представлена датчиками, такими как датчик освещенности, влажности, температуры, количества оставшейся воды в колбе.

Подсистема реагирования состоит из погружной помпы, обеспечивающей полив при необходимости, лампы, повышающей освещенность, кулера, обеспечивающего обдув, и прочих элементов реагирования на состояние растения.

Программирование собранной системы проводится на языке C, который является одним из основных и широко используемых языков программирования. Изучение данного языка поможет школьникам развить системное мышление [2].

В качестве микроконтроллера наиболее целесообразно использовать плату Arduino mini, по ряду причин. Первой и наиболее весомой из них является то, что Arduino – готовый продукт, который имеет в своем составе регулятор питания, микроконтроллер, программатор, интерфейсы для подключения устройств и программные библиотеки. Нет необходимости думать о программировании микроконтроллера или способах подключения периферии.

Также детям будет легко понять как всё работает, поскольку обучение можно построить по принципу от простого к сложному. Еще одним большим преимуществом Arduino является библиотека примеров, идущая в комплекте.

В целом платформа предназначена для быстрого прототипирования и обучения, что непосредственно отвечает целям проекта.

Опишем общий процесс функционирования системы.

В качестве исходных данных выступает вводимая пользователем со специальной клавиатуры информация о параметрах растения. Система в зависимости от них устанавливается на некоторое фиксированное число поливов, температуру и освещенность в

определенное время, например в сутки.

В качестве альтернативы пользователь может выставлять только такие качественные показатели, как: «крайне сухо», «средняя влажность» и т.п., по которым система устанавливает состояние растения.

Состояние среды изменяется в случае, если показатели состояния растения не совпадают с идеальными, заданными пользователем в начале использования системы. Изменение состояния среды достигается путем подключения подсистемы реагирования.

Ожидаемые результаты от внедрения проекта:

- закрепить практические навыки программирования на языке С у школьников,
- практически закрепить приобретенные теоретические знания в области ботаники и жизнедеятельности растений,
- развитие инженерного мышления у школьников,
- наработать конструкторские навыки (а также дизайнерские навыки) у учеников,
- приобщить учеников к научным исследованиям,
- дать учащимся возможность проявить себя в сборке робототехнических систем,
- повысить уровень заинтересованности робототехникой и ботаникой среди молодежи,
- создать обучающий курс робототехники для школ и иных образовательных учреждений.

Библиографический список

1. Примерные программы по учебным предметам. Информатика и ИКТ. 5 – 11 классы. – 3-е изд., перераб. – М. : Просвещение, 2013. – 64с. – (Стандарты второго поколения).
2. Язык программирования Си.\Пер. с англ., 3-е изд., испр. — СПб.: "Невский Диалект", 2001. - 352 с: ил.

УДК 004.932

М.А. Казанцев, магистрант**Научный руководитель: М.В. Привалов, канд. техн. наук, доц.***Донецкий национальный технический университет***ПАРАЛЛЕЛЬНАЯ СЕГМЕНТАЦИЯ ТРЕХМЕРНЫХ ИЗОБРАЖЕНИЙ НА ОСНОВЕ АЛГОРИТМА РАЗБИЕНИЯ ГРАФА С ПРИМЕНЕНИЕМ РОЕВОГО ИНТЕЛЛЕКТА**

Аннотация: приводится описание метода параллельной сегментации трехмерных изображений на основе алгоритма разбиения графа с применением роевого интеллекта. В основе метода лежит представление изображения в виде графа, где вершины – это пиксели, а длина ребер между вершинами определяется схожестью пикселей. Реализация алгоритма направлена на повышение производительности сегментации за счет применения методов роевого интеллекта колонии медоносных пчел, распараллеливаемых на основе открытого стандарта программирования графических ускорителей OpenCL.

Ключевые слова: сегментация, разбиение графа, алгоритм пчелиной колонии, распараллеливание, OpenCL.

Abstract: article describes the parallel 3D image segmentation algorithm based on the swarm intelligence implementation of the graph cut approach. The idea of this method lays in the image representation as a graph, where pixels define graph's vertices with the edges' weights dependent on a pixels similarity. Implementation of the algorithm is directed towards an increase of the segmentation performance by means of the bee colony swarm intelligence methods parallelization with the help of the open GPU programming standard OpenCL.

Keywords: segmentation, graph cut, artificial bee colony algorithm, parallelization, OpenCL.

Введение. В современном мире бурно развиваются технологии неинвазивной, а также малоинвазивной компьютерной медицинской диагностики, большинство из которых основано на получении медицинских изображений. Благодаря этому детализация оцифрованной визуальной информации постоянно растёт. Вместе с тем, рост уровня детализации приводит к увеличению объема и, соответственно, трудоемкости обработки информации такого типа. В ряде диагностических методик оперативность обработки и представления информации существенна, что делает задачу повышения быстродействия методов обработки медицинских изображений актуальной.

Параллельно с медицинской аппаратурой развиваются и становятся более производительными современные вычислительные системы, в особенности, системы, основанные на GPGPU вычислениях и массивно-параллельных вычислениях с применением специализированных ускорителей. Таким образом, решение задачи повышения быстродействия методов обработки медицинских изображений актуально и возможно за счёт применения алгоритмов, хорошо масштабируемых на современных параллельных вычислительных системах.

1. Анализ существующих методов решения. В настоящее время существует значительное количество методов сегментации, которые показали свою эффективность, однако, применяемые в них алгоритмы сегментации разрабатывались в условиях, когда необходимость работы с большими объемами данных отсутствовала, и, как следствие, они значительно теряют в эффективности и быстродействии при увеличении объема входной информации. Это делает актуальным создание таких средств сегментации, которые хорошо бы себя показали в работе с более «тяжелыми» изображениями, а также в лучшей степени реализовали технические и программные возможности современных вычислительных платформ. Спиральная томография (СКТ) и конусно-лучевая компьютерная томография (КЛТ) являются распространёнными методами диагностики, поэтому прикладной задачей, на примере которой возможно проводить исследования подходов к распараллеливанию методов обработки изображений, может стать сегментация 3D СКТ изображений.

Основой для разработки соответствующего заданным требованиям параллельного метода сегментации могут стать метаэвристики, которые хорошо себя показали при решении оптимизационных, комбинаторных, а также других видов задач [1]. Среди свойств эвристических алгоритмов следует отметить недетерминированность, а также пространственный поиск субоптимального решения, что в итоге позволит получить наиболее приемлемый результат исходя из найденных вариантов.

Среди разновидностей эвристических алгоритмов особого внимания заслуживают роевые биологические метаэвристики. В общем случае, роевой алгоритм – это модель поведения, для которой характерна коллективность, мультиагентность и децентрализованность. Взаимодействие между агентами системы приводит к определению общей стратегии поведения, следование которой сделает возможным нахождение решения той или иной задачи [2].

Для роевых алгоритмов характерна природная предрасположенность к параллелизму, а также сохранение более высокого уровня производительности (в сравнении с другими разновидностями алгоритмов) при решении «больших» задач. Такая особенность хорошо сочетается с аппаратной и программной архитектурой современных высокопроизводительных ускорителей вычислений, работа которых подразумевает параллельное выполнение. Это позволит эффективно реализовать данные методы.

Ориентируясь на современные тенденции развития в области графических технологий, наиболее приоритетной задачей является поиск алгоритма, который бы сохранил приемлемый уровень производительности при значительном увеличении объема задачи. Как показывают исследования, алгоритм пчелиной колонии соответствует вышеуказанным требованиям.

Ключевые принципы алгоритма пчелиной колонии [3]:

1. Результаты работы пчел-разведчиков определяют наиболее перспективные места для поиска нектара.
2. Соответственно уровню их «перспективности», на каждый участок отправляется некоторое количество пчел-рабочих.
3. Рабочие собирают нектар, попутно уточняя информацию разведчиков о количестве нектара в некотором радиусе от изначального участка

Выполненные ранее исследования применения алгоритма пчелиных колоний на комбинаторной задаче разбиения полного связного графа размерностью 100 вершин показали эффективность этого алгоритма по сравнению с итерационными и эволюционными вариантами [4]. В качестве критерия выступала минимизация суммарного количества связей между подграфами. Сравнение результатов эксперимента, при размерности графа в 100 вершин, приведено в таблице 1. Сравнение результатов показало эффективность роевых алгоритмов.

Таблица 1 – Результаты сравнения эффективности разбиения графа

№	Алгоритм	Значение целевой функции
1.	Итерационный	822
2.	Эволюционный	804
3.	Генетический	765
4.	Муравьиный	694
5.	Пчелиный	690

Следует отметить, что при увеличении размерности графа до 1000 вершин, разрыв в пользу алгоритма пчелиных колоний увеличивается [4].

Анализ существующих методов кластеризации и сегментации изображений показал, что задача сегментации может быть сведена к задаче разбиения графа. Это позволяет применить к её решению роевые алгоритмы и, в частности, алгоритм пчелиной колонии.

2. Сегментация изображения методом разбиения графа. В основе метода лежит представление изображения в виде графа, вершинами которого являются пиксели, при этом длина ребер между вершинами определяется схожестью пикселей на основе их характеристик, в соответствии с которыми осуществляется сегментация [5]. Вес (длина) ребра рассчитывается по формуле (1):

$$W(V_i, V_j) = |I(P_i) - I(P_j)|, \quad (1)$$

где V_i, V_j – i-я и j-я вершины соответственно, $I(P_i)$ – интенсивность пикселя P_i .

На начальном этапе выполнения алгоритма происходит сравнение пикселей изображения с 4 соседними (северный, южный, западный, восточный) [5]. После этого, будет получено большое количество разрозненных сегментов, которые, по возможности, следует объединить. Критерием объединения отдельных сегментов является выполнение условия (2):

$$Diff(C_1, C_2) < MInt(C_1, C_2), \quad (2)$$

где $Diff(C_1, C_2)$ – ребро минимальной длины между двумя сравниваемыми сегментами, $MInt(C_1, C_2)$ – меньший из максимальных перепадов интенсивностей внутри рассматриваемых сегментов. Схематически объединение сегментов может быть представлено следующим образом (рис. 1):

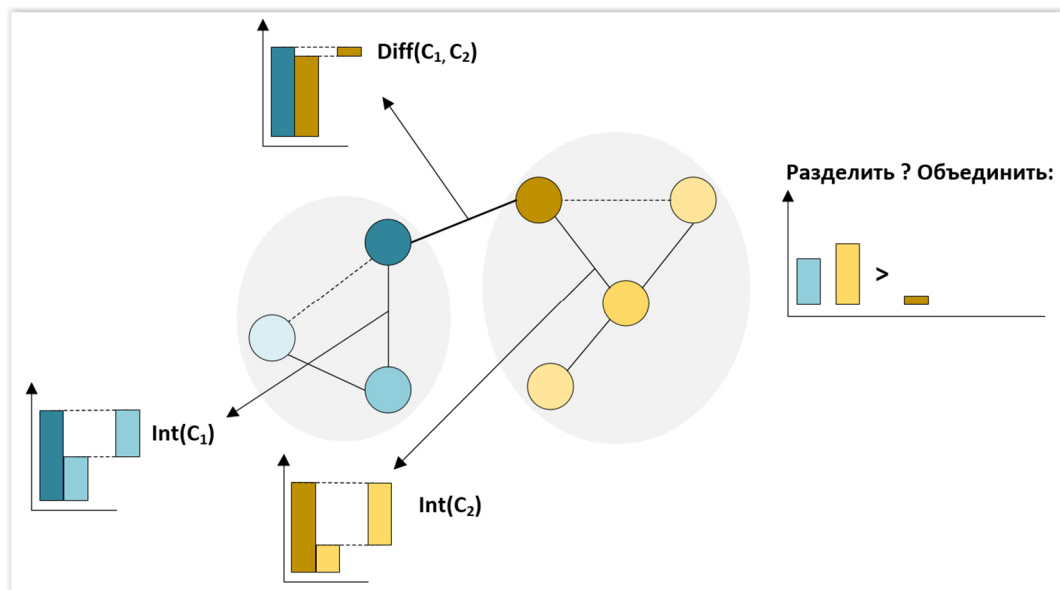


Рис. 1. Объединение сегментов

Существующие методы сегментации изображений условно можно разделить на две основные группы: высокопроизводительные, а также те методы, которые показывают высокий уровень качества конечного результата. Согласно этой классификации, сегментация методом разделения графа относится к первой группе. Несмотря на это, достоинствами данного метода являются простота модификации и естественный параллелизм, что даёт возможность реализовать его с применением параллельных вычислений.

Другой особенностью такого подхода является хорошая масштабируемость алгоритма разбиения графа. Благодаря этому, при наличии достаточно мощного аппаратного обеспечения есть возможность повысить качество конечного результата путем увеличения количества ребер в графе. На начальном этапе алгоритма сегментации будет происходить сравнение пикселей с 8 соседними вместо 4 (добавляются северо-западный, северо-восточный, юго-западный, юго-восточный пиксели). Такие изменения повлекут за собой значительное увеличение вычислительной нагрузки, однако, вместе с тем возрастет качество конечного результата. Помимо этого, исследования показали наличие возможности модификации способа расчета длины ребер путем добавления в формулу не только спектральных, но и пространственных характеристик пикселей [5], что даёт в результате (3):

$$dist(P_i, P_j) = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2 + (r_i - r_j)^2 + (g_i - g_j)^2 + (b_i - b_j)^2}, \quad (3)$$

где x_i , y_i и z_i – координаты пикселя, r_i , g_i , b_i – его показатели по каждой из цветовых составляющих модели RGB. Такой способ расчета позволит более точно определять на изображении разрозненные участки, которые относятся к одному объекту.

3. Сегментация изображения методом разбиения графа с использованием алгоритма пчелиной колонии. Опираясь на формализованные понятия в решении оптимизационных задач с использованием методов роевого интеллекта пчелиной колонии [3], следует более подробно рассмотреть применение этих методов к задаче сегментации изображений.

В данном случае понятие «пчелы» исходит из специфики архитектуры современных ускорителей вычислений, которая представляет из себя совокупность аппаратных и программных элементов, способных выполнять различные задачи параллельно, тем самым значительно сокращая временные затраты. Например, в рамках применения массивно-параллельных вычислений на видео- и специализированных ускорителях пчелой может считаться одна нить вычислений или рабочий элемент. Среди существующих платформ распараллеливания графических и массивно-параллельных вычислений платформа OpenCL [6] основана на универсальном и открытом стандарте, в связи с чем в данной работе предлагается адаптировать параллельный алгоритм именно к реализации на ней.

На начальном этапе необходимо распределить множество пикселей изображений на некоторое количество блоков. Принадлежность пикселя тому или иному блоку определяется показателем его яркости. Значение целевой функции для каждого блока зависит от количества пикселей изображения, которые принадлежат этому блоку. Пропорционально значению целевой функции, для блока пикселей определяется некоторое количество пчел, общее число которых пропорционально количеству блоков.

Такой подход позволит оптимизировать распределение нагрузки на вычислительные элементы путем оценки размера сегментов и их количества на изображениях.

Число пчел для каждого блока пикселей определяется по формуле (4) [3]:

$$b_i = B \times \frac{f(x_i) - \min(F(X))}{\max(F(X)) - \min(F(X))}, \quad (4)$$

где b_i – количество пчел для текущего блока, B – общее количество пчел, $f(x_i)$ – значение целевой функции для текущего блока, $\min(F(X))$ и $\max(F(X))$ – минимальное и максимальное значение целевой функции среди всех блоков соответственно.

На основании полученных значений предлагается производить распределение мощностей вычислительных элементов аппаратной базы по блокам пикселей для дальнейшей параллельной сегментации изображения методом разбиения графа.

Определение сегментов начинается с произвольных точек на изображении, число которых зависит от размера блока пикселей, в котором они расположены.

Для каждого блока пикселей следует определить некоторое количество рабочих групп вычислительных элементов [6], которое зависит от размера блока и допустимого размера рабочей группы на данном ускорителе вычислений. Пиксели блока равномерно распределяются между рабочими группами, с занесением данных о пикселях в локальную память групп.

В процессе сегментации каждым рабочим элементом в память группы заносятся данные о ребре с максимальной длиной, а определенные к какому-либо сегменту пиксели получают маркировку. После того как рабочая группа заканчивает сегментацию своей части пиксельного блока, данные о ребрах с максимальной длиной, а также данные о маркировке пикселей записываются в глобальную память. После окончания сегментации всеми рабочими группами внутри какого-либо блока должна выполняться попытка объединения сегментов между участками блока.

Процесс сегментации изображения описанным методом схематически изображен на рис. 2(а). Такой алгоритм соответствует программной архитектуре платформы параллельных вычислений согласно открытому стандарту OpenCL (рис. 2(б)).

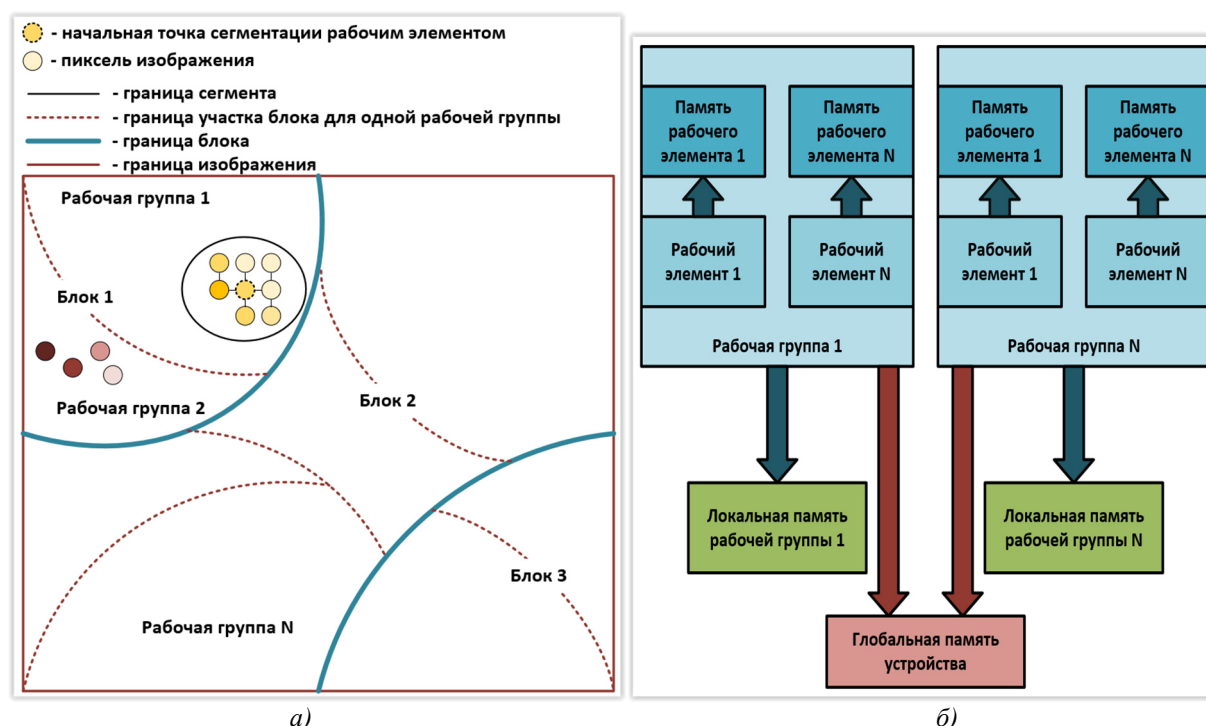


Рис. 2. Алгоритм и платформа а) Схематическое представление параллельной сегментации изображения методом разбиения графа с использованием алгоритма пчелиной колонии; б) Программная архитектура OpenCL

Заключение. Предлагаемый метод сегментации изображений на основании разбиения графа с помощью роевого алгоритма пчелиных колоний может быть использован для сегментации изображений, раскрывая возможности и особенности программной и аппаратной архитектуры доступных в настоящее время ускорителей вычислений. Теоретические исследования показывают, что такой метод обладает естественным параллелизмом и хорошей масштабируемостью, что позволит снизить сложность работы с изображениями высокого уровня детализации и эффективно выполнять сегментацию медицинских изображений, сформированных современной аппаратурой.

Особенностью реализации данного метода с использованием современных технологий вычислений на ускорителях является необходимость динамически менять назначение рабочих групп при больших размерах блоков пикселей. Таким образом, программная реализация представленного метода и экспериментальная проверка эффективности его работы при различных входных данных является направлением дальнейших исследований.

Библиографический список

1. Метаэвристики: монография / Ю.А. Скобцов, Е.Е. Федоров. – Донецк: Изд-во «Ноулидж» (Донецкое отделение), 2013 – 426 с.
2. Курс «Эволюционные вычисления», лек.11 «Роевые и муравьиные алгоритмы» (Д.В. Сперанский, Ю.А. Скобцов; НОУ «ИНТУИТ»), 2014.
3. Роевой алгоритм в задачах оптимизации (В.В. Курейчик, Д.Ю. Запорожец; «Известия ЮФУ. Технические науки»), 2010.
4. Использование пчелиных алгоритмов для решения комбинаторных задач (В.М. Курейчик, А.А. Кажаров; Технологический институт Южного федерального университета, г. Таганрог, Россия; persianland1987@gmail.com, kur@tsure.ru), 2010.
5. Efficient Graph-Based Image Segmentation (by Pedro F. Felzenszwalb (MIT) and Daniel P. Huttenlocher; Cornell University), 2004.
6. An Introduction to the OpenCL Programming Model (by J. Thompson and K. Schlachter; NYU: Media Research Lab; thompson@cims.nyu.edu, ks228@cs.nyu.edu), 2012.

СЕКЦИЯ 3
Интеллектуальный анализ данных.
Технологии Big Data. Системы
поддержки принятия решений

УДК 004.415:519.816

А.А. Вилинский, студент 4-го курса

Научный руководитель: Е.В. Козлова, канд. техн. наук, доц.

Севастопольский государственный университет

ПРОГРАММНАЯ СИСТЕМА ПОДДЕРЖКИ ПРИНЯТИЯ РЕШЕНИЙ О ВЫБОРЕ БРЕНД-ПРОЕКТА

Аннотация: Рассматривается один из практических подходов к решению задачи обоснованного выбора варианта бренд-проекта для заданной совокупности критериев. Предлагается структурная схема программной системы поддержки принятия решения, исследуются возможности практического использования данного программно-вычислительного комплекса.

Ключевые слова: система поддержки принятия решений, бренд-проект, программно-вычислительный комплекс, нечеткие множества.

Annotation: One of practical approaches to the decision of the task of a reasonable choice of option a brand project for the given set of criteria is considered. The skeleton diagram of program system of support of decision-making is offered, possibilities of practical use of this program computer system are researched.

Key words: decision support system, DSS, brand-project, fuzzy sets, software-hardware complex.

Появление сети Интернет, а также доступных персональных компьютеров на рубеже 20-21 века стало большим шагом к всеобщей глобализации. Современному человеку постоянно приходится принимать множество решений в профессиональной сфере и обыденной жизни. Неверно принятое решение может повлечь не только экономические, но и экологические, человеческие потери. Именно это привело к массовому внедрению систем поддержки принятия решений (СППР, англ. DSS – Decision Support System) на всех предприятиях малого, среднего и крупного бизнеса, в государственных учреждениях и корпорациях, коммерческих предприятиях, медицинских учреждениях, страховых компаниях[1,2].

В период глобализации на рынке все больше появляется крупных компаний и конгломератов, которые уже давно вышли за рамки границ одного государства и являются транснациональными. Статистика показывает, что, не смотря на такое увеличение количества значимых брендов, большинство проектов обречены на провал ещё на стадии планирования. Только на территории Российской Федерации, где создание брендов набирает всё большую популярность, треть всех проектов терпит неудачу на стадиях реализации[1].

Это связано, в частности, с тем, что лица, принимающие решения (ЛПР), зачастую пользуются методиками расчета денежных инвестиций, который конкретный проект может принести на определённом интервале времени. Однако финансовый рынок зачастую не стабилен и даже государство уже не может выступать гарантом надёжности своей валюты. Поэтому следует подойти к данному вопросу с другой точки зрения: мы должны не делать попытки предсказать прибыльности проекта, а выбрать наилучший, тот, который укрепитесь в головах у людей по тем или иным причинам.

Бренд представляет собой ассоциативный набор образов – это и логотип, и цвета, и музыкальное сопровождение: всё то, что поможет человеку отличить данную продукцию или услугу от любого другого аналога[4]. Исходя только из этого можно понять, что большую долю в успехе выбора бренд-проекта будет играть субъективизм. Проработанность каждого проекта, детализация, оформление, запоминаемость, скорость выхода на рынок – пример совокупности критериев, по которым следует оценивать проекты. Однако просто оценив каждый проект дифференциальным методом, невозможно добиться приемлемых результатов, потому что люди не совершают выбор, не имея альтернатив. Это является главным постулатом о принятии решения. Поэтому целесообразно использовать парные сравнения по каждому критерию.

Мощным математическим аппаратом для этого является теория нечётких множеств. Использование нечёткой логики было предложено Лотфи Заде ещё в 1965 году, и по сей день является одним из наиболее часто используемых методов многокритериального анализа[1].

Идея использования нечётких множеств следующая: специалист предметной области должен оценить попарно проекты по каждому критерию. Оценку обычно выносят по девятибалльной шкале Саати [3,5]:

- 1 – нет преимущества;
- 3 – слабое преимущество;
- 5 – существенное преимущество;
- 7 – явное преимущество;
- 9 – абсолютное преимущество;
- 2,4,6,8 – промежуточные оценки.

На основе данных оценок строится матрица:

$$A = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1k} \\ a_{21} & a_{22} & \dots & a_{2k} \\ \dots & \dots & \dots & \dots \\ a_{k1} & a_{k2} & \dots & a_{kk} \end{pmatrix},$$

где a_{ij} – преимущество проекта P_i перед проектом P_j , ($i, j = \overline{1, k}$), определённое ЛПР по девятибалльной шкале.

На этом работа ЛПР по организации подготовки исходных данных заканчивается, и СППР производит вычисления степени принадлежности каждого элемента P_j нечеткому множеству G . При этом, чем ближе степень принадлежности к единице, тем лучше[3]. На выходе специалист получает список возможных решений и уже должен сам сделать решение о выборе бренд-проекта.

Отличительной чертой данного метода является использование принципа Беллмана-Заде: выбор альтернативы, максимально удовлетворяющей одновременно всем критериям. Это обусловлено тем, что в наше время нельзя создать хороший бренд-проект, если хотя бы один из параметров сильно уступает другим проектам, другими словами – недостаток по одному критерию нельзя восполнить избытком по другому[1].

Структурная схема программного комплекса для решения такой задачи показана на рисунке 1.

Актуальными проблемами для данной системы, как и для СППР в целом, является субъективизм лица, принимающего решения. Более того, от его опыта зависит качество результата работы СППР, так как наиболее точно оценить преимущества одного проекта над другим на стадии ввода матрицы является задачей, не подконтрольной самой СППР, а следовательно, отражает оценочные суждения руководителя.

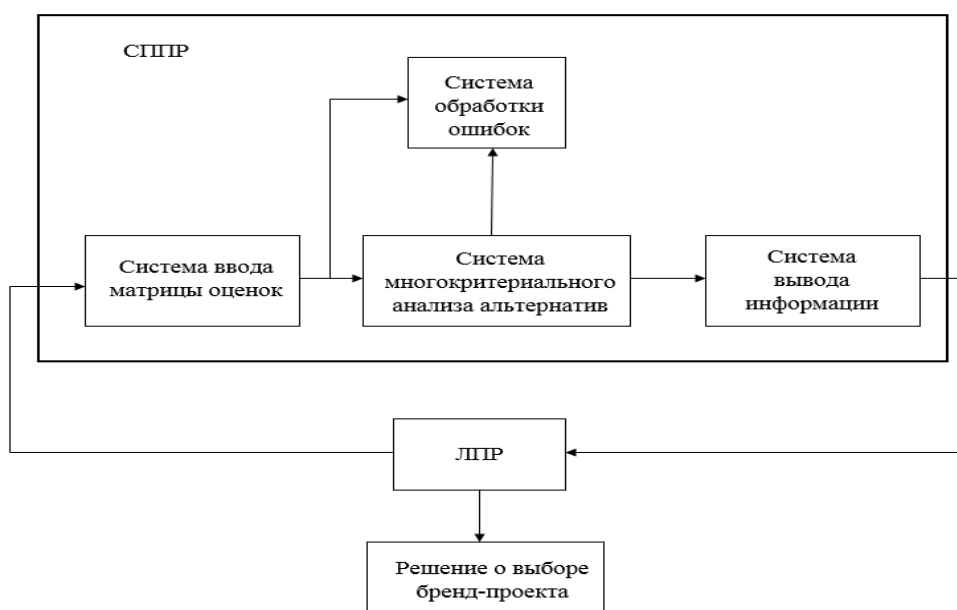


Рис. 1. Структурная схема программного комплекса

Возможно, с развитием нейронных цепей, ученым удастся научить СППР производить первоначальные субъективные оценки в автоматическом режиме, однако это уже будет система продвинутого искусственного интеллекта.

В рамках дальнейших исследований предполагается проведение оценки эффективности применения предлагаемого комплекса для решения практических задач в различных областях экономики.

Библиографический список

1. Келлер К. Л. Стратегический бренд-менеджмент: создание, оценка и управление марочным капиталом / К. Л. Келлер — М.: «Вильямс», 2005. — 704 с.
2. Ротштейн А. П. Многокритериальный выбор брендопроекта с помощью нечетких парных сравнений альтернатив / А. П. Ротштейн, С. Д. Штовба, Е. В. Штовба // Управление проектами и программами. — 2006. — Т. 2 — С. 138–146.
3. Саати Т. Л. Взаимодействие в иерархических системах / Т. Л. Саати // Техническая кибернетика. — 1979. — №1. — С. 68–84.
4. Трахтенгерц Э. А. Компьютерные системы поддержки принятия управленческих решений / Э. А. Трахтенгерц // Проблемы управления. — 2003. — Вып. 1. — С. 13-28.
5. Zopounidis, C. Fuzzy Sets in Management, Economics and Marketing. World Scientific / C. Zopounidis, P. M. Pardalos, G. Baourakis. — New Jersey : World Scientific, 2001. — 284 p.

УДК 004.6

Р.Ю. Измайлов, магистрант**Научный руководитель: Е.Н. Машенко, к.т.н., доц.***Севастопольский государственный университет***КОНЦЕПЦИЯ ОТКРЫТЫХ ДАННЫХ***Аннотация: в данной статье представляется концепция открытых данных, призванная решить проблемы неструктурированности и недоступности данных из-за авторских прав.**Ключевые слова: анализ данных, открытые данные.**Annotation: this article presents a concept of open data, designed to solve the problems of non-structured and inaccessible data due to copyright.**Key words: data analysis, open data.***Анализ данных и его проблемы**

Интеллектуальный анализ данных (data mining) - процесс обнаружения в больших объемах данных ранее неизвестных, практически полезных и доступных интерпретации знаний, необходимых для принятия решений в различных сферах человеческой деятельности [1]. Обычно результатом анализа являются некоторые агрегированные данные, на основе которых делаются прогнозы, выводы о корреляции тех или иных параметров. Анализом данных занимаются ученые, частные лица, различные организации.

Ярким примером автоматического анализа данных является сервис Sentdex. Сервис Sentdex был разработан частным лицом. Sentdex предоставляет результаты машинного анализа данных настроения людей по всему миру касательно различных тем (политика, финансы, география) [2]. На сервисе используются автоматические алгоритмы анализа текстов, в частности Natural Language Processing для анализа настроений письменного языка.

На рис. 1 приведены данные сервиса по анализу политических тем.

	Topic	7-day Volume of Mentions	Sentiment	Sentiment Rising or Falling
	War	25601	good	▼
	Government Budget	12551	good	▲
	Health Care	8290	good	▲
	Environment	7831	good	▼
	National Security Agency (N.S.A.)	7235	good	▼
	Obamacare	6661	good	▼
	Economy	6484	good	▲
	Crime	6278	poor	▼

Рис. 1. Результат машинного анализа данных сервиса Sentdex.com [1]

Сервис анализирует следующие источники: CNN, Fox, USA Today, ABC, MSNBC, CNBC, CBS, Huffington Post, Yahoo Politics, Washington Post, и Reuters [2]. Информация структурирована в виде таблицы. Первый столбец таблицы - некоторая политическая тема, второй столбец - общее число упоминаний за 7 дней, третий столбец - настроение с которым тема упоминается. В четвертом столбце показан индикатор падения или роста настроения.

Известны следующие проблемы в сфере анализа данных:

- большинство данных являются неструктурированными, что усложняет их автоматический сбор и анализ;
 - авторское право значительно ограничивает объемы данных доступные для анализа.
- Эти проблемы решает концепция открытых данных.

Концепция открытых данных

Открытые данные (англ. *open data*) — концепция, отражающая идею о том, что определённые данные должны быть свободно доступны для машиночитаемого использования и дальнейшей републикации без ограничений авторского права, патентов и других механизмов контроля [3].

Основные принципы открытых данных [4]:

- первичность;
- полнота;
- актуальность;
- пригодность к машинной обработке;
- отсутствие дискриминации по доступу;
- отсутствие проприетарных форматов;
- лицензионная чистота.

Открытые государственные данные – это информация (в том числе документированная), созданная в пределах своих полномочий государственными органами, либо поступившая в указанные органы и организации, а также информационно-аналитическими организациями, участвующими в публикации собственных открытых данных на территории Российской Федерации, которая подлежит размещению в сети Интернет в формате, обеспечивающем ее автоматическую обработку в целях повторного использования без предварительного изменения человеком (машиночитаемый формат), и может свободно использоваться в любых соответствующих закону целях любыми лицами независимо от формы ее размещения (простая совокупность сведений, база данных и т.д.) [5].

Для того чтобы данные были открытыми необходимо:

- их структурировать;
- распространять по свободным лицензиям (Creative Commons, GNU GPL).

Открытые данные распространяются в машиночитаемых форматах, то есть форматах, обеспечивающих обеспечивающем их автоматическую обработку данных для повторного использования без предварительного изменения человеком. Примеры таких форматов: CSV, XML, JSON, ZIP, GZ, XLS, XLSX, RDF.

Источники открытых данных

Источником открытых данных может стать любая организация (государственная или частная), любое частное лицо. Открытые данные распространяются по распределенной сети Интернет. Принято, что если у организации есть представительство в виде сайта, то открытые данные следует расположить по адресу (URL): адрес сайта + /open-data/. Например, sentedex.com/open-data/.

Открытые данные в России

Благодаря правительственной инициативе в России разработан и внедрен единый портал открытых данных [5]. На портале данные разбиты по категориям (рис. 2). Кроме того, на портале публикуются новости и законодательные акты, касающиеся темы открытых данных.

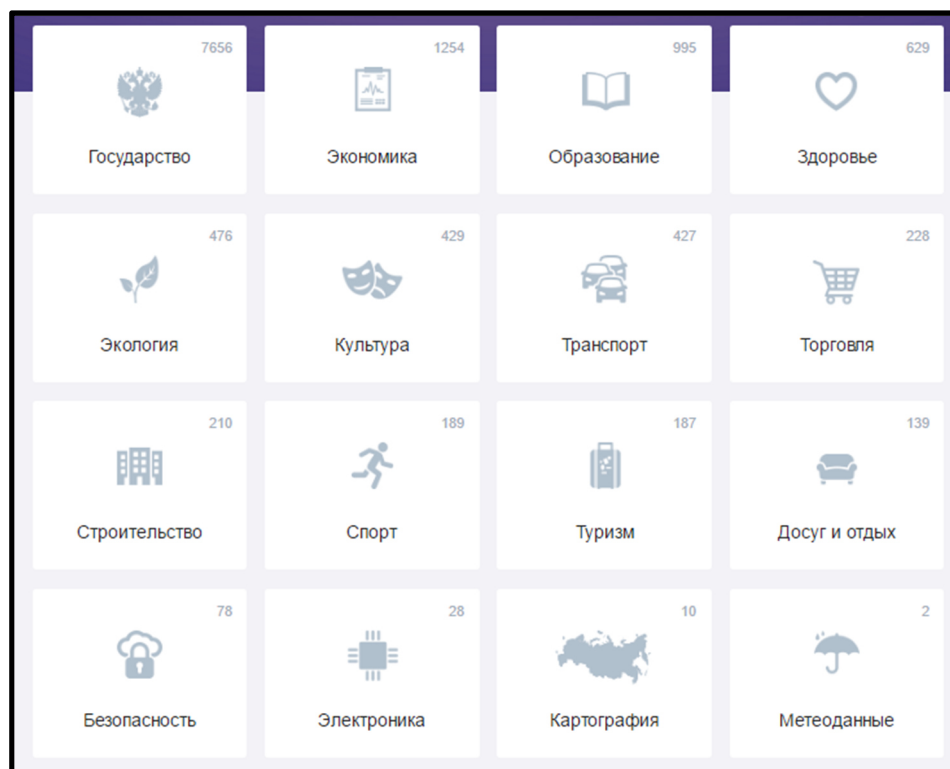


Рис. 2. Категории открытых данных [5]

Выводы

Концепция открытых данных призвана снять ограничения и открыть доступ к данным. Это ускорит темпы научного и технического прогресса. Принятие этой концепции повсеместно откроет безграничные возможности для исследований во всех областях человеческой деятельности.

Библиографический список

1. И.А. Чубукова. Data mining. Интуит, 2006. - 279 с.
2. Sentdex - Quantifying the Qualitative [Электронный ресурс]. Режим доступа - <http://sentdex.com>. Дата доступа - 04.03.17.
3. Открытые данные [Электронный ресурс]. Режим доступа - https://ru.wikipedia.org/wiki/Открытые_данные. Дата доступа - 04.03.17.
4. Правительство России. Открытые данные. [Электронный ресурс] - Режим доступа - http://government.ru/open_data/. Дата доступа - 04.03.17.
5. Портал открытых данных Российской федерации [Электронный ресурс] - Режим доступа - <http://data.gov.ru>. Дата доступа - 04.03.17.

УДК 621.391

Ю.Э. Криворучко, магистрант**Научный руководитель: В.С. Чернега, канд. техн. наук, доц.***Севастопольской государственной университет***МОДЕЛИРОВАНИЕ ПРОБЛЕМНО-ОРИЕНТИРОВАННЫХ ТЕКСТОВЫХ СООБЩЕНИЙ**

Аннотация: в статье описывается построение модели текстовых сообщений банковской тематики, которая в дальнейшем будет использоваться в качестве модели структуры в модифицированном алгоритме сжатия данных.

Ключевые слова: частота, моделирование, сжатие, банковский текст, суффиксное дерево.

Annotation: the article described the construction of a model text messages banking theme, which will be used as a model structure in the modified data compression algorithm.

Key words: frequency, modeling, compression, bank text, suffix tree.

Большинство информационных систем выполняют сбор данных для дальнейшей передачи либо хранения. Обработка текстовых данных выполняется для обеспечения последующего сжатия данных с целью эффективного хранения больших объемов, а также кодирования данных для выполнения последующей передачи по каналу связи. С точки зрения увеличения степени сжатия и уменьшения времени сжатия данных, эффективнее выполнять сжатие с помощью предметно-ориентированных компрессоров [4]. В таких компрессорах должна быть организована структура данных в соответствии с предметной областью. Для формирования структуры данных для системы сжатия банковских сообщений, необходимо выполнить моделирование текстов банковской тематики. В статье [3] описан системный подход моделирования текста, в котором учитывается два уровня порядка: внешний - в виде иерархической структуры и внутренний, соответствующий единству образов источников. Полученная модель может быть использована для идентификации авторства текста. Перед моделью банковского текста не стоит задача идентификации, поэтому при моделировании будет использоваться только второй уровень порядка.

Цель работы является моделирование сообщений банковской тематики для дальнейшего использования модели структуры в специализированном архиваторе, который ориентирован на сжатие банковских текстовых сообщений.

В данной работе в качестве проблемно-ориентированных текстовых сообщений будем рассматривать текстовые сообщения банковской тематики. Моделирование выполнялось на основе построения суффиксного дерева типового банковского договора. Для анализа полученной модели выполнен подсчет статистической вероятности появления словосочетаний, состоящих из трех слов, в текстах банковской тематики. В предшествующей работе [2] был выполнен анализ статистических характеристик текстовых банковских сообщений русского языка. Были рассмотрены следующие лексические единицы: символы, n-граммы (биграммы, триграммы), слова. Исследование вероятности появления словосочетаний в банковских сообщениях русского языка выполнялось по аналогичной технологии. Результаты исследования приведены в таблице.

Исследование выполнялось на основе банковских текстов, в частности типовых договоров, используемых в банковской сфере. Общий объем текста, который использовался для подсчета статистических вероятностей составил более 250 тысяч символов. В таблице 1 представлены результаты исследования наиболее часто встречаемых в текстах банковской тематики словосочетаний, состоящих из трех слов.

В результате исследований выявлено что в банковских договорах наиболее часто встречаемыми словосочетаниями являются «именуемый в дальнейшем», «действующего на основании», «лице действующего на», «в лице действующего». Из последних трех словосочетаний, можно составить связное по смыслу словосочетание в «в лице действующего на основании». Похожая тенденция наблюдается по всей приведенной таблице, например, «стороны именуемые в» и «именуемые в дальнейшем». Поэтому, можно сделать вывод о том,

что можно выполнять анализ статистической характеристики по словосочетаниям, состоящих из большого количества слов. Данные, приведенные в таблице 1, позволят ускорить поиск совпадающих строк в таблице кодирования.

Таблица 1 – Частота появления словосочетаний в текстах банковской тематики

Словосочетание	Частота появления	Словосочетание	Частота появления
именуемый в дальнейшем	0,00337	Стороны заключили настоящий	0,00147
действующего на основании	0,00318	стороны именуемые в	0,00147
лице действующего на	0,00291	именуемые в дальнейшем	0,00147
в лице действующего	0,00287	Договор о нижеследующем	0,00147
Почтовый адрес Телефон/факс	0,0028	другой стороны именуемые	0,00144
основании именуемый в	0,00268	договор в дальнейшем	0,00144
на основании именуемый	0,00268	настоящий договор в	0,00144
адрес Почтовый адрес	0,00257	РЕКВИЗИТЫ И ПОДПИСИ	0,0014
адрес Телефон/факс ИНН/КПП	0,00257	в том числе	0,00129
Юридический адрес Почтовый	0,00257	за каждый день	0,00125
в соответствии с	0,0025	каждый день просрочки	0,00125
Корреспондентский счет БИК	0,0025	по настоящему договору	0,00117
Телефон/факс ИНН/КПП Расчетный	0,0025	дальнейшем Банк с	0,00106
счет Банк Корреспондентский	0,0025	Банк с одной	0,00106
Банк Корреспондентский счет	0,0025	нижеследующем ПРЕДМЕТ ДОГОВОРА	0,00091
Расчетный счет Банк	0,0025	о нижеследующем ПРЕДМЕТ	0,00091
ИНН/КПП Расчетный счет	0,0025	вступает в силу	0,00076
счет БИК Подпись	0,00246	в сумме рублей	0,00076
с одной стороны	0,00159	по настоящему договору	0,00117
одной стороны и	0,00155	обязательств по настоящему	0,00064
с другой стороны	0,00151	соответствии с действующим	0,00057
в дальнейшем Договор	0,00147	средств на счете	0,00053
дальнейшем Стороны заключили	0,00147	денежных средств на	0,00053
дальнейшем Договор о	0,00147	за пользование кредитом	0,00053
в дальнейшем Стороны	0,00147	с действующим законодательством	0,00053
заключили настоящий договор	0,00147	неустойку в размере	0,00053

Кроме этого дополнительное сжатие может быть обеспечено за счет знания модели текста банковской тематики. Для ее получения было выполнено моделирование. При моделировании текста банковской тематики выполнялось с помощью алгоритма построения суффиксного дерева [1, 5]. Входные данные, в терминах алгоритма – это строка, представляющая собой тексты типовых банковских договоров. В исследовании для построения модели в качестве входных данных использовался договор банковского вклада.

Договор содержит 3,5 тысяч символов. При моделировании текста в качестве лексической единицы выбрано слово. Построение начинается с пустого дерева. Первый добавляемый суффикс является самым длинным и в точности соответствует входным данным. Следующий суффикс для рассмотрения получается путем удаления первого слова в предыдущем суффиксе. Каждый суффикс добавляется от корня дерева. Построение дерева заканчивается после добавления последнего слова из входных данных. Полученное дерево будет в точности отражать структуру текста.

Полученные результаты будут использоваться для улучшения алгоритма сжатия в разрабатываемой системе сжатия банковских данных.

Библиографический список

1. Гасфилд Д. Строки, деревья и последовательности в алгоритмах: Информатика и вычислительная биология / Пер. с англ. И. В. Романовского. — 2-е изд. — СПб.: Невский Диалект, 2003. — 654 с.
2. Криворучко Ю.Э. Исследование статистических характеристик проблемно-ориентированных текстовых сообщений / Ю.Э. Криворучко // Мир компьютерных технологий: Сборник статей студенческой научно-технической конференции. – Севастополь: СевГУ, 2016. – С.79-82.
3. Мордвинов А.В. Системный подход в моделировании текста / А.В. Мордвинов // Вестник ННГУ. – Н. Новгород, 2010. – С.185-190.
4. Чернега В.С. Адаптивное кодирование текстовых сообщений в системах передачи данных / В.С. Чернега, А.В. Исаева // Вестник СевНТУ, Вып.58. – Севастополь, 2004. – С.15-22.
5. Breslauer D., Italiano G. Near real-time suffix tree construction via the fringe marked ancestor problem / D. Breslauer, G. Italiano // Journal of Discrete Algorithms, V.18, 2013. – P.32-48.

УДК 004.422.833

И.В. Смирнов, студент 4-го курса, А.В. Крячко, студент 2-го курса**Научный руководитель: И.В. Шевченко, канд. техн. наук, доц.***Севастопольский государственный университет***КОНЦЕПТУАЛЬНАЯ МОДЕЛЬ ОРГАНИЗАЦИИ ПРОЦЕССОВ ОБРАБОТКИ БОЛЬШИХ ДАННЫХ В ПРОЕКТЕ C2S НА ОСНОВЕ ТЕХНОЛОГИИ SPRING XD***Аннотация: В статье рассмотрена технология Spring XD для работы с Big Data в проекте C2S (социальная научная кооперация).**Ключевые слова: Spring XD, Big Data, Большие данные, Java, организация модели работы приложения.**Annotation: In the article is considering Spring technology for work with Big Data in the project C2S (Social scientists cooperation).**Key words: Spring XD, Big Data, Large data, Java, organization of the application operation model.*

Введение. Большие Данные, на сегодняшний момент, являются одним из ключевых драйверов развития информационных технологий. Это направление, относительно новое для бизнеса. Связано это с тем, что в эпоху информационных технологий, особенно после бума социальных сетей, по каждому пользователю интернета стало накапливаться значительное количество информации, что в конечном счете дало развитие направлению Big Data [1].

Термин «Большие Данные» вызывает множество споров, многие полагают, что он означает лишь объем накопленной информации, но не стоит забывать и о технической стороне, данное направление включает в себя технологии хранения, распределенные вычисления, а также сервисные услуги. Следует отметить, что к данной сфере относится обработка именно большого объема информации, который затруднительно обрабатывать традиционными способами. На рис. 1 представлена таблица, сравнивающая характеристики таблица традиционной базы и базы Больших Данных.

Характеристика	Традиционная база данных	База Больших Данных
Объем информации	От гигабайт (10^9 байт) до терабайт (10^{12} байт)	От петабайта (10^{15} байт) до эксабайт (10^{18} байт)
Способы хранения	Централизованный	Децентрализованный
Структурированность данных	Структурирована	Полуструктурирована и неструктурированная
Модель хранения и обработки данных	Вертикальная модель	Горизонтальная модель
Взаимосвязь данных	Сильная	Слабая

Рис. 1 – Сравнение традиционной базы данных и базы данных Больших Данных

Сфера Больших Данных характеризуется следующими признаками [2–4]:

Volume – объем, накопленная база данных представляет собой большой объем информации, который трудоемко обрабатывать и хранить традиционными способами, для них требуются новый подход и усовершенствованные инструменты.

Velocity – скорость, данный признак указывает как на увеличивающуюся скорость накопления данных (90% информации было собрано за последние 2 года), так и на скорость обработки данных, в последнее время стали более востребованы технологии обработки данных в реальном времени.

Variety – многообразие, т.е. возможность одновременной обработки структурированной и неструктурированной разноформатной информации. Главное отличие структурированной информации – это то, что она может быть классифицирована. Примером такой информации может служить информация о клиентских транзакциях. Неструктурированная информация включает в себя видео, аудио файлы, свободный текст, информацию, поступающую из социальных сетей. На сегодняшний день 80% информации входит в группу неструктурированной. Данная информация нуждается в комплексном анализе, чтобы сделать ее полезной для дальнейшей обработки.

Veracity – достоверность данных, все большее значение пользователи стали придавать значимость достоверности имеющихся данных. Так, у интернет-компаний есть проблема по разделению действий, проводимых роботом и человеком на сайте компании, что приводит в конечном счете к усложнению процессов анализа данных.

Value – ценность накопленной информации. Большие Данные должны быть полезны компании и приносить определенную ценность для нее. К примеру, помогать в усовершенствовании бизнес-процессов, составлении отчетности или оптимизации расходов.

При соблюдении указанных выше 5 условий, накопленные объемы данных можно относить к числу больших.

1. Постановка задачи. В рамках статьи предлагается концептуальная модель обработки больших данных в проекте социальной научной кооперации (C2S) на основе Spring XD (eXtreme Data) [5, 6]. Данная работа является продолжением исследований, изложенных в работах [7 – 9]. Целью работы является аналитический обзор технологий Spring XD и возможностей их применения при разработке проекта научной кооперации. В проекте C2S предусмотрено хранилище больших данных в котором будут храниться публикации, учебные пособия, научные данные. Учебные пособия, публикации являются структурированными данными и будут храниться в базах данных SQL, а научные данные являются неструктурированными данными и будут храниться в NoSQL базах данных.

2. Концептуальная модель и сервис-ориентированная среда для научной кооперации C2S

Структура рассматриваемой концепции среды сервис-ориентированной научной кооперации приведена на рис. 2.

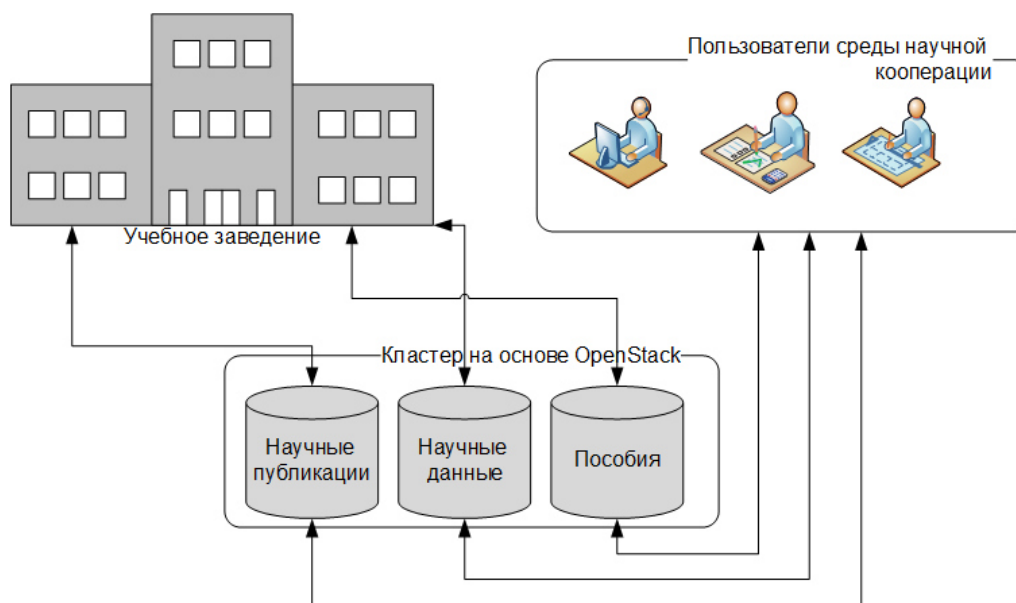


Рис. 2 – Модель сервис-ориентированной среды научной кооперации C2S

Модель научной кооперации представляет собой отдельный сайт, интернет-площадку для обмена научными артефактами. Рассмотрим особенности проекта Spring XD, применительно к процессам анализа Больших Данных и задачам научной кооперации.

Spring XD представляет собой единую, распределенную и масштабируемую систему приема аналитических данных в режиме реального времени, пакетной обработки и экспорта данных. Цель проекта заключается в упрощении разработки приложений, работающих с большим объемом данных.

Big Data приложения имеют много общих характеристик с приложениями бизнес-интеграции и пакетной обработки данных. Spring предлагает эффективные решения для построения приложений, такие как Spring Integration и Spring Batch проекты. Spring XD основана на этих проектах и предлагает легковесную среду выполнения, которая легко настраивается и собирается с помощью DSL-соединения (DSL- Digital Subscriber Line). Отличительные особенности экосистемы Spring XD:

1. Высокая пропускная способность распределения данных из различных источников в хранилища большого объема информации, таких как файловая система HDFS (Hadoop Distributed File System – файловая система, предназначенная для хранения файлов больших размеров, поблочно распределённых между узлами вычислительного кластера) и платформа операционного анализа Splunk.

2. Аналитика в режиме реального времени, например, сбор метрик и подсчет текущих значений контролируемых метрик качества ИТ-сервисов.

3. Управление рабочим процессом через пакетные задачи. Задачи сочетают в себе взаимодействие со стандартными бизнес-системами (например, RDBMS– Radiation Data Base Management System), а также Hadoop операциями (например, MapReduce, HDFS, Pig, Hive or Cascading).

4. Высокая пропускная способность передачи данных на экспорт, например, из HDFS в RDBMS или NoSQL СУБД.

Spring XD является частью большой системы Spring которая включает в себя большое количество модулей для работы с разными вычислительными системами и по различным сценариям (Рис. 3).

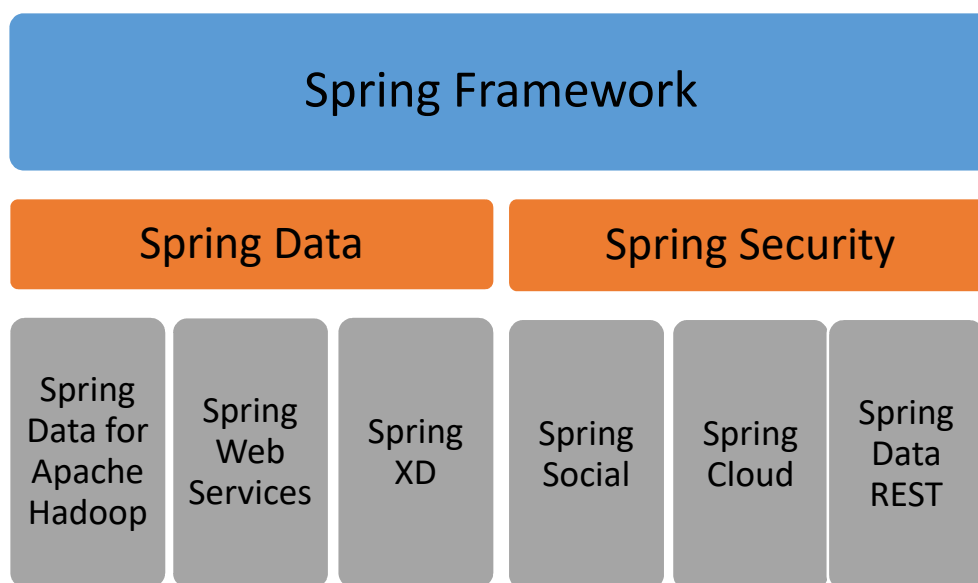


Рис. 3. – Общая иерархическая модель системы Spring

Spring Data – это проект который упрощает использование таких технологий доступа к данным, как реляционные и нереляционные СУБД, MapReduce фреймворки и облачные сервисы. Spring Data состоит из подпроектов для конкретных СУБД.

Spring Data for Apache Hadoop – это проект который упрощает использование программной платформы Apache Hadoop, предоставляя унифицированную модель конфигурирования и простое в использовании API для HDFS, MapReduce, Pig и Hive. Также обеспечивает интеграцию с такими проектами Spring, как Spring Integration и Spring Batch.

Spring Batch – легковесный, комплексный пакетный фреймворк, предназначенный для разработки надежного пакета приложений, необходимых для ежедневного использования бизнес-системой.

Spring Integration – модуль который расширяет модель программирования в Spring, благодаря поддержке Enterprise Integration Patterns. Spring Integration включает в себя легковесный обмен сообщениями между Spring-приложениями через событийно-управляемые адаптеры. Эти адаптеры обеспечивают высокоуровневую абстракцию над возможностями Spring, такими как удаленное взаимодействие, обмен сообщениями и выполнение заданий по расписанию. Первичной целью Spring Integration является предоставление простой модели создания интеграционных бизнес-решений, сохраняя разделение функциональных задач, что необходимо для написания поддерживаемого и тестируемого программного кода.

Spring Data REST – это часть проекта Spring Data, которая позволяет обращаться к JPA-репозиториям как к REST-сервисам.

Spring Data для Neo4j (NoSQL база данных основанная на принципе графов) – это часть проекта Spring Data, целью которой является предоставление совместимой и похожей на Spring модели программирования для определенной СУБД, сохраняя её первоначальные возможности и функциональность.

Spring Data для MongoDB (документно ориентированная СУБД) – это часть проекта Spring Data, целью которой является предоставление совместимой и похожей на Spring модели программирования для данной СУБД, сохраняя её возможности и функциональность.

Spring Framework – обеспечивает базовую поддержку управления зависимостями, управление транзакциями веб-приложений, доступ к данным, обмен сообщениями и другое.

Spring Cloud – проект упрощает подключение к облачным сервисам и получение возможностей окружения в облачных платформах, таких как Cloud Foundry и Heroku. Особая поддержка Spring-приложений через Java и XML конфигурации делает подключение к облачным сервисам тривиальной задачей. Для подключения могут быть использованы существующие облачные коннекторы (Cloud Foundry и Heroku) или разработаны собственные для конкретной облачной платформы.

Spring Social – обеспечивает соединение между Spring-приложением с поставщиками Software-as-a-Service (SaaS) API, таких как Facebook, Twitter и LinkedIn.

Spring Security - это мощный адаптируемый фреймворк аутентификации и контроля доступа. Он является де-факто стандартом для обеспечения безопасности в Spring-приложениях.

Spring Web Services – нацелен на облегчение разработки SOAP-сервиса методом Contract-First, позволяя создавать легкоизменяемые web-сервисы путем манипулирования XML-настройками.

Проведенный анализ компонент позволяет сделать вывод о целесообразности применения проекта Spring для создаваемой системы управления социальной научной кооперацией C2S и организации адекватной работы с Big Data при помощи Spring XD. Spring XD позволяет работать с различными базами данных, как через ORM так и напрямую.

Модель взаимодействия Spring XD с кластером данных на основе платформы OpenStack приведена на рис 4 и рис. 5.

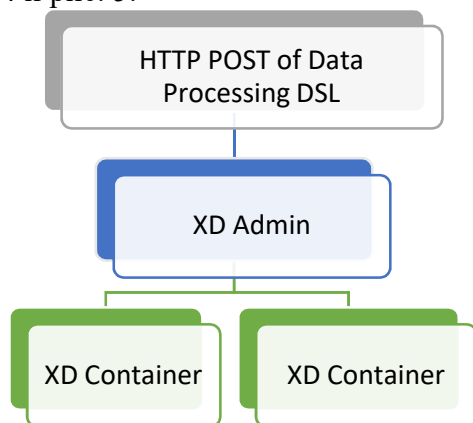


Рис. 4. – Концепция работы Spring XD с кластерами данных

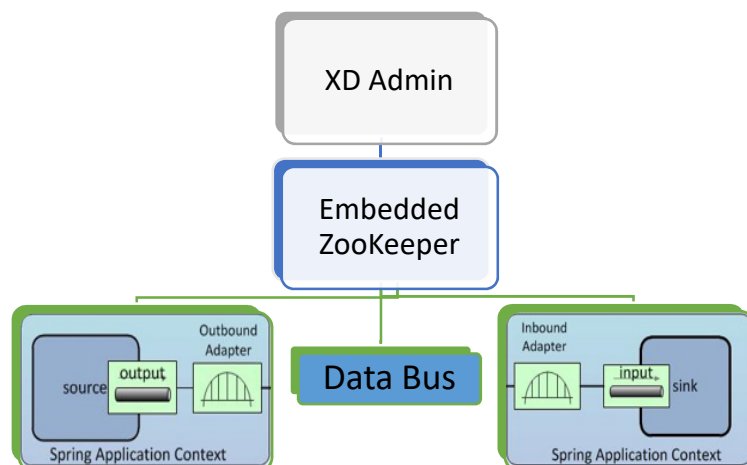


Рис. 5. – Раскрытие модели XD Container

Развернув данную модель на проекте C2S мы сможем организовать мониторинг изменения данных, получаемых интеллектуальными агентами от систем мониторинга многомерных объектов в режиме реального времени, что является сложной и критичной задачей.

Заключение. В работе приведен обзор Spring XD и базовая концепция работы с большими данными с помощью Spring XD. Рассмотренная технология является одним из способов решения задачи обработки Big Data. Перспективой дальнейших исследований является разработка формальной модели системы научной кооперации C2S.

Работа выполнена при финансовой поддержке Российского Фонда Фундаментальных исследований по проекту № 15-29-07936.

Библиографический список

1. Denisov V. S. Annotations-driven configuration framework for Java applications // International Journal of Open Information Technologies. 2015. – №10 – С.16–20.
2. Клеменков П. А., Кузнецов С. Д. Большие данные: современные подходы к хранению и обработке // Труды ИСП РАН. 2012. – №1. – С.143-158.
3. Ненаженко Д.В., Радченко Г.И. Удаленная визуализация больших объемов данных // Вестник ЮУрГУ. Серия: Вычислительная математика и информатика. 2015. – №1. – С.21–32.
4. Шарапов Руслан Владимирович Аппаратные средства хранения больших объёмов данных // ИВД. 2012. – №4/2. – С.67.
5. Документация Spring XD. Режим доступа: <http://docs.spring.io/spring-xd/docs/current-SNAPSHOT/reference/html/> (дата обращения: 04.05.2016).
6. Flo for Spring XD. Режим доступа: <http://docs.pivotal.io/spring-flo/> (дата обращения: 04.05.2016).
7. Шевченко В.И., Смирнов И.В., Разработка сервис-ориентированной среды для научной кооперации Материалы всероссийской научно-технической конференции молодых ученых, аспирантов и студентов «ИСУМ – 2016», Севастополь 19-21 мая 2016 г.– Севастополь: Изд-во СевГУ, 2016. – С. 386–390.
8. Шевченко В.И., Смирнов И.В., Тарасова А.В. Модель интерактивного взаимодействия с системами мониторинга на основе TELEGRAM API // Системы контроля окружающей среды — Севастополь: Изд-во НПЦ "ЭКОСИ-Гидрофизика", 2016. – №4 (24). – С. 62 – 65.
9. Скатков А.В., Шевченко В.И. Расширение референтной модели облачной вычислительной среды в концепции крупномасштабных научных исследований // Труды ИСП РАН. 2015. – №6. – С.285–306.

УДК 004.048:004.622

С.С. Струг, магистрант студент 6-го курса

Научный руководитель: Т.А. Васяева, канд. техн. наук, доцент кафедры АСУ

Донецкий национальный технический университет

ЭВОЛЮЦИОННЫЙ ПОДХОД К ПОСТРОЕНИЮ ПРОДУКЦИОННОЙ МОДЕЛИ ДЛЯ ЭКСПЕРТНЫХ СИСТЕМ

Аннотация: В статье рассматривается применение генетического программирования для получения продукционных правил. Компьютерные программы представлены в виде деревьев (правил принятия решений), а генерация поколений достигается путем применения генетических операторов: мутация, кроссингвер, репродукция.

Ключевые слова: Генетическое программирование, мутация, кроссингвер, репродукция, терминальное множество, функциональное множество

Annotation: The article deals with the application of genetic programming to obtain production rules. Computer programs can be represented as trees (decision rules), and the production of successive generations is achieved by applying genetic operators: mutation, crossover, reproduction.

Key words: Genetic Programming, mutation, crossover, reproduction, terminal set, function set

Введение

Одной из самых серьезных проблем и трудно формализуемых при разработке экспертных систем является извлечение знаний у экспертов в виде системы правил (продукций). В работе предпринята попытка автоматизировать этот процесс. Здесь в качестве потенциального решения, особи популяции, рассматривается продукция (система продукций).

Генетическое программирование

Для решения поставленной задачи предложено использовать генетическое программирование (ГП) [1]. Решение задачи на основе ГП можно представить следующей последовательностью действий.

1. Установка параметров эволюции;
2. Инициализация начальной популяции;
3. $T:=0$;
4. Оценка особей, входящих в популяцию;
5. $T:=T+1$;
6. Отбор родителей;
7. Создание потомков выбранных пар родителей – выполнение оператор кроссингвера;
8. Мутация новых особей;
9. Расширение популяции новыми порожденными особями;
10. Сокращение расширенной популяции до исходного размера;
11. Если критерий останова алгоритма выполнен, то выбор лучшей особи в конечной популяции – результат работы алгоритма. Иначе переход на шаг 4.

Предлагается следующий метод кодирования особей для генетического программирования. Каждая особь представляет собой дерево, которое соответствует синтаксическому выражению, представляющее множество правил в дизъюнктивной нормальной форме [2]. На рисунке 1 представлен пример дерева в дизъюнктивной нормальной форме. Дерево представлено 3-мя правилами. Такое представление особи значительно упрощает интерпретацию результата. В данном примере расшифровка будет следующей: ЕСЛИ правило 1 ИЛИ правило 2 ИЛИ правило 3, ТО результат 1, ИНАЧЕ результат 2.

По построенному специальным образом дереву можно получить систему продукций, которая классифицирует с заданной точностью данные обучающей выборки. Входное обучающее множество должно быть представлено в виде булевых переменных, которые представляют терминальное Функциональное множество состоит из логических операций: AND, OR, NOT, которые представляют внутренние вершины дерева. В качестве фитнес-функции рассматривается доля правильно поставленных результатов.

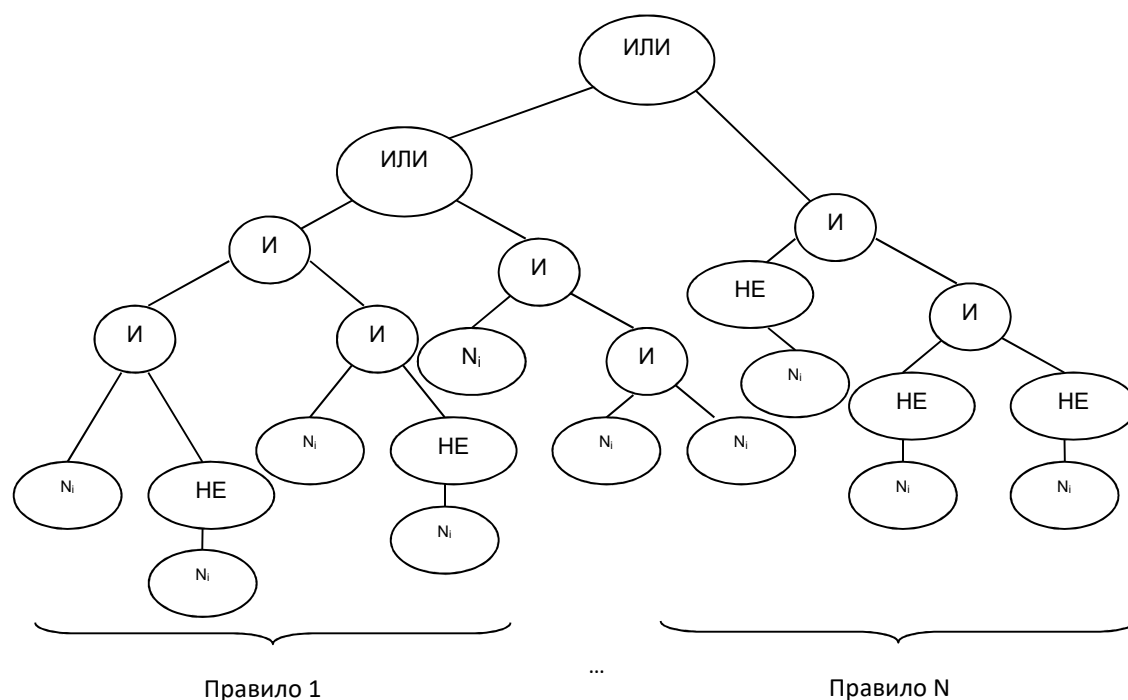


Рис. 1. Пример дерева в дизъюнктивной нормальной форме.

Генерация начальной популяции

На данном этапе происходит генерация начальной популяции, в соответствии с заданными параметрами. Популяция состоит из набора деревьев, сгенерированных случайным образом. Генерация каждого дерева происходит рекурсивно, начиная с генерации первым функционального узла ИЛИ и его аргументов. В качестве аргументов на первом шаге может быть только узел ИЛИ. Далее для каждого дочернего узла случайным образом определяется тип и значения его аргументов по следующим принципам:

- после узла ИЛИ может быть только функциональный узел (значениями которого могут быть – ИЛИ или И);
- после узла И может быть функциональный узел (значениями которого могут быть – И или НЕ) или терминальные узлы;
- после узла НЕ может быть только терминальный узел.

Процесс выполняется по левой ветви до тех пор, пока не будет выбран дочерним терминальный узел. Затем генерируются правые ветви.

Вероятность функционального и терминального узлов меняется по следующему принципу: чем ниже вершина, тем больше вероятность терминального узла и меньше функционального. Для функционального узла на каждом последующем шаге увеличивается вероятность узла И и уменьшается вероятность узла ИЛИ.

При формировании дерева в одной ветви ИЛИ (т.е. для одного правила) не используется один и тот же терминальный символ более одного раза.

Предусмотрены методы создания деревьев: полный, растущий и комбинированный.

Отбор родителей. Предложено использовать отбор пропорционально значению целевой функции реализованный методом рулетки или турниром. При этом если два или более потомка имеют одинаковую фитнес-функцию, то выбирается дерево минимальной сложности.

Кроссинговер

Для древообразной формы представления используются следующие три основных оператора кроссинговера:

- узловой кроссинговер;
- кроссинговер поддеревьев;
- смешанный.

Учитывая строго определенное представление дерева необходимо модифицировать операторы кроссинговера. В узловом операторе кроссинговера обмен возможен только для терминальных узлов. В кроссинговере поддеревьев родители могут обмениваться только поддеревьями ветви И. При смешанном операторе кроссинговера для некоторых узлов выполняется узловой оператор кроссинговера, а для других - кроссинговер поддеревьев.

Также предлагается выполнять оператор кроссинговера для худшего правила в дереве. Правило считается худшим, если целевая функция имеет минимальное значение. Каждое правило можно рассматривать как отдельное дерево способное решать поставленную задачу, поэтому вычисление фитнес функции для каждого правила в отдельности логически обосновано.

Вычисление фитнес функции не только для каждого правила в отдельности, но и каждого узла И также имеет смысл. При выполнении оператора кроссинговера поддеревьев предлагается осуществлять поиск точки разрыва следующим образом: вычисляется фитнес функции для каждого узла И начиная с первого снизу. Если значение фитнес функции для узла И находящегося выше, хуже чем на предыдущем шаге то обмену подлежит один из узлов аргументов данного узла И.

Мутация

Для деревьев используются следующие операторы мутации: узловая; усекающая; растущая. Как и в случае с оператором кроссинговера, оператор мутации должен быть модифицирован. Узловая мутация выполняется для терминального узла или первой снизу вершины ИЛИ. Усекающая мутация выполняется только для узлов И или НЕ. При растущей мутации ветви наращиваются согласно правилам инициализации деревьев.

Сокращение дерева

Предлагается использовать оператор сокращения дерева. Как и оператор кроссинговера или мутации, данный оператор выполняется с определенной вероятностью. Если количество правил в дереве превышает определенный порог, то обрезаются целое правило. Если количество правил не превышает указанное число, то обрезаются худшая часть дерева в худшем правиле, т.е. выполняется усекающая мутация.

Редукция

Предлагается использовать выполнения следующих вариантов редукции: элитная стратегия; чистая замена; равномерная случайная замена (с указанием количества заменяемых особей в %).

Выводы

На основе генетического программирования разработан математический аппарат для получения продукционных правил. Предложенный метод может быть использован при решении задач классификации на 2 класса.

Библиографический список

1. Ю.А. Скобцов. Основы эволюционных вычислений.- Навчальний посібник. – Донецьк: ДонНТУ, 2008.- 326с.
2. Скобцов Ю.А. Эволюционный подход к формированию знаний для медицинских экспертных систем с учетом неопределенности данных / Скобцов Ю.А., Васяева Т.А. // Штучний інтелект. – 2008. – № 4. – С. 631-637.

УДК 004.62:004.9

О. С. Черненко, студентка 4-го курса

Научный руководитель: В. А. Строганов, ст. преп.

Севастопольский государственный университет

ПРИМЕНЕНИЕ TF-IDF АЛГОРИТМА В РЕКОМЕНДАТЕЛЬНЫХ СИСТЕМАХ ГОСУДАРСТВЕННЫХ ЗАКУПОК

Аннотация: в статье описан tf-idf алгоритм и приведено его применение в рекомендательной системе государственных закупок.

Ключевые слова: рекомендательные системы, контентный анализ, tf-idf алгоритм, государственные закупки, классификация.

Annotation: the article describes the tf-idf algorithm and its use in the recommendation system of state trades.

Key words: recommendation systems, content analysis, tf-idf algorithm, state trades, classifications.

Рекомендательная система – это система, которая на основании поведения пользователя (просмотров, оценок, комментариев) определяет его интересы и выполняет формирование рекомендаций. В зависимости от типа используемых данных выделяют следующие классы рекомендательных систем:

- системы, использующие коллаборативную фильтрацию;
- системы, основанные на контентном анализе;
- гибридные системы.

Первый класс методов основывается на схожести поведения пользователей. Выполняется анализ действий конкретного пользователя и сравнение его с другими пользователями системы. Таким образом, основная идея методов данного класса состоит в том, что похожие по поведению пользователи имеют схожие интересы и предпочтения. Основным недостатком данной группы методов является «холодный старт» – отсутствие рекомендаций для новых пользователей [1]. Второй класс систем основывается на анализе контента рекомендуемых объектов, на основании этого выполняется определение степени их схожести. Третий класс компонует методы контентного анализа и коллаборативной фильтрации [2].

В настоящее время на электронных площадках, которые обеспечивают проведение аукционов согласно законодательству РФ, выделяют две основные роли: участник и заказчик. Заказчик является инициатором аукциона, и выполняет задачу создание закупок, участник закупок является поставщиком товаров и услуг. Для привлечения большего числа участников к участию в закупке целесообразно внедрять механизм рекомендаций на подобных площадках. Рекомендательная система посредством предоставления списка рекомендаций, основываясь на истории действий пользователя (история просмотров, участие в аукционах), позволит увеличить число заявок на участие и расширить для заказчиков область выбора, а для участников ускорить процесс поиска. Так как в сфере государственных закупок необходимо обеспечение конфиденциальности участников аукционов, сравнение поведения пользователей будет нарушением этого принципа, поэтому наиболее подходящим классом является контентно ориентированная система.

В качестве алгоритма для реализации контентного анализа был выбран TF-IDF (term frequency – inverse document frequency) алгоритм. Основной идеей данного алгоритма является подсчет частоты встречаемости термина в документе. Исходный набор данных для данного алгоритма представляет собой совокупность документов, разбитых на слова [1]. Коэффициент tf-idf отражает степень важности термина в документе. TF – частота встречаемости термина в документе, в простейшем случае определяется по формуле 1.

$$tf = \frac{t}{N}, \quad (1)$$

где t – количество вхождения термина в документе, N – общее число слов в документе. Так как частицы, союзы и прочие служебные части речи имеют большую частоту встречаемости в документе, но не отражают содержание документа, необходимо ввести еще один показатель – idf (инверсная частота документов) [3]. Данный показатель позволит отсеять часто употребляемые слова не показывающие основной смысл, определяется по формуле 2.

$$idf = \frac{D}{DF}, \quad (2)$$

где D – общее число документов в коллекции, DF – количество документов, содержащих искомый термин [4].

Таким образом основная идея данного алгоритма состоит в том, что термин является более важным, если он имеет большое число вхождений в один документ и редко входит в остальные документы коллекции, то есть не является распространённым [1].

Для того, чтобы применить рассмотренный выше алгоритм в рекомендательной системе необходимо выделить основные характеристики закупок. К ним могут быть отнесены: заказчик, организующий закупку, регион, правило проведения закупки, наименование, классификатор ОКПД2. Основными характеристиками, которые могут быть использованы для идентификации, являются ОКПД2, наименование закупки, поэтому для простоты будем учитывать при анализе только эти показатели.

С целью получения информации об интересах участника в системе предусмотрен механизм отслеживания взаимодействия пользователя с системой: просмотры закупок, подача заявок. На основании данных действий формируется профиль пользователя. В профиле сохраняются последние закупки, с которыми взаимодействовал пользователь, в дальнейшем эти закупки выступают в качестве ключевых при формировании рекомендаций.

Процесс формирования рекомендаций начинается с анализа профиля пользователя. Происходит выделение ключевых слов из наименований закупок на основании tf-idf алгоритма. В ходе анализа происходит выделение части ОКПД2 кода (первые две цифры), которые указывают на класс рассматриваемой продукции или услуг, данный показатель позволит сузить поиск в базе. На основании полученного массива ключевых слов и класса ОКПД2, происходит анализ существующих в системе закупок в статусе «прием заявок» и принадлежащих к выделенному классу. Для каждой закупки, удовлетворяющей условию, происходит подсчет tf показателя и на основании полученных оценок происходит ранжирование закупок. В результате анализа пользователю в качестве рекомендаций выдаются n – закупок с максимальным показателем.

Подытожив можно сказать, что была реализована система рекомендаций государственных закупок для электронной площадки на основании tf-idf алгоритма. В общем случае были достигнуты поставленные цели – не смотря на простоту рассмотренного алгоритма, система предоставляет рекомендации близкие к интересам пользователя и обеспечивает достаточную точность. В перспективе развития проекта, с целью увеличения точности рекомендаций возможно добавление в систему других алгоритмов и выдача пользователю пересечения полученных результатов.

Библиографический список

1. Гомзин, А.Г. Системы рекомендаций: обзор современных подходов [Электронный ресурс]/ А.Г. Гомзин, А.В. Коршунов. – Электронные данные. – Режим доступа: http://www.ispras.ru/proceedings/docs/2012/22/isp_22_2012_401.pdf, свободный.
2. Джонс, М. Рекомендательные системы [Электронный ресурс]: Часть 1. Введение в подходы и алгоритмы/ М. Джонс – Электронные данные. – Режим доступа: <http://www.ibm.com/developerworks/ru/library/os-recommender1/index.html>, свободный.
3. Beginners Guide to learn about Content Based Recommender Engines [Электронный ресурс] – Электронные данные. – Режим доступа: <https://www.analyticsvidhya.com/blog/2015/08/beginners-guide-learn-content-based-recommender-systems/>, свободный.
4. Гулин, А. Алгоритм текстового ранжирования Яндекса на РОМИП-2006[Электронный ресурс]/ А. Гулин, М. Маслов, И. Сегалович – Электронные данные. – Режим доступа: http://cache-kiev01.cdn.yandex.net/download.yandex.ru/company/03_yandex.pdf, свободный.

УДК 004.588+004.827

Д.В.Шулянский, магистрант

Научный руководитель: С.В.Хмелевой, канд. техн. наук, доц.

Донецкий национальный технический университет

ОПРЕДЕЛЕНИЕ СХОЖЕСТИ СЛОВ ПРИ ПОМОЩИ ОРФОГРАФИЧЕСКОГО И ФОНЕТИЧЕСКОГО АНАЛИЗА В СИСТЕМАХ АВТОМАТИЧЕСКОГО ТЕСТИРОВАНИЯ ЗНАНИЙ

Аннотация: Целью данной работы является анализ существующих методов анализа текста, а также исследование применимости различных методов сравнения текста в системах автоматического тестирования языковых знаний. Помимо этого в статье предлагается метод, с помощью которого можно оценивать ответы учащихся.

Ключевые слова: орфографический анализ, фонетический анализ, мера схожести слов, тестирование знаний, ошибки

Annotation: The purpose of this paper is to analyze existing methods of text analysis, and to study the applicability of various text comparison methods in systems of natural languages knowledge assessment. The author suggests a method by which students' responses can be assessed.

Key words: orthographic analysis, phonetic analysis, measure of stings similarity, knowledge assessment, mistakes

1. Введение

На сегодняшний день во всем мире стремительно развиваются информационные технологии, в том числе в сфере обучения и контроля знаний. На фоне распространения дистанционного обучения задача быстрого и объективного автоматического контроля знаний обучающихся становится более актуальной. Не смотря на это, практически все существующие автоматизированные методы оценки знаний значительно уступают экспертному оцениванию знаний.

Чтобы система тестирования языковых знаний давала гибкую оценку работе учащихся, необходимо, чтобы она сама определяла критичность допущенных ошибок, и, исходя из этого, решала, какую оценку поставить за выполнение упражнения. Для реализации поставленной задачи необходимо проанализировать существующие методы анализа слов, а также предложить метод определения их схожести.

2. Орфографическая схожесть слов

Алгоритмы нечеткого поиска (также известного как поиск по сходству) являются основой систем проверки орфографии и полноценных поисковых систем вроде Google, Yandex или офисных пакетов, например, Microsoft Office. Например, такие алгоритмы используются для функций наподобие «Возможно вы имели в виду ...» в тех же поисковых системах.

Для ознакомления с алгоритмами нечёткого поиска воспользуемся работами Мосалева [1], Сметанина [2] и Бойцова [3].

Проанализировав вышеперечисленные работы, видим, что основополагающей алгоритмов нечёткого поиска являются метрики. Метрикой нечеткого поиска называют функцию расстояния между двумя словами, позволяющую оценить степень их сходства. С помощью этих метрик и будем определять схожесть словоформ.

Наиболее известны следующие метрики: расстояния Хемминга, Левенштейна, Дамерау – Левенштейна, Джаро-Винклера, Вагнера — Фишера, метод n-грамм, метод распознавания форм Ратклиффа/Обершелпа.

Необходимо выделить метрики для работы со всеми типами ошибок. Выделяют четыре типа случайных ошибок, встречающихся в письменной речи:

- пропуск символа (lisening);
- замена одного символа любым другим (aducation);
- удвоение символа (frozen);
- перестановка двух соседних символов (fligth).

Расстояние Хемминга для реализации поставленных целей совершенно не подходит, поскольку алгоритм работает только со строками одинаковой длины.

Наиболее распространённой метрикой является расстояние Левенштейна или расстояние редактирования и его более «экономная» модификация – алгоритм Вангера – Фишера. Данные алгоритмы работают со следующими ошибками: пропуск символа, замена и удвоение символов. Перестановка же двух соседних символов (например, «Flight» и «Fligh») сочтётся за двойную ошибку, что, по нашему мнению, неправильно для английского языка.

Описанного недостатка лишены следующие метрики: расстояние Дамерау – Левенштейна, расстояние Джаро-Винклера и метод распознавания форм Ратклиффа/Обершелпа.

Расстояние Дамерау – Левенштейна является модификацией расстояния Левенштейна. В отличие от расстояния Левенштейна добавлена операция транспозиции символов.

Иными алгоритмами, учитывающими ошибку транспозиции, являются расстояние Джаро-Винклера и метод распознавания форм Ратклиффа/Обершелпа. Проанализировав работы Александра Родишевского [4] видим, что данные методы сравнения строк очень похожи между собой, однако автор для работы с опечатками и орфографическими ошибками рекомендует использовать расстояние Джаро-Винклера.

Из-за того, что результатом работы алгоритма Джаро-Винклера получается вещественное число, лежащее в диапазоне от 0 (строки совершенно не похожи) до 1 (строки идентичны), данный метод нельзя считать метрикой, а результат работы – расстоянием. Результат работы этого алгоритма является мерой сходства между строками.

3. Фонетическая схожесть слов

При изучении иностранных языков часто возникает ситуация, когда человек помнит, как слово звучит, однако не помнит точное написание этих слов, что также приводит к ошибкам написания. Чтобы учесть подобные случаи необходимо применять алгоритмы фонетического кодирования.

Алгоритмы фонетического кодирования сопоставляют слову его фонетический код, который одинаков для схожих по звучанию лингвистических единиц. Основные алгоритмы фонетического кодирования:

- Soundex[5]
- NYSIIS [7]
- Metaphone [8]

Soundex[4,5,6] был разработан для построения фонетических кодов личных имён. Данный алгоритм кодирует буквы слова цифрами, причём первый символ кода – буква. Например:

- ammonium : A555
- implementation : I514
- Robert : R163

Алгоритм NYSIIS («New York State Identification and Intelligence System») [4, 7] даёт несколько лучшие результаты относительно оригинального Soundex, используя более сложные правила преобразования исходного слова в результирующий код. Этот алгоритм также был разработан для работы с собственными именами. Фонетический код, получаемый с помощью алгоритма NYSIIS состоит из 6 букв, полученных на основе входного слова.

Алгоритм фонетического кодирования Metaphone [4, 8] был также разработан в ответ на недостатки Soundex. Алгоритм преобразует исходное слово с учетом правил английского языка, при этом теряется значительно меньше информации. Metaphone использует 16 символов согласных 0BFHJKLMNPRSTWXY. «0» представляет «th», «X» представляет «sh» или «ch», остальные представляют и обычное звучание в английском языке. В начале слова также могут быть гласные из множества AEIOU.

4. Тестирование методов на контрольном наборе слов с ошибками

В ходе испытаний различных методов орфографического и фонетического анализа были воспроизведены такие ошибки:

- Пропуск символа

- Замена символа другим
- Транспозиция – перестановка двух символов местами
- Вставка символа
- Неверное употребление времени глагола
- Неверно употреблённое лицо глагола
- Слова, схожие по звучанию, однако орфографически они разнятся.

На контрольном наборе были протестированы следующие методы анализа строк:

- Орфографический анализ:
 - Схожесть строк по Джаро-Винклеру
 - Расстояние Левенштейна
 - Расстояние Демарау-Левенштейна
- Методы фонетического кодирования:
 - Метафон
 - Двойной метафон
 - NYSIIS
 - Каверфон 2.0
 - Cologne Phonetic

Были отобраны лучшие методы, результаты работы которых сведены в таблицу 1.

Таблица 1. Результаты испытаний методов анализа строк на тестовом наборе

Тестовый набор		Орфографический анализ			Фонетический анализ	
Правильный ответ	Ответ с ошибкой	Схожесть по Джаро-Винклеру	Расстояние Левенштейна	Расстояние Демарау-Левенштейна	Метафон	NYSIIS
Пропуск символа						
have been reading	have ben reading	0,9640	1	1	HFBN - HFBN	HAVABA - HAVABA
have not finished	have not finishd	0,9977	1	1	HFNT - HFNT	HAVANA - HAVANA
meant	meat	0,9533	1	1	MNT - MT	MAD - MAT
am having	am havin	0,9926	1	1	AMHF - AMHF	ANAVAN - ANAVAN
have been washing	ave been washing	0,9804	1	1	HFBN - AFBN	HAVABA - AVABAN
Замена символа						
have broken	hase broken	0,9504	1	1	HFBR - HSBK	HAVABR - HASABR
mix	mex	0,8000	1	1	MKS - MKS	MAX - MAX
come	came	0,8500	1	1	KM - KM	CAN - CAN
held	hold	0,8500	1	1	HLT - HLT	HALD - HALD
goes	does	0,8333	1	1	KS - TS	G - D
mean	meat	0,8833	1	1	MN - MT	MAN - MAT
Транспозиция						
thinking	htinking	0,9583	2	1	ONKN - TNKN	TANCAN - HTANCA
brighten	brighthen	0,9750	2	1	BRTN - BRK0	BRAGTA - BRAGTA
has been teaching	has been taeching	0,9919	2	1	HSBN - HSBN	HASBAN - HASBAN
have bought	have boughth	0,9945	2	1	HFBT - HFBK	HAVABA - HAVABA
catch	cathc	0,9533	2	1	KX - K0K	CATC - CATC

Таблица 1. (продолжение)

Тестовый набор		Орфографический анализ			Фонетический анализ	
Правильный ответ	Ответ с ошибкой	Схожесть по Джаро-Винклеру	Расстояние Левенштейна	Расстояние Демарау-Левенштейна	Метафон	NYSIIS
Вставка символа						
deal	dealt	0,9600	1	1	TL - TLT	DAL - DALT
am writing	am writting	0,9890	1	1	AMRT - AMRT	ANWRAT - ANWRAT
have typed	have typpe	0,9917	1	1	HFTP - HFTP	HAVATY - HAVATY
has been planting	has been plaanting	0,9938	1	1	HSBN - HSBN	HASBAN - HASBAN
broke	bbroke	0,9500	1	1	BRK - BRK	BRAC - BRAC
Неверное время						
have heard	heard	0,5222	5	5	HFHR - HRT	HAVAHA - HAD
am going	go	0,7500	6	6	AMKN - K	ANGANG - G
have worked	worked	0,4192	5	5	HFWR - WRKT	HAVARC - WARCAD
have been practising	practised	0,4407	13	13	HFBN - PRKT	HAVABA - PRACTA
go	went	0,0000	4	4	K - WNT	G - WAD
Неверное лицо						
smells	smell	0,9722	1	1	SMLS - SML	SNAL - SNAL
have	has	0,7778	2	2	HF - HS	HAV - H
tastes	taste	0,9722	1	1	TSTS - TST	TAST - TAST
am feeling	is feeling	0,8667	2	2	AMFL - ISFL	ANFALA - ISFALA
shall	will	0,6333	3	3	XL - WL	SAL - WAL
Произношение						
bend	bent	0,8833	1	1	BNT - BNT	BAD - BAD
is weighing	is wein	0,9449	4	4	ISWK - ISWN	ISWAGA - ISWAN
am thinking	am sinking	0,9317	2	2	AM0N - AMSN	ANTANC - ANSANC
read	reed	0,8667	1	1	RT - RT	RAD - RAD
fly	flai	0,7778	2	2	FL - FL	FLY - FL
looks	loox	0,8483	2	2	LKS - LKS	LAC - LAX

Ошибки пропуска символа успешно обнаруживают методы орфографического анализа строк: схожесть по Джаро-Винклеру, расстояние Левенштейна и расстояние Демарау-Левенштейна. Однако, при таких ошибках, как meant-meat, когда пропуск символа может привести к совершенно другому слову. Ни один из представленных методов не позволяет оценить должным образом, поскольку они работают просто с набором символов. Для устранения данной проблемы необходим словарный анализ текста, который сможет обнаружить ошибку подобного рода.

С ошибками с заменой символа несколько лучше работает метод схожести Джаро-Винклера, и методы фонетического кодирования. Данные ошибки оказывают большое влияние на неправильные глаголы английского языка: замена какого-либо символа может привести к иной форме глагола, или изменится смысл слова. Для работы с такими ошибками необходимо выполнять дополнительный анализ словоформ – морфологический анализ словоформ.

Поскольку транспозицию двух соседних символов считаем за одну ошибку, то расстояние Левенштейна не подходит для определения ошибок данного рода. Методы фонетического кодирования весьма чувствительны к транспозиции символов – порой получаются очень разные коды. На наш взгляд мера схожести строк по Джаро-Винклеру с данной ошибкой справляется лучше остальных метрик. Вследствие перестановки двух символов получение другого слова или изменение формы маловероятно.

Применять приведенные методы анализа строк для определения неверно употребленного времени глаголов неправильно, хотя они и дают большую разницу между словами в приведенном тестовом наборе. В случае правильных глаголов используемые методы дадут неверный результат. Для распознавания подобного рода ошибок необходимо определять формы слова, что является задачей морфологического анализа.

Орфографические методы анализа строк нельзя применять для ошибок неверно использованного лица глаголов – они не чувствительны к таким ошибкам. Несколько лучшие результаты показывают методы фонетического анализа слов. Для английского языка метод фонетического кодирования Метафон лучше справляется с данной ошибкой, ведь он отбрасывает гласные буквы, если они не в начале слова, что позволяет акцентировать внимание на согласных, которые в английском языке являются основными образующими различных лиц глаголов. Несмотря на преимущества данного метода для определения подобных ошибок гораздо правильнее использовать анализ словоформ – морфологический анализ слов.

Методы фонетического кодирования слов малочувствительны к ошибкам, вследствие которых слова звучат одинаково либо похоже. Как раз этот момент и определит тот факт, что слова звучат одинаково, из-за чего можно ошибиться. Количество ошибок в слове при этом определит орфографический анализ строк. При этом необходимо учитывать, что в английском языке различные формы неправильных глаголов звучат очень похоже, однако в данном случае ошибка является критичной. Для того, чтобы определить такие критические моменты, необходимо использовать морфологический анализ словоформ.

4. Метод определения схожести слов

Проанализировав возможные ошибки, методы определения схожести слов или расстояния между ними, а также алгоритмами построения фонетических кодов слов видим, что использование лишь вышеперечисленных средств анализа строк недостаточно для полного анализа ошибок при изучении английского языка. Причиной этого является то, что все алгоритмы работают со словами как с набором символов, а не смысловой единицей языка.

Чтобы сделать тестирование языковых знаний автоматическим сначала необходимо получить морфологические признаки правильного ответа, затем наиболее вероятные признаки ответа учащегося. Поскольку ответы могут содержать ошибки, необходимо определить, на какую словоформу они похожи, затем получать её признаки. Для определения схожести будем использовать меру схожести по Джаро-Винклеру. Если морфологические признаки словоформы, на которую ответ похож наибольшим образом, соответствует признакам правильного ответа, тогда уже стоит рассмотреть ответ учащегося с точки зрения «набора символов»

Прежде всего, необходимо определить, схож ли ответ учащегося с правильным ответ по звучанию. Для этого используем метод фонетического кодирования Метафон, поскольку на наш взгляд он несколько лучше отражает реальное звучание слова, не смотря на отсутствие гласных букв, которые могут читаться по-разному, например, слово *read* в инфинитиве и второй форме глагола. Однако необходимо модифицировать алгоритм для выдачи более длинных кодов. Затем с помощью расстояния Левенштейна определяем расстояние редактирования между этими кодами. Выбор пал на эту метрику того, что перепутать два звука местами сложнее, нежели буквы, а пропущенные гласные буквы в фонетическом коде вынуждают относиться к ошибкам критичнее. В общем случае формула для определения схожести звучания слов выглядит следующим образом:

$$P_s = \frac{|Ph1|-1}{|Ph1|}, \quad (1)$$

Где P_s – фонетическая схожесть двух строк $|Ph1|$ – длина фонетического кода правильного ответа, 1 – расстояние между фонетическими кодами правильного ответа и ответа учащегося. При этом 1 – строки имеют одинаковое или близкое звучание, 0 – строки абсолютно несозвучны.

После определения фонетической схожести строк необходимо получить орфографическую схожесть. Делаем это при помощи расстояния Демарау-Левенштейна. Формула вычисления орфографической схожести при этом будет аналогична фонетической (1):

$$O_s = \frac{|S1|-1}{|S1|}, \quad (2)$$

Где O_s – орфографическая схожесть двух строк $|S1|$ – длина фонетического кода правильного ответа, 1 – расстояние между правильным ответом и ответом учащегося. При этом 0 – строки абсолютно несозвучны, 1 – строки имеют одинаковое или близкое звучание.

В конечном итоге схожесть слов определяется по следующей формуле:

$$Sim = P_s * O_s, \quad (3)$$

Где Sim – уровень схожести слов.

В итоге получаем метод определения схожести слов по звучанию и по написанию, что позволяет использовать его в системах тестирования языковых знаний.

Вывод

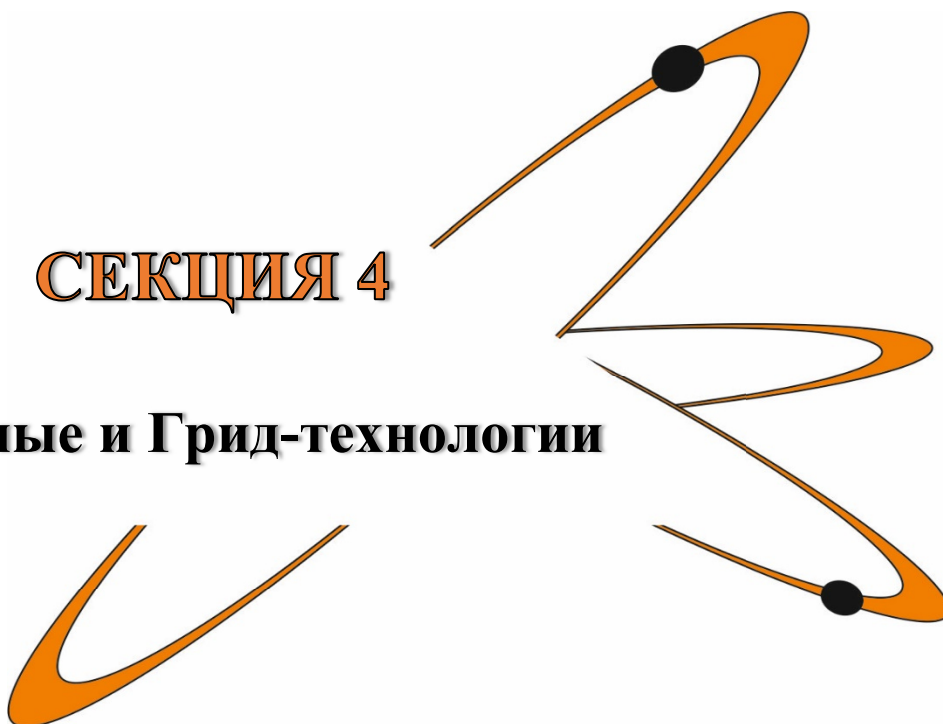
В данной статье автором были проанализированы методы анализа текста, рассмотрены соответствующие алгоритмы. На основе приведенного тестового набора была исследована возможность применения орфографических и фонетических методов сравнения строк в системах автоматического тестирования языковых знаний. В конце работы предложен метод определения схожести текста, который может быть применён при тестировании знаний языков.

Библиографический список

1. П.М. Мосалев. «Обзор методов нечеткого поиска текстовой информации» – Журнал «Вестник Московского государственного университета печати» Выпуск № 2 / 2013.
2. Н.Сметанин. «Нечёткий поиск в тексте и словаре» [Электронный ресурс] / Хабрахабр [2011] - Режим доступа: <https://habrahabr.ru/post/114997/> .- Загл. с экрана
3. Бойцов Л.М. «Классификация и экспериментальное исследование современных алгоритмов нечеткого словарного поиска» [Электронный ресурс] / Searchivarius - Режим доступа: <http://boytsov.info/pubs/rcdl2004.pdf> .- Загл. с экрана
4. Родишевский А. «Алгоритмы приблизительного сравнения текста, часть 1» [Электронный ресурс] / Мосфоэдро - Режим доступа: <http://www.morfoedro.it/doc.php?n=222&lang=ru> .- Загл. с экрана
5. Родишевский А. «Алгоритмы приблизительного сравнения текста, часть 2» [Электронный ресурс] / Мосфоэдро - Режим доступа: <http://www.morfoedro.it/doc.php?n=223&lang=ru> .- Загл. с экрана
6. Soundex [Электронный ресурс] / Википедия – свободная энциклопедия - Режим доступа: <https://ru.wikipedia.org/wiki/Soundex> .- Загл. с экрана
7. NYSIIS [Электронный ресурс] / Википедия – свободная энциклопедия - Режим доступа: <https://uk.wikipedia.org/wiki/NYSIIS> .- Загл. с экрана
8. Метафон [Электронный ресурс] / Википедия – свободная энциклопедия - Режим доступа: <https://uk.wikipedia.org/wiki/%D0%9C%D0%B5%D1%82%D0%B0%D1%84%D0%BE%D0%BD> .- Загл. с экрана

СЕКЦИЯ 4

Облачные и Грид-технологии



УДК 004.052

М.С. Лисов, магистрант**Научный руководитель: А.А. Брюховецкий, канд. техн. наук, доц***Севастопольский государственный университет***ИССЛЕДОВАНИЕ ДИНАМИКИ ХАРАКТЕРИСТИК ТРАФИКА В ОБЛАЧНОЙ ВЫЧИСЛИТЕЛЬНОЙ СРЕДЕ**

Аннотация: В настоящее время стремительно увеличивается потребность в хранении и обработке больших объёмов данных, что провоцирует интенсивное развитие вычислительных сред, использующих облачную парадигму. Однако этим революционным технологиям сопутствует ряд уязвимостей, одним из вариантов возможных контрмер для провайдеров является тщательный мониторинг облачного трафика. Авторами рассматривается один из возможных методов мониторинга неоднородности трафика в cloud-среде.

Ключевые слова: управление рисками, облачные вычисления, мониторинг, состояние трафика, G-эффект, неоднородность среды.

Annotation: At present, the need for storing and processing large amounts of data is rapidly increasing, which provokes intensive development of computing environments using the cloud paradigm. However, this revolutionary technology is accompanied by a number of vulnerabilities, one of possible countermeasures for providers is the careful monitoring cloud traffic. The authors consider one of the possible methods of monitoring the heterogeneity of traffic in cloud environment.

Keywords: cloud computing, monitoring, traffic state, G-effect, environmental heterogeneity.

Облачные вычисления — это модель предоставления потребителю масштабируемых вычислительных ресурсов в виде сервиса через Интернет. Использование IaaS и PaaS моделей обслуживания позволяет ИТ-компаниям получать преимущества в области повышения эффективности, экономии затрат и масштабирования [1]. В условиях, в том числе, облачных вычислительных сред (ОВС) большая часть проблем анализа данных связана с исследованиями стохастических динамических систем, в которых обнаружение существенных, но редких информационных ситуаций часто имеет решающее значение.

Целями мониторинга современных систем могут быть [2-5]:

- сбор информации об элементах cloud-среды;
- детальный анализ и исследование облачной системы на предмет поиска узких, высоконагруженных мест;
- обеспечение масштабируемости вычислительных ресурсов, позволяющей управлять серверами различных провайдеров;
- отслеживание системных событий и операций и др.

Известные в настоящее время методы для решения задач мониторинга в облачной системе не могут рассматриваться как эталонное решение, так как не обеспечивают требуемого качества контроля состояния трафика, в частности, с их помощью затруднительно отследить моменты изменения таких свойств трафика как нестационарность, дисперсность, интенсивность. С этой целью авторами рассматривается понятие гетероскедастичности (G-эффект) [6] как неоднородности наблюдений, выражающегося в неодинаковой дисперсии случайной ошибки регрессионной модели. Факт неоднородности состояния сетевого трафика может использоваться для обнаружения внешних воздействий (аномальные события), или изменения состояния сетевого трафика, которые могут быть вызваны при взаимодействии триады: поставщик-брокер-потребитель сервисных услуг. Решение о неоднородности состояния трафика принимается на основе оценки изменения дисперсий остатков, с использованием метода непараметрической статистики (критерий Спирмена).

Для исследования динамики качественного изменения состояния облачного трафика была построена модель СМО с отказами в имитационной среде моделирования Anylogic. Модель исследовалась при следующих условиях:

- интенсивность генерации заявок $M(\lambda)=0,7$, $\sigma(\lambda)=0,09$;
- интенсивность обслуживания заявок $M(\mu)=0,8$ и $\sigma(\mu)=[0,17; 0,2; \dots; 0,4]$
- емкость накопителя – 5, алгоритм функционирования – FIFO;

В таблице 1 представлены некоторые результаты исследования, где выделены значения, соответствующие наличию G-эффекта. Здесь приняты следующие обозначения:
 при $\alpha=0,001$ применено полужирное начертание к тексту;
 при $\alpha=0,005$ применено курсивное начертание к тексту;
 при $\alpha=0,01$ применен полужирный курсив к начертанию текста.

Таблица 1 - Результаты экспериментов для разных объемов $n = 20, 40, 60$.

$\sigma(\mu)$	$t(20-2)$	$t(40-2)$	$t(60-2)$
0,17	2,2725895	2,83495661	<i>3,12970374</i>
0,2	2,24476097	<i>3,09896582</i>	<i>3,32599512</i>
0,24	2,58488622	<i>3,2671309</i>	<i>3,5031363</i>
0,26	2,74241378	4,00336224	3,98263886
0,32	2,90837104	3,87959043	3,95290855
0,33	<i>3,12032513</i>	3,92946553	4,04929184
0,35	4,14987472	3,99552639	4,15284213
0,37	3,95655019	3,95655019	4,15972527
0,38	3,92175671	3,92175671	4,16598949
0,4	4,14987472	3,93718749	4,23089348

В ходе исследования проведены эксперименты с целью определения влияния ряда значений параметров на изменения состояния сетевого трафика. В частности, исследовалось влияние следующих параметров (диапазоны изменения):

- интенсивность генерации заявок - λ и $\sigma(\lambda)$,
- интенсивность обслуживания заявок - μ и $\sigma(\mu)$,
- нагрузка системы - ρ ,
- объемы выборок - n ,
- моменты времени измерения характеристик - t ,
- уровни значимости - p .

Разработаны сценарии и проведены эксперименты на основе предложенной модели. Получены результаты в среде имитационного моделирования AnyLogic для различных условий проведения экспериментов, которые подтвердили факты наличия возмущений в моделируемом трафике.

Библиографический список

1. Скатков А.В., Шевченко В.И. Расширение референтной модели облачной вычислительной среды в концепции крупномасштабных научных исследований // Труды ИСП РАН. 2015. №6 С.285-306.
2. Грушин Д.А., Кузюрин Н.Н. Балансировка нагрузки в системе Unihub на основе предсказания поведения пользователей // Труды Института системного программирования РАН, 2015. Том 27. Вып. 5. С. 23–34.
3. Циперман Г.Н., Стохастическая модель процесса идентификации сервисов информационной системы / Г.Н.Циперман // Труды ИСП РАН, 2014. Том 26. Вып. 5. С. 7–28.
4. Ширяев А.Н. Вероятностно-статистические методы в теории принятия решений. 2-е изд., новое. / А.Н. Ширяев // М.: МЦНМО, 2014. 144 с.
5. Шелухин, О. И. Самоподобие и фракталы. Телекоммуникационные приложения / О. И. Шелухин, А. В. Осин, С. М. Смольский. М.: Физматлит, 2008. 237 с.
6. Крамер Г., Математические методы статистики. / Г.Крамер // М.: Издательство Мир, 1975. 648 с.

УДК 004.021

И.В. Макаров, студент 1-го курса магистратуры**Научный руководитель: О.В. Ченгарь, к.т.н., доцент***ФГАОУ ВО «Севастопольский государственный университет»***АНАЛИЗ МЕТОДОВ РАСПРЕДЕЛЕНИЯ РЕСУРСОВ В GRID СИСТЕМЕ**

Аннотация: в статье рассмотрены алгоритмы и методы распределения ресурсов в GRID-системах, их свойства и область применения с целью дальнейшего использования при построении моделей GRID-систем с различными параметрами, направленными на выполнение разнотипных задач.

Ключевые слова: GRID-система, эвристические алгоритмы, распределение вычислительных ресурсов

Annotation: the article discusses algorithms and methods of resource allocation in GRID-systems, properties, application areas and differences for the purpose of subsequent application in the construction of GRID-systems with various parameters aimed at performing various tasks.

Key words: GRID-system, heuristic algorithms, resources, distribution of computing resources

Для достижения высокого качества обслуживания и высокой пользовательской производительности Grid-систем существенную роль играет эффективная система планирования и распределения вычислительных ресурсов. Данная задача в Grid-системах представляет интерес как самостоятельное исследовательское направление, которому посвящено значительное количество научных статей публикаций [1, 2, 3, 4]. В то же время, известно, что несмотря на существенное продвижение в области исследования методов и моделей оптимизации планировщиков Grid, задачи планирования относятся к классу NP-полных и точного решения в общем случае не имеют. Это и определяет целесообразность разработки и исследования новейших эффективных подходов к вопросам планирования распределения ресурсов в Grid-системах.

Большое количество программного обеспечения (например, GLOBUS, UNICORE, Legion и Gridbus) были разработаны с целью обеспечения инфраструктуры, которая предоставит пользователям удаленный доступ к ресурсам через общую защищенную, масштабируемую всемирную сеть. Диаграмма рабочего процесса связана с автоматизацией процедур, посредством которых файлы передаются между участниками в соответствии с определенным набором правил для достижения конечного результата. Система управления рабочим процессом определяет, управляет и осуществляет рабочий процесс при помощи вычислительных ресурсов.

В целом, диаграммы рабочего процесса создаются при помощи средств моделирования рабочего процесса или генерируются автоматически с помощью информационных служб Grid, таких как MDS (Monitoring and Discovery Services) и VDS (Virtual Data System) до начала работы Grid. [5]

Большое количество эвристических методов легли в основу построения решений взаимозависимых задач планирования распределения ресурсов в однородных и специализированных кластерных средах. Несмотря на это научные исследования в этой области продолжаются, т.к. существует ряд проблем при планировании рабочего процесса функционирования приложений в Grid-среде, таких как:

- ресурсы распределяются по сетям Grid и многие пользователи вынуждены конкурировать за эти ресурсы;
- ресурсы не находятся под контролем планировщика;
- ресурсы являются гетерогенными и могут демонстрировать себя по-разному при выполнении одной и той же задачи;
- многие приложения для составления расписаний требуют больших объемов данных и перемещения больших объемов информации между источником и приемником.

Таким образом, моделирование рабочих процессов Grid необходимо для расчета неспецифицированных и гетерогенных рабочих сред. Также актуальным остается вопрос передачи больших объемов информации через различные каналы связи.

В рамках решения данных задач выбор метода распределения вычислительных ресурсов является одной из важнейших проблем при реализации Grid. Вследствие этого необходимо организовать планирование и распределение ресурсов с целью минимизации времени решения всего потока задач и времени простоя заявок в очереди.

Таким образом, целью данной работы является анализ большого количества методов планирования и возможности их комбинированного использования при различных состояниях системы в различные моменты времени.[6]

Условно рассматриваемые методы можно распределить по некоторым уровням иерархии, представленным на рисунке 1.

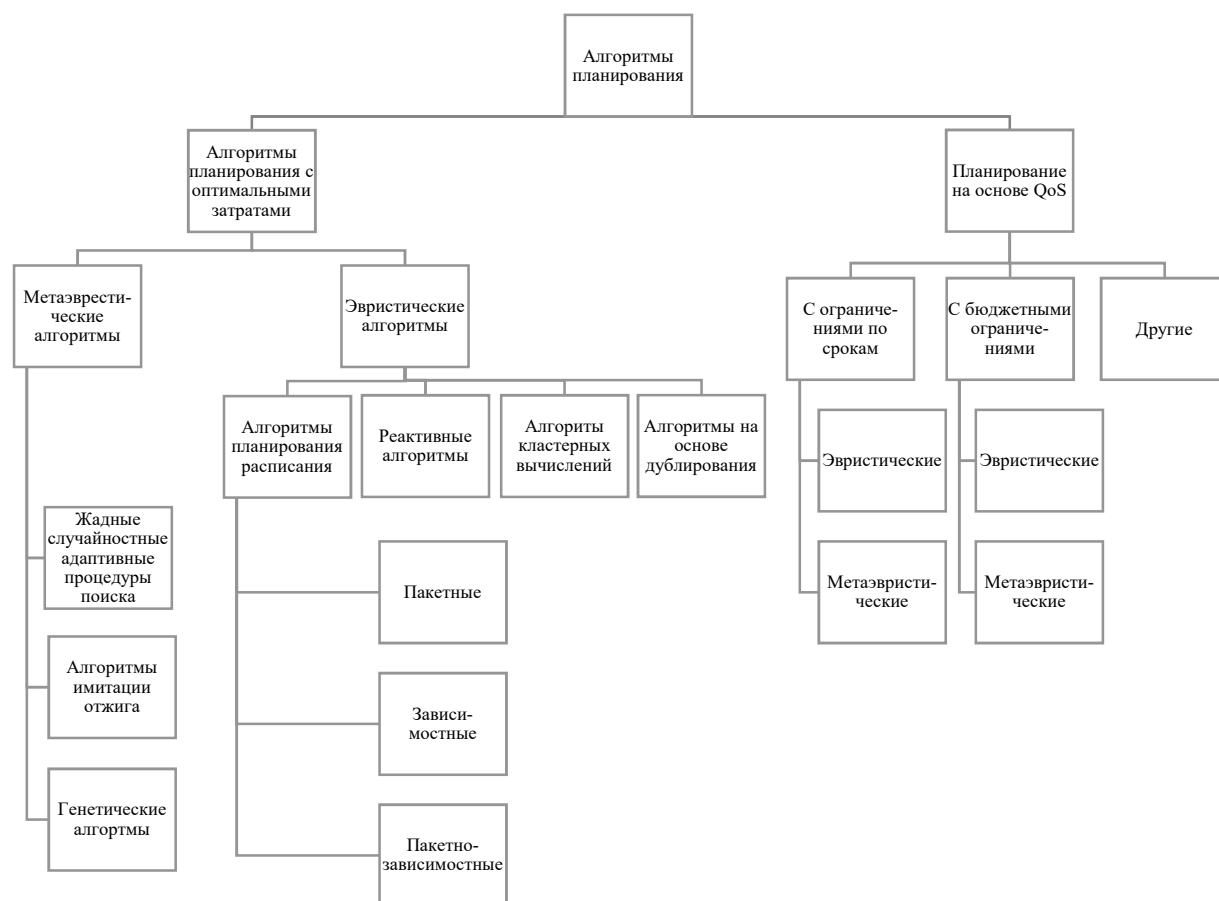


Рис 1 - Систематика алгоритмов планирования рабочего потока в GRID

Алгоритмы планирования рабочего процесса на основе оптимальных затрат ориентированы на среды GRID, в которых ресурсы распределяются между различными пользователями на основе модели сообщества (также известной как GRID-сообщество). При распределении ресурсов на основе модели сообщества денежные затраты не учитываются при доступе к ресурсам. Алгоритмы планирования рабочего процесса, основанные на лучших результатах, характерны завершением выполнения задач в кратчайшие сроки при минимизации времени функционирования приложения. Время функционирования приложения – это время, затрачиваемое с момента запуска приложения и до тех пор, пока все выходные данные не будут доступны пользователю [7].

В целом, алгоритмы планирования на основе оптимальных затрат делятся на эвристические и метаэвристические. Эвристический подход основан на разработке алгоритма планирования, который подходит только к конкретному типу задач, в то время как метаэвристический подход заключается в разработке алгоритма, который обеспечивал бы общий метод решения для разработки приложений.

В основном, эвристические алгоритмы подразделяются на четыре группы, а именно: планирование отдельных задач, планирование списков, планирование на основе кластеров и на основе дубликатов. Планирование отдельных задач – это самый простой способ планирования рабочих процессов, и он принимает решения по составлению расписания на основе отдельной задачи. Близорукий алгоритм [8] был реализован в некоторых GRID-системах, например в Condor DAGMan [3]. Алгоритм назначает невыполненную задачу ресурсу, который, как ожидается, завершит задание раньше всех, пока не будут поставлены в план все задачи.

Эвристика планирования расписания приоритезирует задачи основного процесса и рассматривает задачи, основанные на приоритетах. Существуют две основные фазы в эвристике планирования расписания: этап приоритезации задач и этап выбора ресурсов. Фаза установления приоритета задачи устанавливает приоритет каждой задачи с соответствующим значением ранга и формирует список расписаний путем сортировки задач в соответствии с их рангом [9]. Фаза выбора ресурсов выбирает задачи в порядке их приоритетов и ставит в соответствие оптимальному ресурсу. Различные эвристики планирования расписаний используют различные атрибуты и стратегии для определения приоритетов задач и оптимальных ресурсов для каждой задачи.

Пакетные алгоритмы составления расписания предназначены для обеспечения стратегии упорядочения и сопоставления исключительно параллельных задач на ресурсах, с целью завершения этих задач в кратчайшие сроки.

Зависимостные алгоритмы планирования расписаний созданы с целью обеспечить стратегию упорядочения и сопоставления задач основного процесса на гетерогенных ресурсах на основе анализа зависимостей всего графика задач с целью завершения взаимозависимых задач в кратчайшие сроки. В работе [4] также был предложен гибридный алгоритм, объединяющий предыдущие два метода.

Алгоритмы на основе кластеров и на основе дублирования предназначены для того, чтобы избавиться от времени передачи результатов между взаимозависимыми задачами. Кластерное планирование кластеризует и назначает задачи в один и тот же ресурс, тогда как планирование на основе дублирования использует время простоя ресурса для дублирования некоторых родительских задач, которые тоже планируются на других ресурсах.[2]

Метаэвристические методы обеспечивают как общую структуру, так и стратегические рекомендации для эвристики для решения вычислительных задач. Обычно, они применяются к большим и сложным системам, обеспечивая эффективный способ быстрого перехода к наиболее оптимальному решению.

Жадные случайностные адаптивные процедуры поиска (Greedy Randomized Adaptive Search Procedure - GRASP) представляют собой итеративные методы рандомизированного поиска. В работе [5] показано, что данные алгоритмы способны эффективно решать задачи планирования размещения рабочих мест. Генетические алгоритмы обеспечивают надежные методы поиска, позволяющие получать высококачественные решения из большого пространства поиска за полиномиальное время. Алгоритмы имитации отжига основаны на методе Монте Карло для статистического поиска. Понятие первоначально связано с тем, как кристаллические структуры могут быть сформированы в более упорядоченное состояние с использованием процесса отжига.

Многие приложения для построения расписаний требуют некоторых гарантий качества услуг (QoS). Для таких приложений требуется планирование рабочего процесса, чтобы иметь возможность анализировать требования пользователей к QoS и устанавливать соответствие между процессами с целью выполнения процессов и соответствия ограничениям QoS.[1]

Планирование с ограничениями по срокам рассчитано на то, чтобы приложения доставляли результаты до конечного срока. Различие между ограниченным сроком планирования и планированием с оптимальными затратами заключается в том, что при ограничении сроков планирования также необходимо учитывать денежные затраты при

планировании задач. Планирование с ограниченным бюджетом предполагает минимизацию времени выполнения рабочего процесса при удовлетворении заданных бюджетов пользователей.

Далее рассмотрим пример экспериментальных сравнений для алгоритмов планирования рабочего процесса. Сравним эвристический метод с ограничениями по срокам, представленный ранее. Для симуляции использовалась среда GridSim, в которой необходимые модели находятся в свободном доступе в GridSim Index Service (GIS). [6]

Сравним планирование с ограничениями по срокам (Time Distribution, TD) с алгоритмом поиска с возвратом (Backtracking, BT) в двух моделях потока операций – сбалансированном и несбалансированном, с точки зрения ограничения по сроку и по стоимости. Чтобы показать результаты более ясно – нормализуем время выполнения и стоимость (рис. 2). В случае с оценкой по времени выполнения, нормированное значение не должно превышать единицу. В случае с ограничением по стоимости, явных пределов нет. Очевидно, что как GA, так и TD, недостаточно надежны для достижения конечного срока в одиночку. Однако результаты улучшаются при объединении GA и TD путем помещения решения, созданного TD, в исходную популяцию GA. Как показано на графике, значение GA + TD намного ниже, чем работа отдельно взятых алгоритмов, что соответствует лучшим показателям достижения оптимальной с точки зрения соответствующих ограничений, нежели у GA и TD отдельно. В целом, комбинация алгоритмов показывает гораздо более оптимальные результаты, нежели отдельно взятые [3].

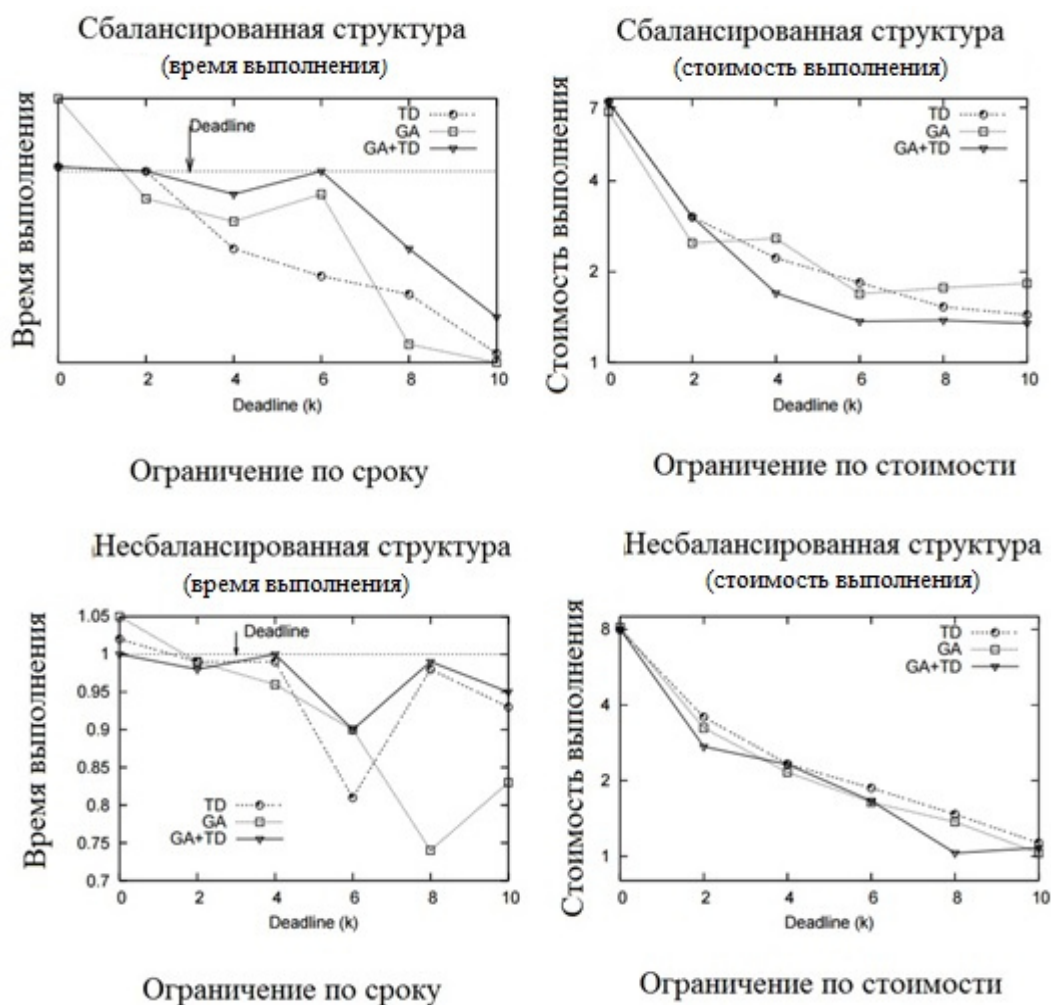


Рис. 2 - Сравнение планирования с ограничениями по сроку и по стоимости для алгоритмов TD и BT

В данной работе представлен обзор и анализ алгоритмов планирования в GRID-системах и составлена их классификация. Рассмотрены несколько моделей распределения ресурсов GRID-систем на основе этих алгоритмов, изучены результаты их работы. Также представлено несколько алгоритмов на основе эвристики и метаэвристики, позволяющие оптимизировать время выполнения рабочих процессов в GRID-системах. Также было детально изучено сопоставление этих алгоритмов с точки зрения вычисления времени выполнения приложений и сценариев распределения ресурсов. По причине того, что модель предоставления услуг в GRID-сетях основывается на «идеальных» условиях, то качество обслуживания и доступность услуг не могут быть гарантированы. Поэтому были приведены алгоритмы планирования в динамических средах GRID.

Также подтверждено утверждение о том, что не существует универсального по эффективности метода, позволяющего осуществить распределение доступных ресурсов в различной их совокупности для любого класса задач, в связи с чем подтверждается актуальность заявленной проблемы и необходимость дальнейшего исследования в этом направлении.

Библиографический список

1. Ефимов Сергей Николаевич, Тынченко Вадим Сергеевич Модели и алгоритмы формирования GRID-систем для структурно-параметрического синтеза нейросетевых моделей // Вестник СибГАУ. 2008. №4 (21). URL: <http://cyberleninka.ru/article/n/modeli-i-algoritmy-formirovaniya-grid-sistem-dlya-strukturno-parametriceskogo-sinteza-neyrosetevykh-modeley>
2. Кафтанников Игорь Леопольдович, Ханкин Константин Михайлович Простая модель GRID-системы для оценки влияния алгоритма диспетчеризации задач на энергопотребление // Вестник ЮУрГУ. Серия: Компьютерные технологии, управление, радиоэлектроника. 2013. №2. URL: <http://cyberleninka.ru/article/n/prostaya-model-grid-sistemy-dlya-otsenki-vliyaniya-algoritma-dispatcherizatsii-zadach-na-energopotreblenie>
3. Cantu-Paz, E. Designing scalable multi-population parallel genetic algorithms / E. Cantu-Paz // I11GAL Report 98009 /The University of Illinois. 1998. P. 82-122.
4. Forti A. DAG Scheduling for grid computing systems / A. Forti // Ph.D. Thesis, University of Udine, Department of Mathematics and Computer Science. - Italy, 2005 - 2006. - P. 43 - 46, 52 - 55.
5. Tompkins M.F. Optimization techniques for task allocation and scheduling in distributed multi-agent operations / M.F Tompkins // Diss. Massachusetts Institute of Technology. - 2003. - P. 20 - 23.
6. Fangpeng D. Scheduling Algorithms for Grid Computing: State of the Art and Open Problems / D. Fangpeng, Akl.G. Selim // School of Computing, Queen's University Kingston, Ontario, Technical Report. - 2006. - N 504. - P. 7 - 32.
7. H. Topcuoglu, S. Hariri, and M. Y. Wu. Performance-Effective and LowComplexity Task Scheduling for Heterogeneous Computing, IEEE Transactions on Parallel and Distributed Systems, 13(3): 260-274, March 2002.
8. L. Wang et al., Task Mapping and Scheduling in Heterogeneous Computing Environments Using a Genetic-Algorithm-Based Approach, Journal of Parallel and Distributed Computing, 47:8-22, 1997.
9. J. Yu and R. Buyya, A Taxonomy of Workflow Management Systems for Grid Computing, Journal of Grid Computing, Springer, 3(3-4): 171-200, Spring Science+Business Media B.V., New York, USA, Sept. 2005.

УДК 004.94:004.75

О.А. Пасик, магистрант**Научный руководитель: Е. Н. Машенко, канд. техн. наук, доц.***Севастопольский государственный университет***АНАЛИЗ СТРАТЕГИЙ ВЫБОРА ПРОВАЙДЕРА ОБЛАЧНЫХ IaaS УСЛУГ**

Аннотация: Облачные вычисления и, в частности, модель «инфраструктура как услуга», позволяют компаниям, не специализирующиеся на ИТ, освободиться от сопровождения и развития собственного ИТ-отдела, что позволит сконцентрировать основные усилия и инвестиции на развитии основного рода деятельности компании. Однако существует немало рисков при использовании облачных услуг и в первую очередь – это риски при выборе провайдера. Чтобы избавить пользователя от данного рода рисков была создана такая услуга как облачный брокеринг. В данной работе предлагается имитационная модель процесса взаимодействия облачного потребителя с облачными провайдерами при посредничестве облачного брокера.

Ключевые слова: облачный брокер, облачный потребитель, облачный провайдер, IaaS.

Annotation: Cloud computing and, particularly, Infrastructure as a Service model, allow companies, that do not specialize in IT, to be released from the support and development of their own IT department, which allows them to concentrate their efforts and investments on the development of the company's core business. However, there are a lot of risks when using cloud services and, first of all, these are risks of choosing a supplier. To save users from such a danger, cloud broking service was created. In this paper, we propose an imitation model of the process of interaction between the cloud consumer and cloud providers with the intermediation of a cloud broker.

Key words: cloud broker, cloud consumer, cloud supplier, IaaS.

Введение

Сегодня под облачными вычислениями понимают возможность получения необходимых вычислительных мощностей по запросу из сети, причем пользователю не важны детали реализации этого механизма и он получает из "облака" все необходимое [1].

Облачные вычисления позволяют организовать динамическое предоставление услуг, когда пользователи могут производить оплату по факту и регулировать объем своих ресурсов в зависимости от реальных потребностей без долгосрочных обязательств.

Благодаря облачным вычислениям компании могут предоставлять конечным пользователям удаленный динамический доступ к услугам, вычислительным ресурсам и приложениям (включая операционные системы и инфраструктуру) через Интернет. Вычислительные облака состоят из тысяч серверов, размещенных в физических и виртуальных центрах обработки данных, обеспечивающих работу десятков тысяч приложений, которые одновременно используют миллионы пользователей. По сравнению с традиционным подходом, облачные вычисления позволяют управлять более крупными инфраструктурами и обслуживать различные группы пользователей в пределах одного облака.[2]

Для понимания работы облачных сервисов принято выделять три основных модели предоставления сервисов: *IaaS*, *SaaS* и *PaaS*.

1. Модель IaaS

IaaS расшифровывается как *Infrastructure as a Service* – инфраструктура как услуга, то есть потребление серверной инфраструктуры в модели аренды. Основная идея такого подхода заключается в том, чтобы ИТ инфраструктуру, состоящую из вычислительных серверов, оборудования и сети снять с собственной реализации и поддержки силами ИТ отдела заказчика, и перенести на реализацию и поддержку силами облачных провайдеров [2].

Преимущество использования такой модели заключается в том, что *IaaS* услуги позволяет компаниям, не специализирующиеся на ИТ, освободиться от сопровождения и развития собственного ИТ-отдела, что позволит сконцентрировать основные усилия и инвестиции на развитии основного рода деятельности компании.

Важно понимать, что использование услуг узко специализированных ИТ компаний гораздо эффективнее и качественнее, чем создание и поддержка собственного отдела. Попытки улучшить собственную ИТ-инфраструктуру путем закупки более качественного, а значит и более дорогостоящего оборудования и найма квалифицированных специалистов

приведет только к дополнительным затратам денег и времени, что безусловно навредит компании и отдалит от изначальной бизнес цели.

Поэтому, в первую очередь, нужно освободить IT-отдел компании от сопровождения самых затратных ресурсов, таких, как:

- операционная система;
- база данных;
- серверное оборудование;
- система хранения данных.

2. Риски выбора провайдера

Однако, на сегодняшний день существуют определенные риски при использовании *IaaS* услуг. В первую очередь – это риски при выборе провайдера.

Например, плохо проработанный проект миграции сервисов в облако может привести к простою бизнеса превышающему плановое время простоя. Кроме этого при возникновении проблем в рамках такого проекта возможна частичная потеря данных, неработоспособность перенесенных в облако сервисов, а также общие проектные риски (срыв плановых сроков и показателей, превышение бюджета при некорректной оценке требуемого объема ресурсов и пр.). При отсутствии комплексного и всестороннего тестирования работы приложения в облаке возможна, например, негативная реакция пользователей на скорость отклика приложения, а также непредвиденные сложности в выполнении необходимых операций, например, отсутствие поддержки прозрачной аутентификации при выбранной схеме подключения [2].

Также к рискам стоит отнести случаи, когда провайдеры попросту оказываются недобросовестными или некомпетентными. К таким случаям относятся: нарушение SLA (Service Level Agreement - соглашение об уровне обслуживания), занижение параметров качества не зафиксированных в SLA, прекращение деятельности провайдера.

Все это может привести к незапланированным расходам, сложностям, связанным с поиском нового провайдера и деградации бизнеса.

3. Облачный брокер

Чтобы избежать вышеуказанных рисков, можно воспользоваться услугами облачного брокера.

Согласно NIST, облачный брокер - это сущность, управляющая использованием, производительностью и предоставлением облачных услуг, а также устанавливающая отношения между Облачными Провайдерами и Облачными Потребителями [3].

Брокером является предприниматель или компания, которая представляет собой третью сторону между покупателем и поставщиком облачных услуг.

Роль брокера заключается в том, чтобы экономить время покупателя облачных сервисов на поиске необходимых ему услуг у различных провайдеров и предоставить клиенту информацию о том, какие сервисы помогут достигнуть бизнес целей клиента.

Согласно сценарию, брокер работает с клиентами, чтобы понять рабочий процесс, потребности в ресурсах, составление бюджета и требований по управлению данными. После того, как исследование будет завершено, брокер представит клиенту короткий список рекомендуемых поставщиков облачных сервисов.

Облачному брокеру может также быть предоставлено право заключать договоры с провайдерами от имени клиента. В таком случае, брокер получает право на распределение услуг от нескольких провайдеров в стремлении быть максимально экономически эффективным, насколько это возможно, несмотря на сложности, которые могут возникать при переговорах с несколькими поставщиками облачных услуг одновременно. Участие облачного брокера во взаимодействии потребителя с провайдером согласно NIST изображено на рис. 1.

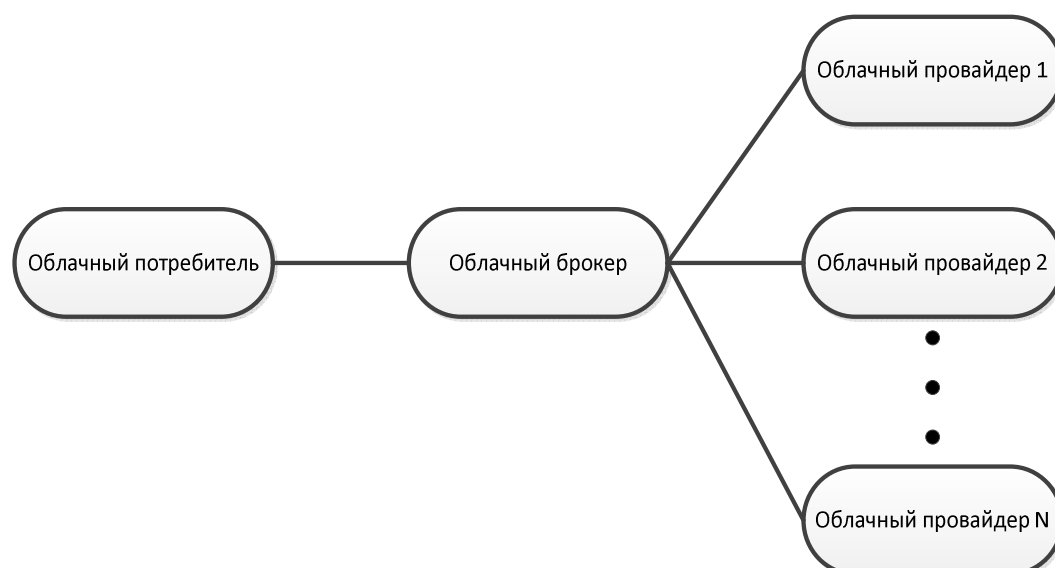


Рис. 1 Участие облачного брокера во взаимодействии потребителя с провайдером согласно NIST

4. Модель брокера

В ходе исследований в среде AnyLogic 7.3 была построена имитационная модель. Модель демонстрирует взаимодействие таких сущностей, как: облачный потребитель, облачный брокер и облачный провайдер. Задачей брокера является распределение клиентов по провайдерам в зависимости от их требований и доступным ресурсам.

В качестве основных агентов модели исследуются агенты типа *Clients*, *Broker* и *Providers*, которые олицетворяют облачных потребителей, облачного брокера и облачных провайдеров соответственно.

Агенты типа *Clients* имеют четыре параметра, которые являются требованиями клиентов:

CPU – количество ядер процессора;

RAM – объем оперативной памяти Гб;

HDD – объем жестких дисков Гб;

Prov_num – идентификатор провайдера, который определяется брокером.

Также тип агента *Clients* является базовым типом для двух других типов агентов: *smallClients* и *bigClients*. Агенты *smallClients* соответствуют такому типу клиентов, которым нужно относительно небольшое количество ресурсов провайдера. Это могут быть фирмы-представители малого и среднего бизнеса. Агенты *bigClients* – это большие фирмы, потребность в вычислительных ресурсах которых гораздо выше, чем у клиентов, описанных выше. Предполагается, что интенсивность поступления заказов услуг и другие параметры у данных типов клиентов разные, при том, что все они обладают одинаковой динамикой поведения и некоторыми общими свойствами. Источники данных типов агентов *enter* и *enter1* изображены на рисунке 2.

Агент *Providers* содержит в себе пять провайдеров, выполненных с помощью дискретно-событийного моделирования. Задачей каждого провайдера является обеспечение поступивших клиентов ресурсами. Каждый провайдер имеет три типа ресурсов *CPU*, *RAM*, *HDD*. Ресурсы провайдеров изображены на рисунке 3. Объекты последовательного захвата ресурсов каждого провайдера изображены на рисунке 4.

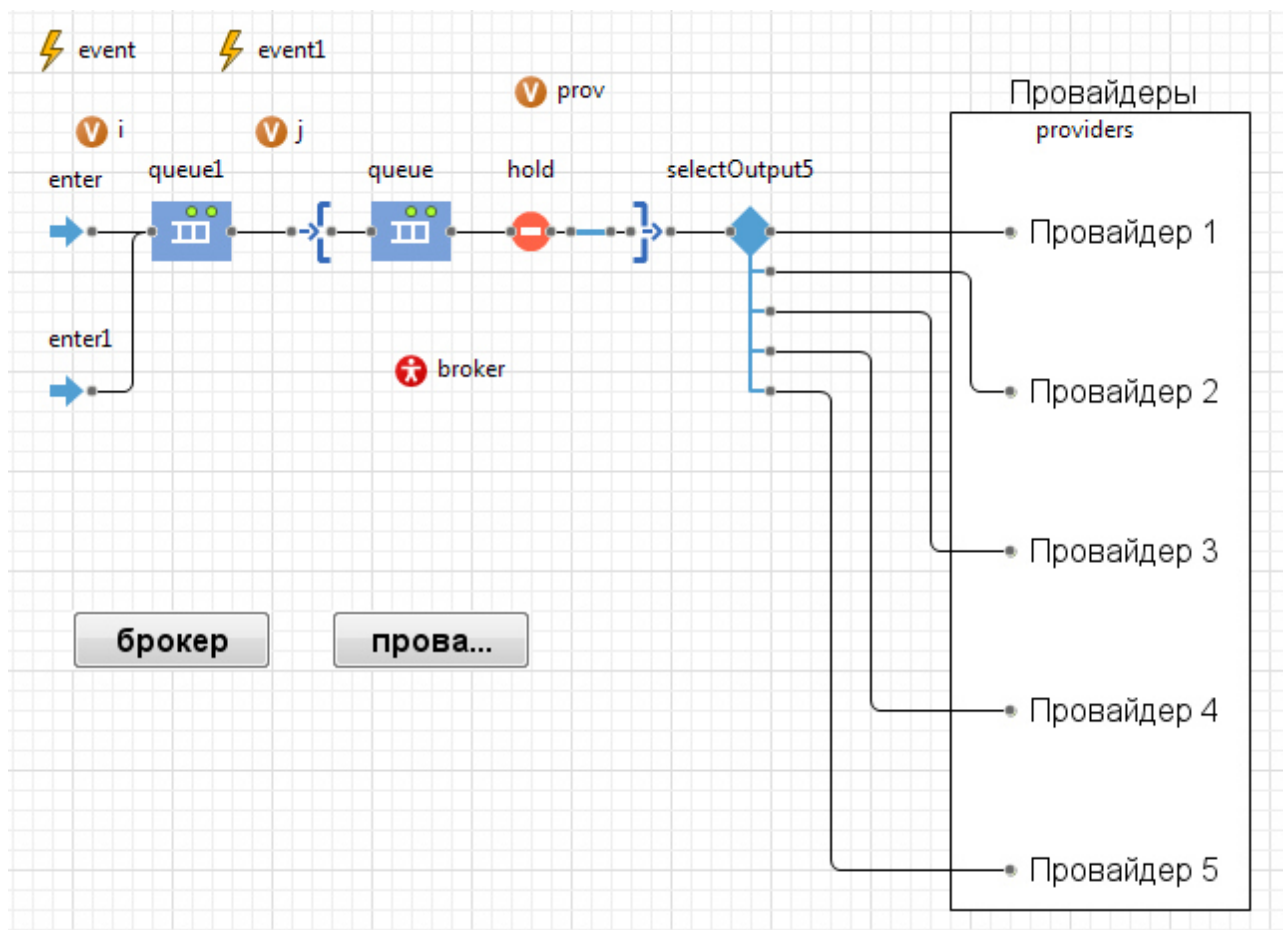


Рис. 2 Поступление заявок клиентов

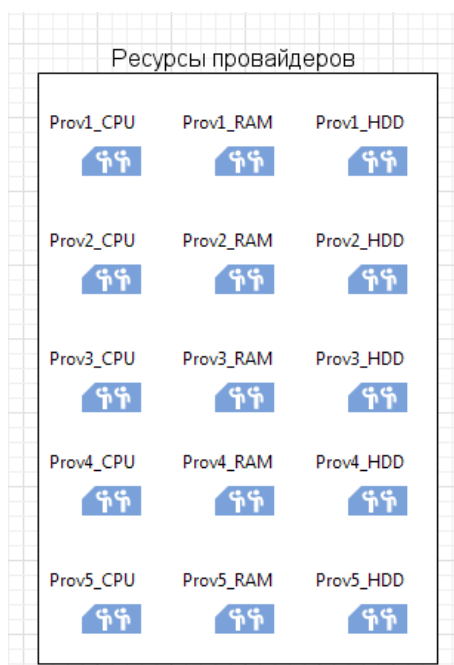


Рис. 3 Ресурсы провайдеров

Основная функция агента *Broker* – выбрать для клиента провайдер, ресурсов которого будет достаточно для того, чтобы клиент мог их арендовать.

Брокер поочередно сравнивает требования клиента с ресурсами провайдеров до тех пор, пока не найдется провайдер, свободных ресурсов которого будет достаточно для удовлетворения требований клиента.

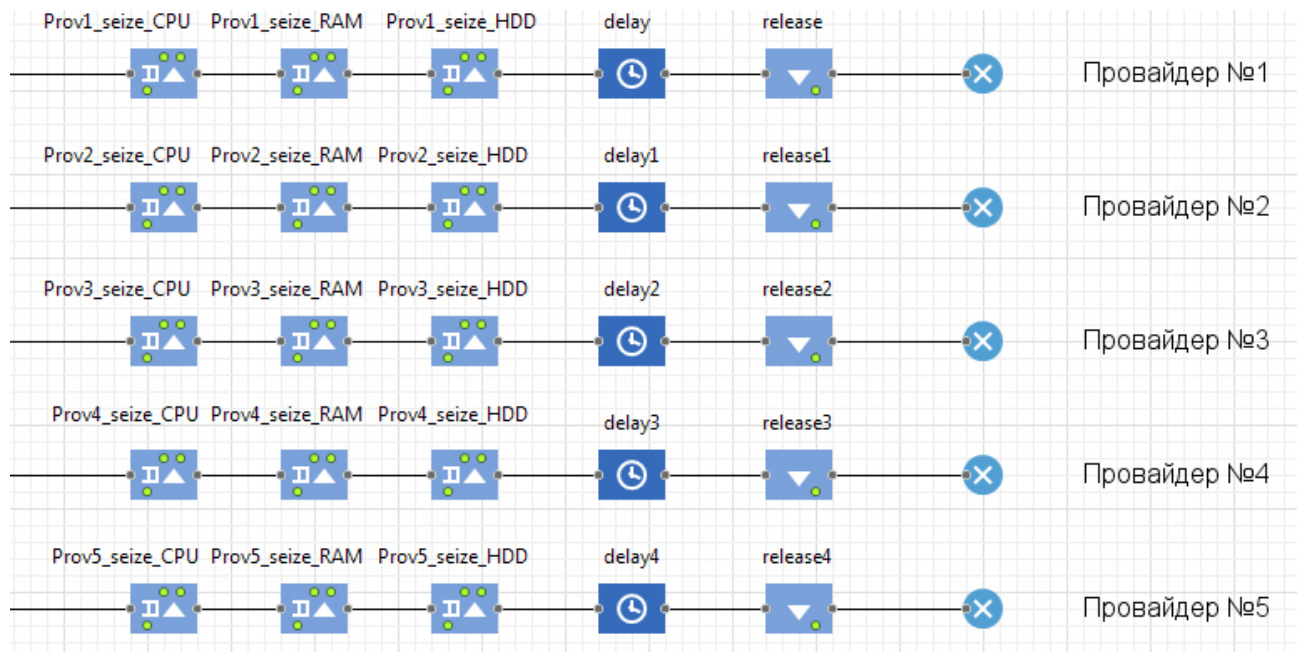


Рис. 4 – Все провайдеры из агента Providers

Выбор провайдера реализован с помощью элементов дискретно-событийного моделирования, изображенных на рисунке 5.

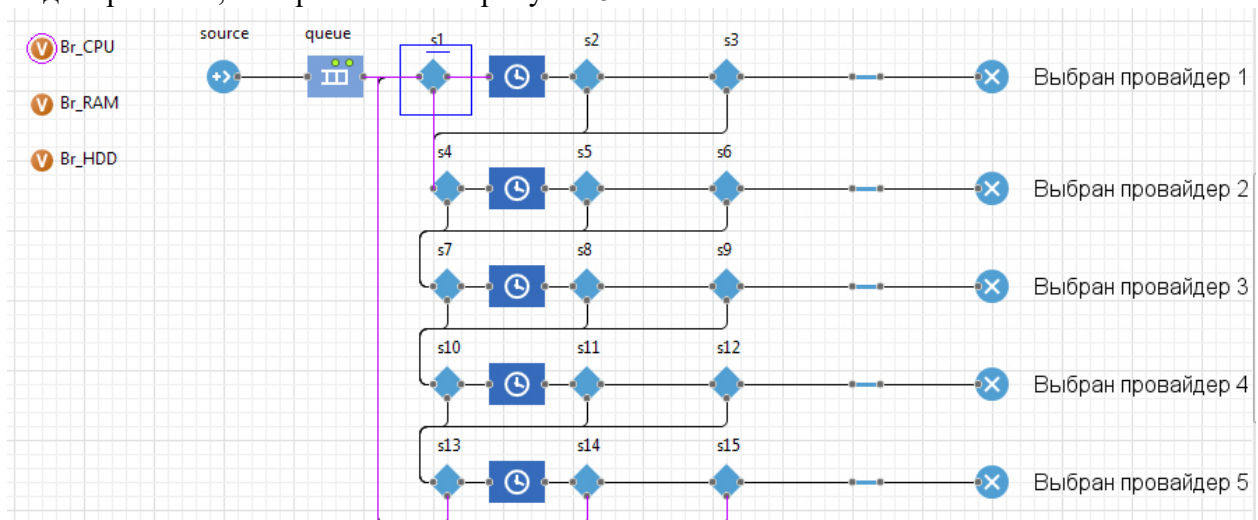


Рис. 5 – Выбор провайдера в агенте Broker

5. Эксперимент

Выполним эксперимент с моделью. Цель данного эксперимента – определить степень влияния брокера на увеличение количества обработанных заявок-клиентов. В качестве оценки результата будем использовать коэффициент обработки Φ .

$$\Phi = \frac{N}{M} \quad (1)$$

где N – количество обработанных заявок,
 M – количество поступивших заявок.

Вычислим данный коэффициент для модели без брокера. При этом вероятность выбора каждого из пяти провайдеров будет одинакова и равна 0.2. Параметры выбора провайдеров изображены на рисунке 6. Модельное время проведения эксперимента равно двум месяцам или 60 дням. Результаты моделирования изображены на рисунке 7.

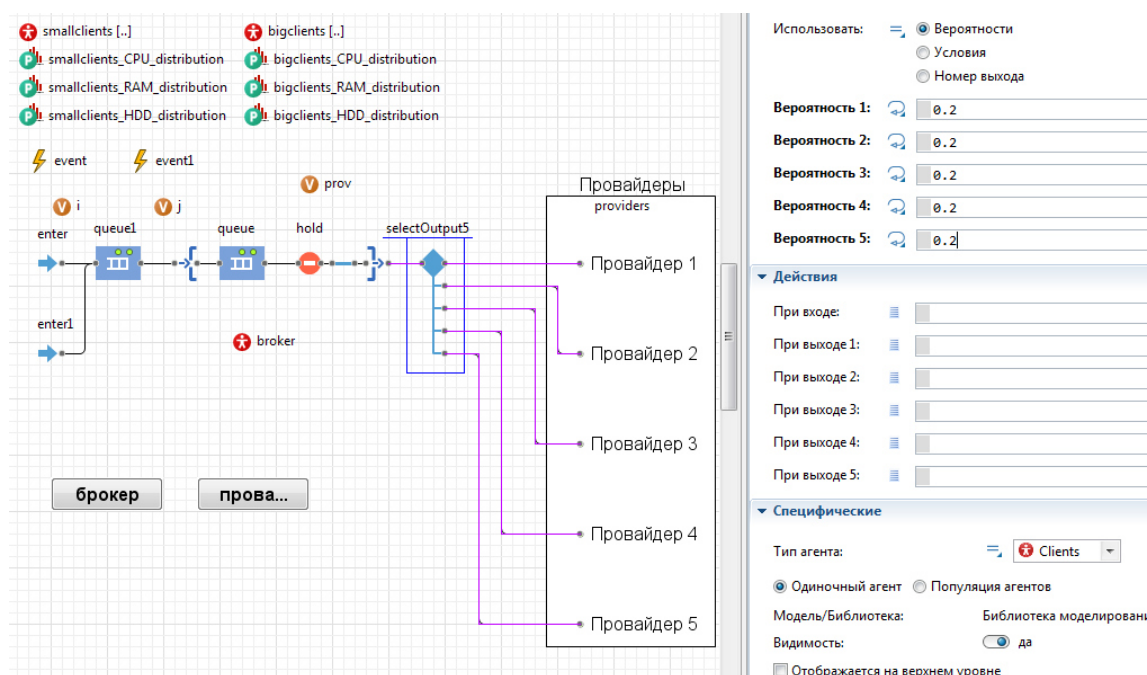


Рис. 6 – Параметры выбора провайдеров без участия брокера



Рис. 7 – Результат работы модели без брокера

Количество обработанных заявок равно 55 из 95 поступивших в очередь на обработку. Коэффициент Φ соответственно равен 0.579.

Далее вычислим данный коэффициент для модели с участием брокера. Количество выданных брокером решений изображено на рисунке 8. Модельное время проведения эксперимента также равно двум месяцам или 60 дням. Результаты моделирования изображены на рисунке 9.

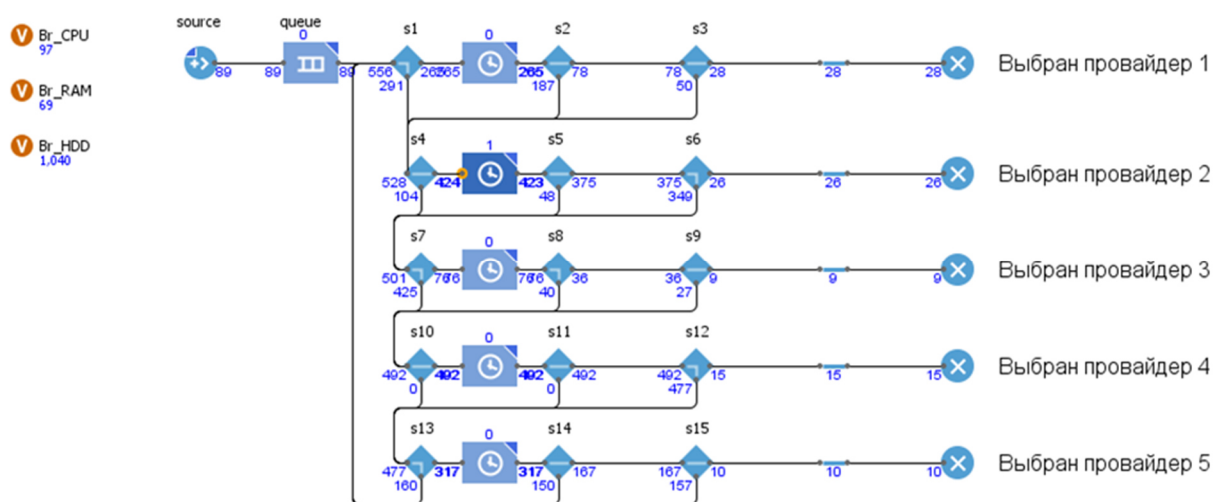


Рис. 8 – Количество выданных брокером решений



Рис. 9 – Результат работы модели с брокером

Количество обработанных заявок равно 72 из 99 поступивших в очередь на обработку. Коэффициент Φ соответственно равен 0.727.

Выводы

Облачные вычисления являются по настоящему революционной технологией, использование которой принесет множество преимуществ ее пользователям, в том числе значительную экономию как времени, так и бюджетных средств. Однако существует немало рисков при использовании облачных услуг. Для их устранения и был создан такой вид услуг, как облачный брокеринг. В результате проведенных исследований было выяснено, что использование услуг узко специализированных компаний в качестве брокеров при выборе облачного *IaaS* провайдера и миграции сервисов в облако может избавить пользователя от простоя бизнеса превышающего плановое время простоя, потери данных, неработоспособности перенесенных в облако сервисов, а также общепроектных рисков. Также брокер может помочь пользователю составить SLA и отслеживать его соблюдение провайдером.

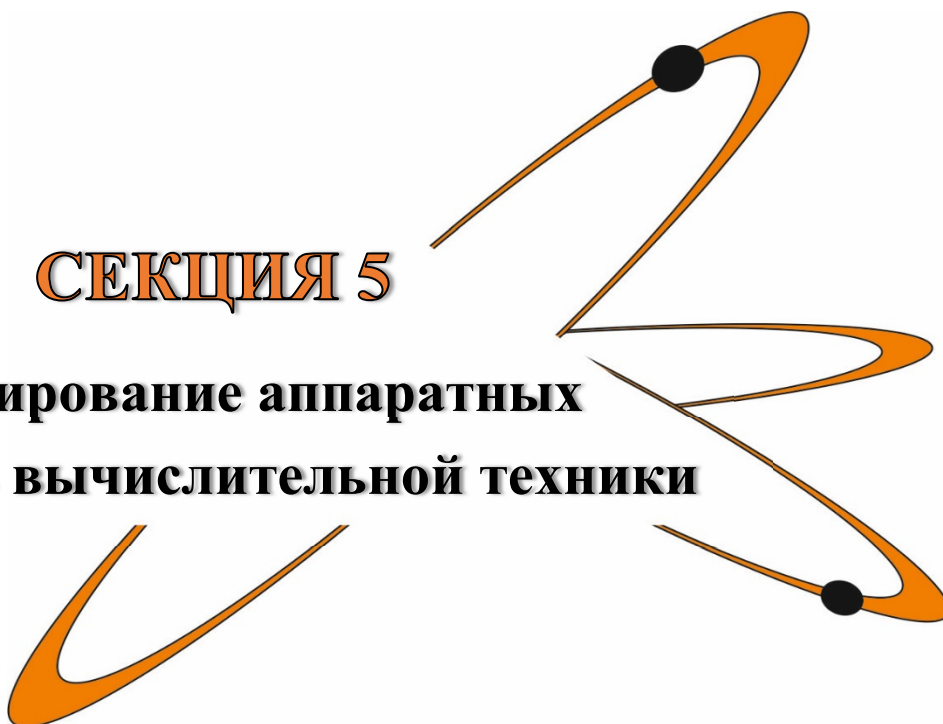
Была построена имитационная модель, демонстрирующая взаимодействие таких сущностей, как: облачный потребитель, облачный брокер и облачный провайдер. В результате проведенных экспериментов было получено, что при использовании брокера коэффициент обработки, показывающий отношение обработанных заявок ко всем поступившим заявкам увеличился практически на 50%.

Библиографический список

1. Клементьев И.П., Устинов В.А. - Введение в облачные вычисления (2-е изд.) - 2016. — 311 с.
2. Риз Джордж. Облачные вычисления: Пер. с англ. — СПб.: БХВ - Петербург, 2011. — 288 с.
3. 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>.

СЕКЦИЯ 5

Проектирование аппаратных средств вычислительной техники



УДК 681.3

В.А. Бойченко, магистрант**Научный руководитель: Д.В. Моисеев, к. т. н., доцент****Севастопольский государственный университет****ОЦЕНКА ТОЧНОСТИ И БЫСТРОДЕЙСТВИЯ ПРИ ВЕРОЯТНОСТНОЙ ФОРМЕ ПРЕДСТАВЛЕНИЯ ИНФОРМАЦИИ***Аннотация: В данной работе рассматривается оценка точности и быстродействия при использовании вероятностной формы представления информации.**Ключевые слова: вероятностная форма представления, точность, быстродействие.**Annotation: In this article viewed value of precision and operating speed using probabilistic form of data presentation**Key words: probabilistic form of data presentation, precision, operating speed***Постановка задачи**

Преобразование сигнала из любой формы в вероятностную основано на замене значения измеряемого сигнала соответствующей ему вероятностью [1].

Величина погрешности вероятностного преобразования зависит от вида закона распределения вспомогательного случайного сигнала $R(t)$ и обратно пропорциональна корню квадратному из числа независимых статистических испытаний K .

Абсолютная погрешность вероятностного преобразования будет равна [2]

$$\Delta = \sqrt{2\Phi^{-1}(P)}\sigma(x_i^*) = \frac{\sqrt{2\Phi^{-1}(P)}}{\sqrt{K}}\sqrt{F_{x_i}(R) - F_{x_i}^2(R)}, \quad (1)$$

где Φ^{-1} – функция, обратная функции Лапласа;

P – вероятность того, что истинное значение $F_{x_i}(R)$ принадлежит интервалу с границами $I_p = \{F_{x_i}^*(R) - \Delta; F_{x_i}^*(R) + \Delta\}$.

Нормирование в единичном диапазоне величины x_i , подвергнутой вероятностному преобразованию, приводит к возможности расчета величины приведённой погрешности по:

$$\gamma = \Delta 100\%. \quad (2)$$

Решение задачи

Для решения задачи определения ошибки при вероятностном преобразовании, в общем виде некоторой произвольной величины использование формул (1) и (2) показывает, что увеличение количества статистических испытаний уменьшает величину погрешности.

Для выявления количественных закономерностей рассмотрим случай, подчинения вспомогательного случайного сигнала $R(t)$ равномерному закону распределения [1, 2].

В этом случае выражение (1) для абсолютной погрешности примет вид:

$$\Delta = \sqrt{2\Phi^{-1}(P)}\sigma(x_i^*) = \frac{\sqrt{2\Phi^{-1}(P)}}{\sqrt{K}}\sqrt{x_i(1-x_i)}. \quad (3)$$

Как известно из [3] максимальная погрешность при вероятностном преобразовании возникает, когда значение преобразуемой величины x_i находится в середине динамического диапазона $[0..1]$, то есть при $x_i = 0,5$.

В этом случае оценкой сверху будет:

$$\Delta \leq 0,7\Phi^{-1}(P)\sqrt{K}. \quad (4)$$

В случае использования доверительного интервала равного $P = 0,9973$ имеем $0,7\Phi^{-1}(P) = 1,645$, отсюда абсолютная погрешность вероятностного преобразования равна

$$\Delta_{\max} \leq \frac{1,645}{\sqrt{K}}. \quad (5)$$

Согласно (3) приведённая погрешность вероятностного преобразования будет равна:

$$\gamma \leq \frac{1,645}{\sqrt{K}} \cdot 100\%. \quad (6)$$

В табл. 1 приведены рассчитанные значения величины γ , отображающие зависимость приведенной погрешности от числа статистических испытаний K при разных значениях величины параметра χ .

Таблица 1 – Значения приведенной погрешности при вероятностном преобразовании

Количество статистических испытаний K	Значение параметра χ				
	0,1	0,2	0,3	0,4	0,5
1	98.700	131.600	150.767	161.176	164.500
4	49.350	65.800	75.383	80.588	82.250
32	17.448	23.264	26.652	28.492	29.080
128	8.724	11.632	13.326	14.246	14.540
512	4.362	5.816	6.663	7.123	7.270

На рис. 1 приведены рассчитанные значения величины γ , отображающие зависимость приведенной погрешности от числа статистических испытаний K при разных значениях величины параметра χ из табл. 1.

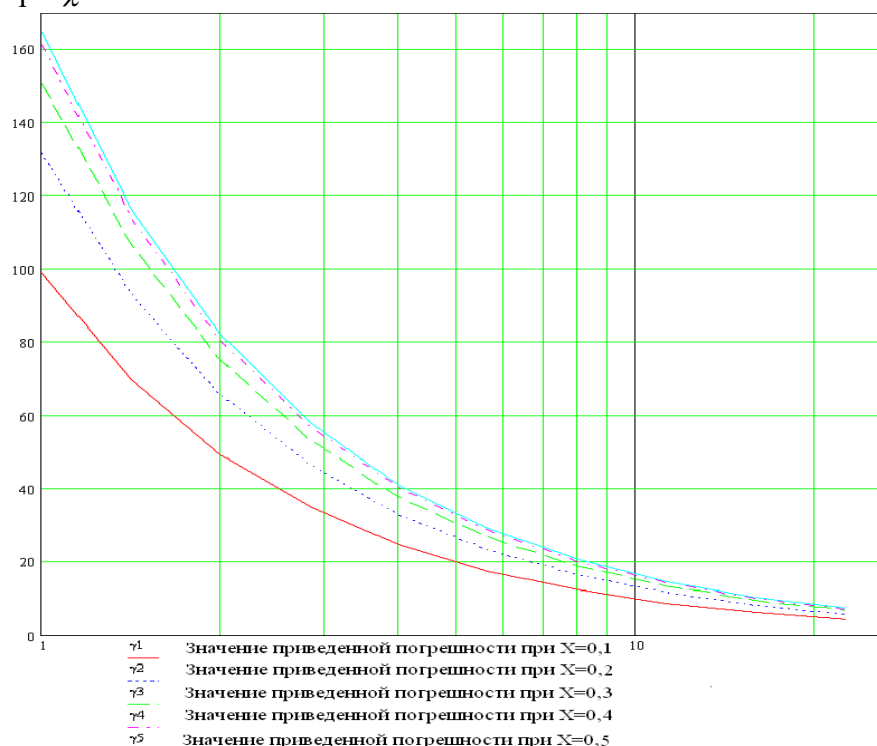


Рис. 1 – Значения величины γ , отображающие зависимость приведенной погрешности от числа статистических испытаний K при разных значениях величины параметра χ

Таким образом, вероятность того, что значение $F_{x_i}(R)$ попадает в доверительный интервал, как и ранее, принята равной $P=0,9973$. Из графиков видно, что приведенная погрешность вероятностного преобразования при равномерном распределении $R(t)$ меньше зависит от значения преобразуемого параметра x .

Как отмечалось выше, важнейшим свойством вероятностного преобразования, является возможность восстановления значения исходного параметра x_i из его вероятностного отображения y_i . Знание интегрального закона распределения вспомогательной случайной функции $R(t)$ позволяет путем функционального преобразования определить оценку параметра величины x_i по найденному значению оценки математического ожидания вероятностного отображения [2].

Выражая из формулы абсолютной погрешности количество испытаний K необходимое для достижения допустимой погрешности получаем:

$$[K] = \left(\frac{\sqrt{2\Phi^{-1}(P)}}{\sqrt{K}} \sqrt{x_i(1-x_i)} \right)^2. \quad (7)$$

Расчет количества испытаний K , при различном значении абсолютной погрешности Δ и значение преобразуемого параметра x_i приведены в табл. 2.

Анализируя выражения (1), (2) и (7) можно прийти к выводу о том, что количество источников вспомогательных сигналов в значительной степени зависит от заданной погрешности преобразования, а также от значения величины x_i и, при прочих равных условиях, является максимальным при $x_i = 0,5$.

Таким образом, становится очевидным основной недостаток вероятностной формы представления данных – обратная зависимость между точностью и быстродействием, что, в свою очередь, приводит к необходимости значительно сужать частотную область применения устройств для реализации заданной точности.

Решение данной проблемы возможно за счет применения параллельного вероятностного преобразования измеряемого сигнала [3].

Таблица 2 – Значения количество испытаний K необходимое для достижения допустимой погрешности при различном значении параметрах

Допустимая абсолютная погрешность, Δ	Значение параметра x				
	0,1	0,2	0,3	0,4	0,5
1%	2436	4330	5683	6495	6766
2%	609	1083	1421	1624	1692
3%	271	482	632	722	752
4%	153	271	356	406	423
5%	98	174	228	260	271
6%	68	121	158	181	188
10%	25	44	57	65	68

Выводы

Всё рассмотренное выше позволяет сделать вывод о том, что при представлении информации в ИИС в виде вероятностных отображений, при проектировании последних следует учитывать обратную зависимость между точностью и быстродействием. Наиболее целесообразно применять вероятностную форму представления данных в ИИС, в случаях, когда решающими требованиями к аппаратному комплексу являются высокая живучесть и надёжность, малые затраты оборудования, полная автоматизация измерительно-вычислительного процесса при ограниченных требованиях к быстродействию.

Указанным требованиям отвечает широкий класс специальных информационно-измерительных систем, применяемых в энергетике, при осуществлении радиационного, газового и экологического контроля.

Библиографический список

1. Моисеев Д.В. Преимущества первичных преобразователей «аналог- вероятность» // Д.В. Моисеев Сбірник наукових праць Академії військово-морських сил імені П.С. Нахімова. – Севастополь: АМВС імені П.С. Нахімова, 2011. – Вип. 1 (5). С 115-120.
2. Сапожников Н.Е. / Оценка точности и быстродействия при вероятностной форме представления информации // Н.Е. Сапожников, Д.В. Моисеев, Ю.Ю. Столярчук. Збірник наукових праць СНУЯЕтаП. – Севастополь: СНУЯЕтаП, 2011. – Вип. 3 (39) С. 134-140.
3. Сапожников Н.Е. / Выполнение параллельных вычислений при вероятностном представлении информации // Н.Е. Сапожников, Д.В. Моисеев, П.С. Бейнер, Н.В. Бейнер. Technology audit and production reserves — № 3/1(11), 2013, С – 9–12.

УДК 004.3

В.А. Бойченко, магистрант**Научный руководитель: Д.В. Моисеев, к. т. н., доцент***Севастопольский государственный университет***ПРИМЕНЕНИЕ СПЕЦИАЛИЗИРОВАННЫХ ВЕРОЯТНОСТНЫХ ПРОЦЕССОРОВ В ИНФОРМАЦИОННО-ИЗМЕРИТЕЛЬНЫХ СИСТЕМАХ***Аннотация: В данной статье предлагается применение информационно-измерительной системы на основе специализированных вероятностных процессоров на объектах АЭС.**Ключевые слова: вероятностная процессоры, информационно-измерительная системы, надежность, производительность.**Annotation: In this article proposed using information-measuring system, which based on specialized probabilistic processors at NPP**Key words: probabilistic processors, information-measuring system, reliability, efficiency***Постановка вопроса**

Временной ресурс большинства основных объектов атомной энергетики либо исчерпан, либо близок к этому, поэтому в настоящее время выполняется большое количество работ связанных с исследованием возможности и продлением этих сроков. Обязательным условием функционирования означенных объектов после продления срока эксплуатации является мониторинг технического состояния объектов, суть которого заключается в непрерывном и систематическом сборе и обработке информации предназначенной для поддержки принятия правильных управленческих решений операторским персоналом по эксплуатации оборудования. Всё это приводит к лавинообразному росту количества первичной информации, поступающей на централизованный блок управления и её обработке по заданным алгоритмам в реальном масштабе времени.

С другой стороны, одним из существенных факторов, определяющих надежность работы объектов атомной энергетики, является оптимизация методов диагностики этих объектов и обработки информационных потоков с выходов первичных измерительных преобразователей. Как известно, техническая диагностика является составной частью технического обслуживания и её основной задачей является сокращение затрат на техническое обслуживание объектов и на уменьшение потерь от простоя в результате отказов. С увеличением сложности объектов диагностики количество информации необходимое для правильного диагностирования всё более возрастает.

Решение

Для решения указанных задач на АЭС предлагается применение информационно-измерительной системы на основе специализированных вероятностных процессоров, структура которой показана на рис.1:

1) устройства первичной обработки информации, территориально совмещённые с первичными информационными преобразователями (датчиками), основной задачей которых является преобразование и обработка исходной информации в реальном масштабе времени с целью сокращения избыточности информации, передаваемой по каналу связи в центральный пункт управления;

2) помехоустойчивые линии связи, в которых кроме передачи информации осуществляется её защита от несанкционированного доступа;

3) устройства вторичной обработки информации в реальном масштабе времени по алгоритмам, позволяющим на основе математического моделирования реальных процессов, осуществлять поддержку принятия эффективных и главное, безопасных, управленческих решений.

Анализ требований, предъявляемых к первой подсистеме, показывает, что основными характеристиками для неё являются малый аппаратный объём, веса и габариты, радиационная

стойкость, представление информации в дискретной форме, требуемая точность и достоверность, невысокий частотный (до 100 кГц) диапазон.

Вторая, телекоммуникационная подсистема должна отвечать требованиям помехоустойчивости, защиты информации, а также единого интерфейса с первой и третьей подсистемами.

Для третьей подсистемы должна быть реализована возможность распараллеливания решаемых в реальном масштабе времени задач, реализация, модернизация и использование аналитических и имитационных математических моделей с возможностью сравнения полученных результатов. Эта подсистема обеспечивает высокие надёжность, живучесть и помехоустойчивость, удобное для оперативного персонала отображение полученных результатов.

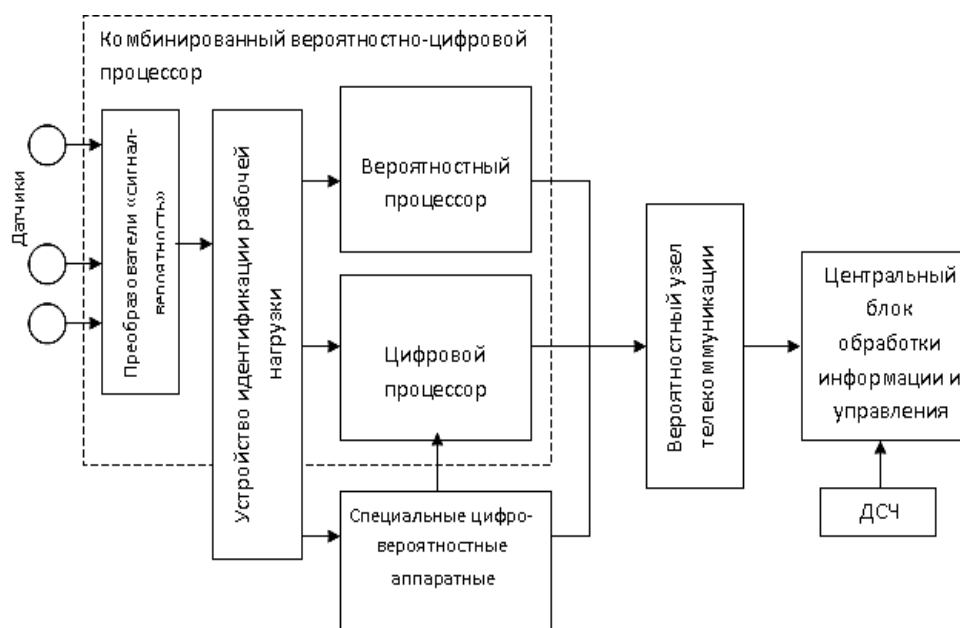


Рис. 1 – Структура системы обработки информации, основанной на вероятностной логике

Для комплексного удовлетворения требованиям, предъявляемым к первой подсистеме целесообразно воспользоваться устройствами обработки информации, представленной в цифровой и вероятностной [1] формах, путём реализации специализированных вероятностных аппаратных и использования радиационно-стойких микроконтроллерных средств. Это позволит создавать измерительно-управляющие подсистемы первого уровня с комбинированными программируемыми вероятностно-цифровыми процессорами под управлением универсального интеллектуального блока обработки информации и специализированные цифро-вероятностные аппаратные комплексы для решения ограниченного набора задач (например: статистического анализа), обладающие малым аппаратным объемом, высокой надёжностью и помехозащищённостью, способные работать в режиме реального времени. В такой подсистеме первичная информация, представленная в любой возможной форме, преобразуется в вероятностную и цифровую формы с помощью преобразователей «значение-вероятность» и «значение-двоичный код» [2]. Затем, с целью наиболее полного использования преимуществ цифровой и вероятностной форм представления дискретной информации, под управлением интеллектуального блока анализа информации, преобразованные входные сигналы автоматически подаются на обработку либо в вероятностное или цифровое АЛУ, либо в специализированный цифро-вероятностный процессор. Получаемые итоговые результаты на выходе подсистемы в виде вероятностного отображения [3] в значительно меньшем информационном объёме, чем исходная информация, поступают в телекоммуникационную подсистему, на входе которой автоматически осуществляется их криптографическая защита [3-5].

Обработка потока первичных данных непосредственно у датчиков не только разгружает работу центрального пульта, но и дает возможность оценить надежность работы первичных источников информации и прогнозировать поведение измерительно-управляющей системы и объекта управления, что дает возможность повысить эффективность работы ИУС в ядерной энергетике.

Необходимо подчеркнуть, что устройства, построенные по принципам вероятностной схемотехники, дают большой выигрыш в аппаратном объеме при сопоставимой производительности [6].

Вывод

Таким образом, цифро-вероятностные процессоры и системы, построенные на основе принципов вероятностной схемотехники, могут являться хорошей альтернативой устройствам, реализованным по принципам цифровой схемотехники, используемых в измерительно-управляющих системах на АЭС, поскольку вероятностная логика дает возможность уменьшить аппаратный объем, повысить помехоустойчивость к ЭМИ, а также снизить энергоемкость.

Библиографический список

1. Моисеев Д.В. / Метод построения вероятностного широкодиапазонного спектрометра повышенной точности / Д.В. Моисеев, Н.Е. Сапожников. Зб. наук. пр. СКУАЕтаП. – Севастополь: СКУАЕтаП, 2010. – Вип. 2(34) С. 209 – 215.
2. Сапожников Н.Е. / Построении преобразователя частота – вероятность //Н.Е. Сапожников, Д.В. Моисеев, Е.В. Скрябина, А.Г. Шокин. Зб. наук. пр. СКУАЕтаП. – Севастополь: СКУАЕтаП, 2010. – Вип. 4(36) С. 217-222.
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. СКУАЕтаП. – Севастополь: СКУАЕтаП, 2011. – Вип. 3 (39) С. 134-140.
4. Сапожников Н.Е. / Построение линейного вероятностного поточного шифратора / Н.Е. Сапожников, Ю.Ю. Столярчук, Д.В. Моисеев // Вісник ДУІКТ – Київ: ДУІКТ, 2011. – Т. 9, № 4. С. 315-318.
5. Шокин А.Г. / Новые методы помехоустойчивого кодирования информации / А.Г. Шокин, Н.Е. Сапожников, Д.В. Моисеев // Восточно-Европейский журнал передовых технологий //научный журнал. - Харьков: Технологический центр, 2012. - №6/9 (60). С – 26–29.
6. Сапожников Н.Е. / Оценка погрешностей выполнения вероятностных арифметических операций сложения и умножения // Н.Е. Сапожников, Д.В. Моисеев, П.С. Бейнер, Н.В. Бейнер. Восточно-Европейский журнал передовых технологий //научный журнал. - Харьков: Технологический центр, 2013. - №3/4 (63). С – 40–42.

СЕКЦИЯ 6
Компьютерные сети



УДК 004.7

А.Д. Бернацкий, магистрант 1-го курса

Научные руководители: Ю.В. Доронина, докт. техн. наук, профессор,

И.В. Дымченко, старший преподаватель

Севастопольский государственный университет

СИСТЕМА ПОДДЕРЖКИ ПРИНЯТИЯ РЕШЕНИЙ ПО ОПТИМИЗАЦИИ ПРОГРАММНО-АППАРАТНОЙ СРЕДЫ

Аннотация: в статье приводится описание архитектуры системы поддержки принятия решения, предназначенной упростить и ускорить разработку ИТ-инфраструктуры на этапе технического проекта, так же повысить гибкость управления проектом и рисками, которые могут возникнуть при эксплуатации ИТ-инфраструктуры.

Ключевые слова: ИТ-инфраструктура, ИТ-сервис, инфокоммуникационная система и сеть, информационная система.

Annotation: The article describes the architecture of the decision support system designed to simplify and accelerate the development of the IT infrastructure at the stage of the technical project, as well as to increase the flexibility of project management and the risks that may arise in the operation of the IT infrastructure.

Key words: IT infrastructure, IT service, info communication system and network, information system

Введение

При проектировании ИТ-инфраструктуры возникает задача выбора оборудования. Проблема заключается в том, что нет постоянного функционала оборудования различного назначения, каждый производитель старается в каждой модели оборудования реализовать как можно больше функционала, что позволяет приобретать вместо нескольких единиц оборудования – одну, однако при этом возникает задача многокритериального выбора.

Цель: предложить архитектуру системы поддержки принятия решения (СППР), которая ускорит процесс принятия решения при проектировании ИТ-инфраструктуры, в том числе геоинформационной платформы в организации.

Основная часть

Решение задачи поддержки принятия решения при проектировании ИТ-инфраструктуры организации, в том числе геоинформационной платформы в организации заключается в создании эффективной среды принятия решения на основе СППР. Рассмотрим основные понятия.

ИТ-инфраструктура – это комплекс взаимосвязанных информационных систем и сервисов, обеспечивающих функционирование и развитие средств информационного взаимодействия предприятия (рисунок 1) [1].



Рис.1. Составляющие ИТ-инфраструктуры

Программно-аппаратный комплекс (ПАК) – это совокупность сервиса и аппаратной части на которой он развернут.

ИТ-сервис – это программное обеспечение, предоставляющее потребителю (сотруднику, работнику) определенные функциональности для обеспечения выполнения ими своих бизнес функций [2].

Аппаратная среда – раздел hardware в ИТ, например, физический сервер, сетевое оборудование.

Важным элементом при оптимизации и проектировании ИТ-инфраструктуры организации является анализ эффективности проектных решений (проекта в целом), на основе комплексного требования. К основным критериям комплексного требования относятся: стоимость внедрения и администрирования, затраты на обеспечение информационной безопасности, а также способность ИТ-инфраструктуры выполнять требуемые функции в заданный период времени.

Принятие проектных решений производится на основе:

- исследовании методов построения оптимальной ИТ-инфраструктуры в зависимости от основных задач данной инфраструктуры;
- исследовании эффективности автоматизации (информатизации);
- анализа эффективности поддержки (администрирования) ИТ-инфраструктуры;
- выявлении технических требований к ИТ-инфраструктуре.

Построение ИТ-инфраструктуры принципиально возможно на основе метода «сверху вниз», т.к. проектирование выполняется для удовлетворения главной цели – интеграции комплекса работоспособных сервисов. Для реализации главной цели принимается множество инженерных решений, при этом построение систем «сверху вниз» требует глубокого анализа существующей ИТ-инфраструктуры.

Анализ существующей ИТ-инфраструктуры подразумевает сбор данных о применяемых ПАК и их взаимосвязях, и взаимозависимостях, физических вычислительных устройств. Только после сбора данной первичной информации возможно принимать решение о необходимости проектирования или перепроектирования, после чего производить планирование этапов проектирования и т.д.

Особенностью работы СППР является автоматическое формирование требований к аппаратной части ИТ-инфраструктуры. Формирование требований предлагается осуществлять при помощи СППР, входными данными для которой являются технические характеристики существующей на момент проектирования ИТ-инфраструктуры и совокупность сервисов для развертывания в организации.

Следующим шагом производится выбор оборудования для инфраструктуры, принимая во внимание представленные базовые и рекомендуемые технические характеристики.

После выбора оборудования, необходимо ввести в СППР характеристики выбранного оборудования, после чего получить оценку эффективности и риски (недочеты) проектного решения, рисунки 2.

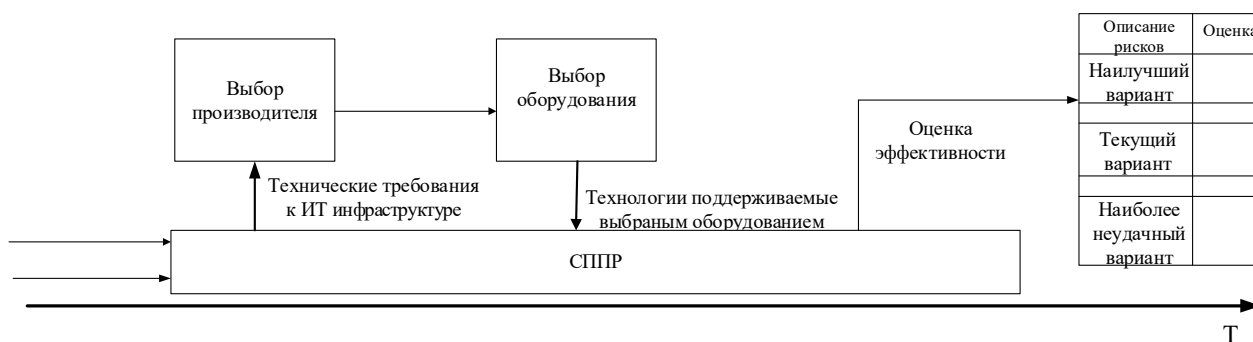


Рис 2. Этапы выбора оборудования при использовании СППР

Технические требования подразделяются на две составляющие: базовые и рекомендуемые, рисунок 2. Минимальные требования – это описание минимальных характеристик, которым должна обладать инфраструктура для того, чтобы в ней мог использоваться сервис. Рекомендуемые требования – описание характеристик, которые существенно упрощают и снижают затраты на администрирование, позволяют улучшить бесперебойную работу сервисов, повысить уровень безопасности и т.д.

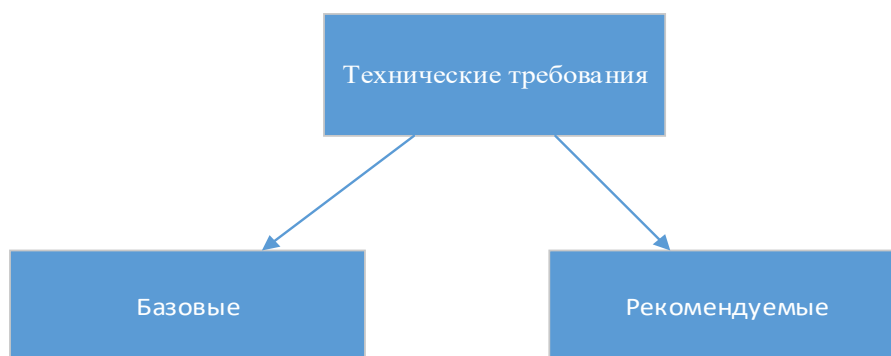


Рис 3. Технические требования

Рекомендуемые требования включают в себя классические рекомендации построения ИТ-инфраструктуры и параметры полученных наблюдений в результате практического опыта проектирования и эксплуатации ИТ инфраструктуры.

СППР формирует технические требования с помощью генетических алгоритмов для основных составляющих ИТ-инфраструктуры: телекоммуникационной, информационной и инженерной составляющей, рисунок 4.

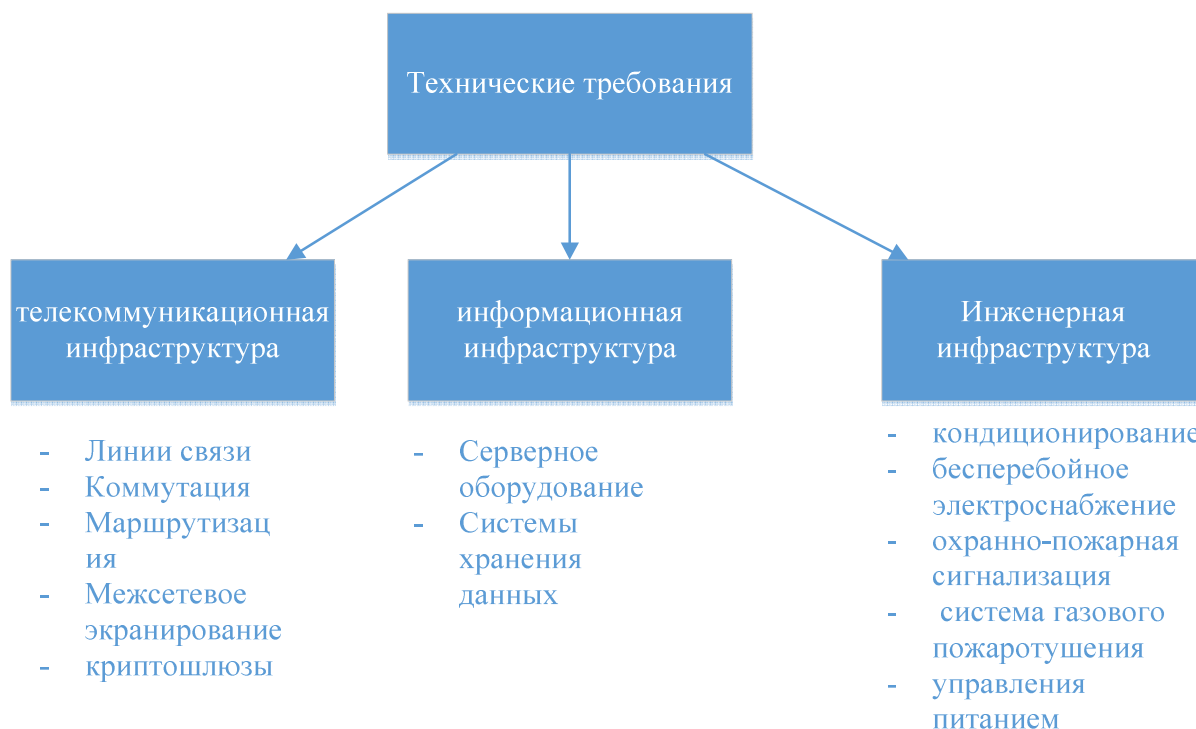


Рис 4. Основные составляющие ИТ-инфраструктуры

Необходимость в множестве проектных решений объясняется ограничением ресурсов, на проектирование и сам проект. Частым случаем является изменение стоимости проекта за период проектирования, например, изменение валютного курса.

С помощью СППР есть возможность получить оценку проекта, а также позволяет внести изменения на конечной стадии проектирования, при увеличении/уменьшении ресурсов с описанием рисков и достоинств проекта.

Ожидаемые результаты при создании СППР: повышение эффективности работы проектировщика и администратора информационных систем.

Выводы

Использование предложенной СППР облегчит проектировщику задачу выбора оборудования, а также укажет на узкие места в спроектированной инфраструктуре. Дальнейшее развитие СППР предполагает автоматизацию процесса подбора оборудования.

Библиографический список

1. Информационная инфраструктура предприятия [электронный ресурс]. Режим доступа: <https://www.lankey.ru/kis/> (дата обращения 15.03.17)
2. Системы предоставления ИТ-сервисов [электронный ресурс] Режим доступа: http://www.ecoprogram.ru/system/it/it_service/ (дата обращения 17.03.17)

УДК 004.77

В.В.Мозговая, магистрант 1-го курса**Научный руководитель: Е.Н.Мащенко, канд. техн. наук, доцент***Севастопольский государственный университет***АНАЛИЗ ПРОБЛЕМ ПРАКТИЧЕСКОЙ РЕАЛИЗАЦИИ БЕСПРОВОДНЫХ СЕНСОРНЫХ СЕТЕЙ**

Аннотация: Рассмотрены основные моменты, связанные с использованием беспроводных сенсорных сетей. Исследованы преимущества и ограничения применения подобного рода сетей. Описаны основные проблемы, возникающие в процессе проектирования сенсорных сетей, а также существенные особенности этого процесса.

Ключевые слова: беспроводные сенсорные сети, сенсоры, БСС, применение беспроводных сенсорных сетей, недостатки беспроводных сенсорных сетей.

Annotation: The paper studies main points connected with using wireless sensor networks. The paper examines the benefits and drawbacks taking into account the character of this kind of networks. It describes the main problems in the course of sensor networks designing and significant features of this process.

Key words: wireless sensor networks, sensors, WSN, WSN applications, disadvantages of WSN.

Сенсорная сеть представляет собой инфраструктуру, которая состоит из сенсорных (измерительных), вычислительных и коммуникационных элементов и предоставляет администратору (наблюдателю) возможность наблюдать и реагировать на события и явления в определенной среде. Администратор, как правило, является гражданским, правительственным, коммерческим, промышленным субъектом. Среда может быть частью физического мира, биологической системой или информационной структурой. Использование сенсорных сетей является одним из наиболее активно развивающихся методов сбора и передачи данных. Основными областями применения сенсорных сетей, на сегодняшний день, являются системы сбора информации, наблюдения, мониторинга и медицинской телеметрии [1].

Сенсорные сети состоят из четырех основных элементов (рисунок 1):

- 1) множество распределенных или локализованных датчиков;
- 2) сеть связи;
- 3) центральный узел кластеризации (сбора) информации;
- 4) совокупность вычислительных ресурсов (узлов) для обработки и корреляции данных, статуса запросов и интеллектуального анализа.

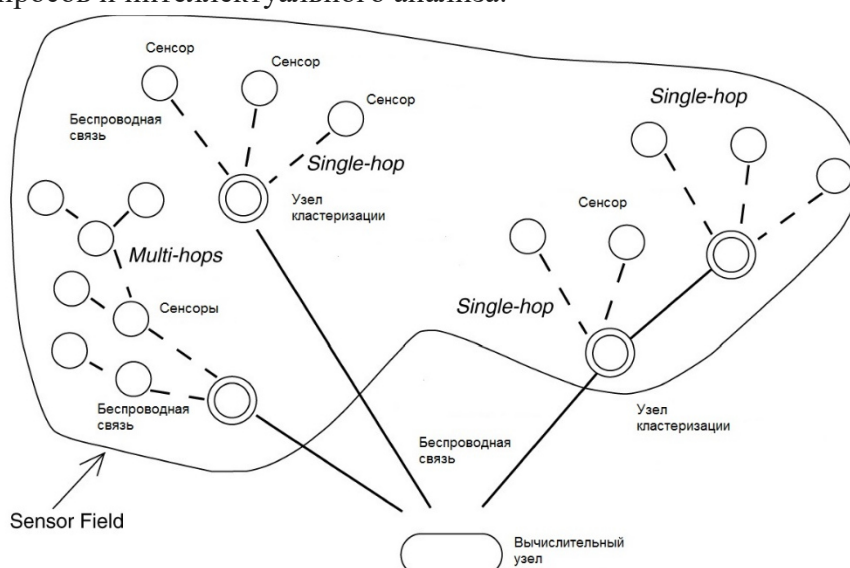


Рис.1. Типовое расположение сенсоров

В данном контексте, сенсорные и вычислительные узлы являются неотъемлемой частью сенсорной сети, однако, фактически, некоторые вычисления могут производиться и внутри самих сенсорных узлов. Из-за потенциально большого количества собираемых данных, алгоритмические методы управления данными играют важную роль в сенсорных сетях.

Вычислительная и коммуникационная инфраструктура, связывающая сенсорные сети, часто является специфической и заточена под конкретные узкоспециализированные приложения.

Исследователи рассматривают беспроводную сенсорную сеть (БСС) как распределенную, самоорганизующуюся сенсорную сеть множества датчиков и исполнительных устройств, объединенных между собой посредством радиоканалов.

Датчик (англ., transducer) – устройство, которое используется для преобразования одного вида энергии в другой [2].

Датчики в БСС характеризуются различными целями, функциями и возможностями, благодаря чему, под давлением последних технологических достижений, БСС нашли применение во множестве различных сфер деятельности, таких как радарные сети, используемые в контроле воздушного сообщения, метеорологические станции, национальные электроэнергетические сети, военная сфера и т.д. Все эти системы используют специализированные компьютеры и протоколы связи и, соответственно, являются очень дорогими. Однако намного дешевле использовать БСС в сферах безопасности, здравоохранения и коммерции, что и планируется в ближайшем будущем [1].

Сенсорная сеть - это междисциплинарная область, в которую кроме всего прочего, входит работа с сетью, обработка сигналов, искусственный интеллект, управление базами данных, алгоритмы управления питанием и т.д. Приложения, принципы сетевой организации и протоколы еще только начинают разрабатываться. Прогресс в области беспроводных и проводных сетевых технологий, широкое распространение сетевых технологий, развитие IT-индустрии (например, создание процессоров более высокой мощности, GRID-вычисления и т.д.) в сочетании с последними инженерными достижениями дали дорогу маломощным и недорогим датчикам и актуаторам.

Актуатор (англ., actuator) – исполнительное устройство, которое реагирует на поступивший сигнал для изменения состояния управляемого объекта.

Сенсор (англ., sensor) – устройство, которое воспринимает контролируемое воздействие (свет, давление, температуру и т. п.), измеряет его количественные и качественные характеристики и преобразует данные измерения в сигнал. Сигнал может быть электрический, химический или другого типа [2].

Сенсоры включают в себя датчики электрического и магнитного поля; датчики частоты радиоволн, инфракрасные датчики, лазеры, датчики местоположения (навигации), сейсмические датчики, экологические, биохимические датчики и т.д.

Современные сенсоры могут быть определены как "умные" недорогие устройства, оснащенные множеством чувствительных элементов. Сенсоры или беспроводные узлы (WNs) также иногда называются мотами (moten) [2].

Сенсорные сети и системы делятся на две категории:

- 1я категория БСС (C1WSN): почти всегда Mesh-системы (многосвязные) с многозвенной радиосвязью между сенсорами, использующие динамическую маршрутизацию как в проводных, так и беспроводных участках сети. Как правило, к этой категории относятся военные системы;

- 2я категория БСС (C2WSN): системы типа "точка-точка" или "звезда", с прямыми связями, использующие статическую маршрутизацию. Как правило, имеет место наличие только одного маршрута от датчика(источника) к передающему узлу сети (наземному или проводному). К этой категории относятся домашние сенсорные системы, такие как "умный дом" или "интеллектуальное здание" [1].

Основные элементы и принципы проектирования беспроводных сенсорных сетей (на примере C1WSN) характеризуются следующими факторами (в C2WSNs есть многие из тех же самых ограничений, но не все):

- большое количество сенсоров (например, 64000 или более узлов должны поддерживаться системой и устройствами адресации);
- большие потоки данных;

- неполные / неопределенные данные;
- высокая вероятность отказа узла;
- высокая вероятность отказа канала (помехи);
- электрические ограничения мощности;
- ограниченная мощность процессора;
- многоуровневая топология;
- ограниченная административная поддержка сети.

Сенсоры могут быть развернуты в различных средах: в открытом пространстве, на поле боя, в биологически и/или химически загрязненных местах, в доме, на теле человека и т.д.. Сенсор обычно имеет встроенные возможности обработки и хранения данных; сенсор может иметь один или более датчиков, работающих в акустическом, сейсмическом, инфракрасном, оптическом и других спектрах. Сенсор имеет интерфейсы беспроводной связи и также часто знает свое местоположение и позиционирование, используя систему глобального позиционирования (GPS) или локальный алгоритм. Каждый из распределенных узлов датчиков, как правило, имеет возможность собирать данные, анализировать их, и направлять их в обозначенную точку.

В течение долго времени, производители использовали собственные технологии сбора данных о производительности устройств. В начале 2000х, поставщики датчиков устройств исследовали пути введения стандартизации. Беспроводные сети обычно передают небольшие объемы простых данных (например, уровень температуры). Для "внутри-здания" приложений, разработчики исключили Wi-Fi (IEEE 802.11b) стандарты для датчиков, так как они являются слишком сложными и поддерживают более высокую пропускную способность, чем на самом деле требуется для обычных датчиков. Инфракрасные системы требуют прямой видимости, что не всегда достижимо; Bluetooth (IEEE 802.15.1) технология была сначала рассмотрена как возможная, но вскоре была признана слишком сложной и дорогой. Это открыло дверь для нового стандарта IEEE 802.15.4 ZigBee (более специализированный, ZigBee имеет надстройки над недавно принятым стандартом IEEE 802.15.4 и поддерживает множество приложений). C2WSNs имеют более низкие уровни стека протоколов связи (физический и MAC), которые сопоставимы с персональной вычислительной сетью (PAN), определенной в стандарте IEEE 802.15, следовательно, возможно использование этих стандартов для C2WSNs. IEEE 802.15.4 работает в 2,4 ГГц промышленном, научном и медицинском (ISM) радиодиапазоне и поддерживает передачу данных со скоростью до 250 кбит в диапазоне от 9 до 60 метров. ZigBee / IEEE 802.15.4 разработана в дополнение к технологиям беспроводной связи, таким как Bluetooth, Wi-Fi и UWB, и предназначена для коммерческих сенсорных "точка-точка" приложений, в которых наличие проводных соединений не представляется возможным и необходима сверхнизкое питание и низкая стоимость [1].

С появлением стандарта ZigBee / IEEE 802.15.4, системы, ожидаемо, перешли к стандартизированному подходу разработки, что позволяет датчикам передавать информацию в стандартизированной форме. C2WSNs и C1WSN, которые работают за пределами здания и в более широкой географической области, могут использовать любые другие стандартные технологии радиосвязи. Рынок C2WSN, как ожидается, значительно возрастет в ближайшем будущем: объем беспроводных устройств с низкой скоростью передачи данных, по прогнозам, в три раза превышает размер Wi-Fi, в связи с ожидаемым развертыванием систем на основе стандарта ZigBee / IEEE 802.15.4.

Традиционно, сенсорные сети используются в контексте мощных приложений, таких как системы обнаружения радиации и ядерных угроз, датчики оружия для судов, биомедицинские приложения, сейсмический мониторинг и другие. Совсем недавно исследователи сосредоточили внимание на сетевых биологических и химических датчиках. Кроме того, эта технология начала распространяться на приложения для простых потребителей. Существующие и потенциальные области применения сенсорных сетей включают в себя, среди прочего, военное зондирование, безопасность, управления воздушным

движением, видеонаблюдение, промышленное производство и автоматизация, управление технологическими процессами, управление запасами и т.д.

Необходимо понимать, что сенсорные сети имеют дело с пространством и временем: месторасположение, зона покрытия и синхронизация данных. Данные являются неотъемлемой "валютой" сенсорной сети. Как правило, БСС обрабатывает большое количество зависящих от времени данных. Следовательно, сенсорные сети часто поддерживают внутрисетевые вычисления. Некоторые сенсорные сети используют обработку в узле-источнике; другие используют иерархическую архитектуру обработки. Вместо отправки необработанных данных к узлам, ответственным за слияние данных, узлы часто используют способность локально выполнять основные вычисления, а затем передавать только подмножество данных, и/или частично обработанные данные. В иерархической архитектуре, обработка происходит последовательно на всех уровнях через которые, представляющая интерес информация, достигает соответствующего процесса принятия решений и/или административного узла. Сенсоры всегда ограничены в энергоснабжении и пропускной способности канала передачи; эти ограничения, в сочетании с типичным развертыванием большого числа сенсоров, вызывают множество проблем при проектировании и управлении БСС:

1. Энергопотребление. Ограничение по энергопотреблению связано с тем, что сенсоры работают от источника питания с ограниченным лимитом энергии (обычно батарейка). Чем реже они будут заменяться или заряжаться, тем более низкую стоимость будет иметь их обслуживание. Также энергопотребление является важным ограничением при использовании сенсоров, доступ к которым сложен, следовательно, источник питания не может быть заменен или подзаряжен. Для уменьшения энергопотребления обычно предусматривается отключение передатчиков сенсорных узлов, когда нет необходимости передачи информации. На сетевом уровне используются оптимальные пути передачи информации от сенсорного узла до координатора (базовой станции), учитывая число промежуточных узлов, требуемую энергию и доступную энергию. Кроме сетевого протокола на потребление энергии влияет конструкция узлов (например, маленький размер памяти, эффективность переключений между заданиями), программное обеспечение, механизмы защиты и даже рабочие приложения. Из-за требования к низкому потреблению энергии типичный сенсорный узел имеет небольшие скорости выполнения операций и объемы хранимой информации. Также из-за этого нежелательно использование некоторых устройств, таких как GPS-приемники.

2. Самоуправление. Сенсорные сети часто должны работать в удаленных областях и в жестких условиях, без возможности их обслуживания и ремонта. Поэтому, сенсорные узлы должны конфигурироваться самостоятельно, взаимодействовать с другими узлами, адаптироваться к поломкам изменениям окружающей среды без вмешательства человека.

3. Дальность расстояния. Выбор беспроводного соединения налагает ряд ограничений на реализацию сенсорных сетей. Например, затухание сигнала ограничивает расстояние передачи информации. Так связь между мощностями сигналов переданной и принятой информацией описывается законом обратного квадрата расстояния, то увеличение расстояния между сенсорным узлом и маршрутизатором/координатором приводит к увеличению мощности передаваемого сигнала. Поэтому более эффективно, с точки зрения затрат энергии, разделить большие расстояния передачи информации в сенсорных сетях на несколько небольших.

4. Способ передачи данных. В сенсорной сети узлы обычно общаются посредством беспроводной связи. Связь может осуществляться посредством радио, инфракрасного излучения (ИК-порта) или оптических сигналов. Одним из наиболее распространенных вариантов радиосвязи является использование полос частот для промышленных, научных и медицинских целей ISM (Industrial, Scientific and Medical), которые определены Сектором радиосвязи Международного союза электросвязи ITU-R и доступны без лицензий в большинстве стран.

В России на основании Решения Государственной комиссии по радиочастотам (ГКРЧ) № 08-24-01-001 от 28.04.2008 и № 07-20-03-001 от 07.05 2007 в качестве ISM выделены частотные диапазоны LPD (англ. Low Power Device) 433,075 – 434,750 МГц, PMR (англ. Private Mobile Radio) 446,00625 – 446,09375 и 868,7 – 869,2 МГц. Эти радиочастоты могут использоваться без оформления специального разрешения ГКРЧ и совершенно бесплатно при условии соблюдения требований по ширине полосы, излучаемой мощности (до 10 мВт в районе частоты 434 МГц, до 500 мВт в районе частоты 446 МГц и до 25 мВт в районе частоты 868 МГц) и назначению радиопередающего изделия [2].

Особые требования к применению сенсорных сетей делают выбор среды передачи сложной задачей. Например, морские приложения требуют использования водной среды передачи. Здесь нужно использовать длинноволновые излучения, которые могут проникать сквозь поверхность воды. В труднодоступной местности или на поле боя могут возникнуть ошибки и большие помехи. Кроме того может оказаться, что антенны узлов не обладают нужной высотой и мощностью излучения для связи с другими устройствами. Следовательно, выбор передающей среды должен сопровождаться надежными схемами модуляции и кодирования, что зависит от характеристик передающего канала.

5. Конструкция. Главной целью беспроводных сенсорных сетей является создание маленьких, дешевых и эффективных устройств. Ограничения по размерам влияет на структуру протоколов и алгоритмов, реализованных в беспроводных сенсорных сетях. Например, таблица всех маршрутов в сети может быть слишком большой и не поместиться в памяти узла. Поэтому только небольшая часть информации (например, список соседних узлов) может храниться в памяти узла.

6. Безопасность. Удаленное расположение сенсоров и их автоматическая работа увеличивает их незащищенность к сторонним вторжениям и атакам. При беспроводном соединении достаточно легко для нарушителя перехватить пакеты, передаваемые сенсорным узлом. Например, наиболее большая угроза осуществления атаки «отказа в обслуживании» (denial-of-service), цель данной атаки нарушить корректное функционирование сенсорной сети. Это может быть достигнуто при помощи различных способов, например, при подаче мощного сигнала, который мешает сенсорным узлам обмениваться информацией («белый шум» или jamming attack). Есть различные варианты защиты систем от злоумышленников, но для многих из них необходимы высокие требования к аппаратным ресурсам, что труднодостижимо на жестко ограниченных по многим требованиям сенсорных узлах. Следовательно, сенсорные беспроводные сети требуют новых решений для создания ключей, их распространения, идентификации и защиты узлов.

7. Затраты на сенсорные узлы. По определению, сенсорная сеть состоит из большого набора сенсорных узлов. Отсюда следует, что стоимость отдельного узла имеет решающее значение для общей финансовой характеристики сенсорной сети. Очевидно, что стоимость каждого узла датчика должна поддерживаться на низком уровне, чтобы быть разработка была финансово целесообразна. Современные сенсорные системы на основе Bluetooth стоят около 10 долларов, но в Bluetooth-технологии передача данных сильно ограничена с точки зрения пропускной способности и расстояния. Таким образом, в статье рассмотрены основные моменты, связанные с использованием беспроводных сенсорных сетей, преимущества и ограничения их применения. Описаны основные проблемы, возникающие в процессе проектирования сенсорных сетей, а также существенные особенности этого процесса.

Библиографический список

1. Sohraby K., Minoli D., Znati T. Wireless Sensor Network, Technology, Protocols and Applications. John Wiley & Sons, New Jersey, 2007. P. 1–10
2. Интернет вещей: учебное пособие / А.В. Росляков, С.В. Ваяшин, А.Ю. Гребешков. – Самара: ПГУТИ, 2015. – С. 46–60.

УДК 004.6

С.А. Сивков, магистрант 2-го курса

Научный руководитель: Ю.К. Апраксин, д.т.н., профессор.

Севастопольский государственный университет

ИССЛЕДОВАНИЕ НЕСОСТОЯТЕЛЬНОСТИ СЕТЕВЫХ ПРОТОКОЛОВ НА ОСНОВЕ КОНЕЧНО-АВТОМАТНОЙ МОДЕЛИ

Аннотация: Предлагается модель для исследования несостоятельности сетевых протоколов на основе взаимодействия систем, представленных конечно-автоматной моделью. В качестве метода проверки несостоятельности протокола выбран анализ дерева достижимых глобальных состояний.

Ключевые слова: несостоятельность протокола, протоколов информационного обмена, расширенный конечный автомат, дерево достижимых глобальных состояний, конечно-автоматная модель TCP протокола.

Annotation: A model for investigating the insolvency of network protocols based on the interaction of systems presented by finite-automaton model is suggested. The analysis of the tree of reachable global states is selected as a method of verification protocol insolvency.

Key words: protocol insolvency, information exchange protocols, extended finite automaton, tree of reachable global states, finite automaton model of TCP protocol.

В современном мире большой объем личной, коммерческой и прочей информации передается через открытые локальные и глобальные сети. Стремительный рост исследований в области сетевых и информационных технологий в последней четверти XX века привел к развитию направления, которое связано с разработкой протоколов информационного обмена. Важной задачей является задача исследования и определения безопасности современных протоколов, а также разработка новых безопасных протоколов.

Анализ безопасности протоколов информационного обмена состоит в обнаружении возможных несостоятельств в протоколах.

В настоящее время обрели развитие математические методы формального анализа безопасности протоколов. Основу таких методов составляют процедуры формального описания протокола с последующей верификацией работы модели протокола. Большинство известных методов формального анализа ориентированы только на доказательство традиционных требований безопасности (например, секретности и аутентичности). Как показывает реальная практика, обнаружение несостоятельств в протоколах может происходить и происходит спустя длительное время после опубликования, разработки и внедрения протоколов. Возможно, что новые несостоятельности и угрозы безопасности обнаруживаются в протоколах, которые ранее уже были проанализированы с использованием одного или нескольких методов [1].

Наиболее подходящей абстрактной моделью протокола является расширенный конечный автомат [2], который в дальнейшем будем называть протокольным автоматом (РА).

РА определяет взаимодействие, как минимум, двух процессов. Описание такой модели можно представить в виде пары взаимодействующих протокольных автоматов $РА_i$ и $РА_j$:

$$РА_i::\langle \{+m_{ji}\} \cup \{-m_{ij}\}, \{P_i^k\}, P_i^0, \gamma_i = \gamma_i(P_i^k, \pm m_{ij}^k), F_i \rangle;$$

$$РА_j::\langle \{+m_{ij}\} \cup \{-m_{ji}\}, \{P_j^r\}, P_j^0, \gamma_j = \gamma_j(P_j^r, \pm m_{ji}^r), F_j \rangle.$$

Здесь введены следующие обозначения:

$\{+m_{ji}\}(\{+m_{ij}\})$ – сообщения, принимаемые $РА_i$ ($РА_j$) от $РА_j$ ($РА_i$);

$\{-m_{ij}\}(\{-m_{ji}\})$ – сообщения, посылаемые $РА_i$ ($РА_j$) к $РА_j$ ($РА_i$);

$\{P_i^k\}$ – множество процессов ($k=1,2,\dots,K$), реализующих сервис протокола $РА_i$;

$\{P_j^r\}$ – множество процессов ($r=1,2,\dots,R$), реализующих сервис протокола $РА_j$;

P_i^0 (P_j^0) – начальные (активизирующие) процессы $РА_i$ ($РА_j$);

$\gamma_i(\gamma_j)$ – функции выходов, определяющие каким процессом $РА_i$ ($РА_j$) формируется сообщение посылка к $РА_j$ ($РА_i$);

F_i (F_j) – совокупность процессов $РА_i$ ($РА_j$), завершающих формирование сервиса.

Каждое состояние РА интерпретируется как внутренний процесс обработки или формирования сообщений. Передача сообщений осуществляется через коммуникационную среду. Модель этой среды представляет собой совокупность очередей с известной системой обслуживания. Например, если два процесса обмениваются сообщениями через полудуплексный канал и дисциплина обслуживания очереди сообщений C_{ij} (передача от $РА_i$ к $РА_j$) и C_{ji} (передача от $РА_j$ к $РА_i$) определяется как FIFO (First Input - First Output), то описание модели канала можно представить тремя правилами функционирования:

- посылка сообщения
 $P'_i = \gamma_i(P_i, -m); C'_{ij} = C_{ij}.m; P'_j = \gamma_j(P_j, -m); C'_{ji} = C_{ji}.m;$
- приём сообщения
 $P'_i = \gamma_i(P_i, +m); C_{ij} = m.C'_{ij}; P'_j = \gamma_j(P_j, +m); C_{ji} = m.C'_{ji};$
- внутреннее событие без посылки, либо приёма сообщений (таймаут)
 $P'_i = \gamma_i(P_i, \emptyset); C'_{ij} = C_{ij}; P'_j = \gamma_j(P_j, \emptyset); C'_{ji} = C_{ji}.$

Здесь $C'_{ij}, P'_i, C'_{ji}, P'_j$ - состояния каналов и РА после завершения события.

Предложенная модель позволяет определять текущее состояние системы взаимодействия двух объектов РС как $(P_i, C_{ij}, P_j, C_{ji})$, которое будем называть глобальным состоянием.

Модель глобального состояния РС, состоящей из N взаимодействующих объектов, можно представить квадратной матрицей $(PC)_{N \times N}$ диагональные элементы которой определяют локальные состояния $РА_i$ ($i = 1, 2, \dots, N$), а на пересечении i -ой строки и j -го столбца фиксируется состояние канала связи C_{ij} .

Глобальное состояние всей системы можно также наглядно представить в виде матрицы, первая строка которой описывает состояние очередей сообщений и подтверждений, передаваемых от объекта P_i к объекту P_j , а вторая – состояние аналогичных очередей сообщений и подтверждений, следующих в обратном направлении:

$$\begin{bmatrix} P_i & \rightarrow & C_{ij} \\ \uparrow & & \downarrow \\ C_{ji} & \leftarrow & P_j \end{bmatrix}.$$

Для реализации задачи валидации протоколов используется метод анализа дерева достижимых глобальных состояний (ДДГС) [2].

К преимуществам метода следует отнести удобную графическую форму представления и достаточно простой способ автоматизации процесса анализа. Созданные на основе этого метода автоматизированные системы применялись для исследования ряда реальных протоколов [3]. Основным недостатком метода является быстрый рост числа глобальных состояний по мере роста сложности протоколов. Известно несколько разновидностей данного подхода: метод перебора, метод диалоговых матриц, метод фазовых диаграмм, метод «прилегающих» состояний и метод совместных путей.

По окончании генерации новых глобальных состояний, создается граф, узлы которого являются частичными глобальными состояниями, а дуги — возможными переходами между этими состояниями. Анализ такого графа дает возможность проверить следующие свойства протокола:

- потерю или неспецифицированный прием входного события (свойство 2). Канал K (J, I) содержит событие, а в автомате I нет перехода из текущего состояния, помеченного приемом этого события;
- тупиковое состояние (свойство 1). Для всех I ($1 \leq I \leq N$) текущее состояние $G(I)$ не имеет переходов, помеченных выходными или внутренними событиями, а все каналы пусты;
- переполнение среды (свойство 5). Возникает в том случае, если среди всех автоматов найдется хотя бы один такой, в котором имеется переход из текущего состояния, помеченный выходным событием в канал $K(I, J)$, и число событий в этом канале равно $Cap(I, J)$;

- избыточная спецификация (свойство 4). Возникает в том случае, если в каком-либо автомате найдется переход, который ни разу не был использован при построении графа достижимых глобальных состояний.

Корневой вершиной ДДГС является начальное глобальное состояние ($P_i^0, C_{ij} = \emptyset, P_j^0, C_{ji} = \emptyset$). Переход на новую (текущую) вершину определяется событием, реализуемым текущим процессом $P_i^k(P_j^r)$. Валидация выполняется путем обхода ДДГС от корневой вершины до листьев дерева. Если обнаруживается несоответствие, то вершины данной ветви, расположенные ниже, уже не анализируются.

Рассмотрим пример валидации системы «клиент-сервер» на основе протокола ТСР.

В технологии разработки протоколов существует набор свойств, обязательный для любых протоколов. Поэтому проверка корректности протокола должна обеспечивать доказательство того, что спецификация протокола обладает этими свойствами. Свойства эти следующие:

1. Отсутствие статических блокировок. Это значит, что в протоколе не существует такого состояния или набора состояний, из которых нет переходов в другие состояния.

$$S = \begin{bmatrix} 2 & \lambda \\ \lambda & 1 \end{bmatrix}.$$

Каналы C_{ij} и C_{ji} пусты, а автоматы PA_i и PA_j , находятся соответственно в состояниях s_2^i и s_1^j , из которых не предусмотрен переход по посылочным операциям.

2. Полнота, то есть протокол обеспечивает реакцию на все возможные входные сообщения (отсутствуют ошибки неспецифицированных приемов).

В таблице переходов, автомат PA_j , находясь в любом состоянии, всегда может принимать сообщения m , находящиеся в канале C_{ij} . Т.е. не должно быть таких ситуаций, когда автомат PA_j не знает, как реагировать на входное сообщение m .

3. Однозначность соответствия состояний, то есть отсутствие таких протокольных объектов, у которых одно состояние может сосуществовать с несколькими различными состояниями какого-либо другого объекта.

В дереве достижимых глобальных состояний отсутствуют глобальные состояния, которые уже повторялись на ранее рассмотренных уровнях.

Не может глобальное состояние $S = \begin{bmatrix} 2 & \lambda \\ \lambda & 1 \end{bmatrix}$ на первом уровне дерева повторяться на другом уровне, например – третьем $S = \begin{bmatrix} 2 & \lambda \\ \lambda & 1 \end{bmatrix}$.

4. Отсутствие избыточности, то есть в спецификации протокола нет не поступающих сообщений и невыполняемых действий.

Не может быть такого глобального состояния $S = \begin{bmatrix} 2 & \lambda \\ \lambda & 5 \end{bmatrix}$, если в рассматриваемых автоматах нет 5 состояния.

5. Ограниченность, которая означает, что во время функционирования протокола число сообщений в каждом канале между протокольными объектами не превышает определенного значения, называемого емкостью канала.

Например, при емкости канала равной 2, не может быть такого глобального состояния, в канале которого находится более 2 сообщений - $S = \begin{bmatrix} 2 & 1 * 3 * 3 \\ \lambda & 5 \end{bmatrix}$ или $S = \begin{bmatrix} 2 & \lambda \\ 2 * 1 * 3 * 4 & 5 \end{bmatrix}$.

6. Отсутствие динамических блокировок. В протоколе отсутствует бесконечный цикл функционирования, при котором не производится полезная работа. Различают динамические блокировки, выход из которых логически невозможен, и динамические блокировки, являющиеся следствием определенных временных характеристик протокола, например, темпа обмена сообщениями.

7. Завершаемость (развитие), т.е. протокол всегда достигает конечного (терминального) состояния. Для циклических протоколов это свойство несколько

видоизменяется. Эти протоколы должны обладать свойством развития, которое состоит в том, что протокол достигает своего начального состояния.

Если начальное глобальное состояние $S = \begin{bmatrix} 0 & \lambda \\ \lambda & 0 \end{bmatrix}$, то и конечное состояние должно быть $S = \begin{bmatrix} 0 & \lambda \\ \lambda & 0 \end{bmatrix}$, где λ – может содержать любые m .

8. Самосинхронизация (восстановление после непредвиденной ситуации). Это свойство подразумевает, что после возникновения ненормальной ситуации протокол за конечное время восстановит свое корректное функционирование.

При соблюдении этих свойств в исследуемом протоколе, можно считать отсутствие несостоятельности у рассматриваемого протокола.

Проиллюстрируем взаимодействие сетевых объектов «Клиент - Сервер» на основе TCP протокола. Из существующей диаграммы состояний протокола, выделяют следующие состояния соединения TCP - LISTEN, SYN_SENT, SYN_RECEIVED, ESTABLISHED, FIN_WAIT_1, FIN_WAIT_2, CLOSE_WAIT, CLOSING, LAST_ACK, TIME_WAIT и фиктивное состояние CLOSED (фиктивно, так как представляет состояние, когда уже нет TCB и соединения) [4]. Ниже кратко описаны все эти состояния.

LISTEN - ожидание запроса на соединение от любого удаленного TCP и порта.

SYN_SENT - ожидание соответствующего запроса на соединение после передачи своего запроса.

SYN_RECEIVED - ожидание подтверждения соединения после передачи и приема запросов на организацию соединения.

ESTABLISHED - соединение действует и принятые данные могут быть доставлены пользователю. Это нормальное состояние для процесса обмена данными через соединение.

FIN_WAIT_1 - ожидание запроса на разрыв соединения от удаленного TCP или подтверждения для ранее переданного запроса на разрыв соединения.

FIN_WAIT_2 - ожидание запроса на разрыв соединения от удаленного TCP.

CLOSE_WAIT - ожидание запроса на разрыв соединения от локального пользователя.

CLOSING - ожидание подтверждения от удаленного TCP для запроса на разрыв соединения.

LAST_ACK - ожидание подтверждения для запроса на разрыв соединения, переданного удаленному TCP (это подтверждение включается в запрос на разрыв соединения от удаленной стороны).

TIME_WAIT - ожидание пока пройдет достаточно времени, чтобы быть уверенным в приеме удаленным TCP подтверждения для его запроса на разрыв соединения.

CLOSED - соединения уже нет (разорвано).

Соединение TCP переходит от одного состояния к другому в ответ на события, к числу которых относятся пользовательские вызовы OPEN, SEND, RECEIVE, CLOSE, ABORT и STATUS, входящие сегменты (в частности те, которые включают флаги SYN, ACK, RST, FIN) и тайм-ауты.

Для проверки несостоятельности протокола, представим работу TCP системой взаимодействующих расширенных автоматов. Представим графическую интерпретацию взаимодействующих PA_1 и PA_2 через протокол TCP (рисунок 1).

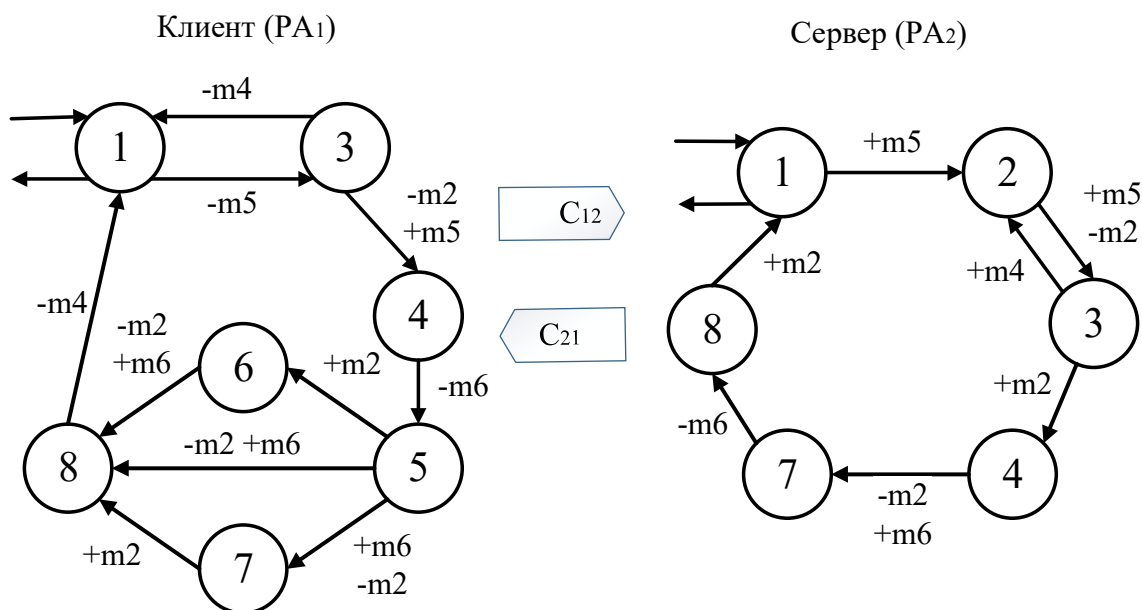


Рис. 1. Исходное представление системы протокольного взаимодействия «Клиент-Сервер» через TCP

Описание протокольных автоматов представлено в таблице 1.

Таблица 1 – Описание состояний протокольных автоматов

Состояние	Клиент	Сервер
1	Начальное состояние. CLOSED	Начальное состояние. CLOSED
2	-	LISTEN
3	SYN_SENT	SYN_RECEIVED
4	ESTABLISHED	ESTABLISHED
5	FIN_WAIT_1	-
6	FIN_WAIT_2	-
7	CLOSING	CLOSE_WAIT
8	TIME_WAIT	LAST_ACK

Возможные сообщения при работе протокола:

- m1: URG: указывает на срочность приема;
- m2: ACK: подтверждение доставки данных;
- m3: PSH: выталкивание данных;
- m4: RST: сброс соединения;
- m5: SYN: синхронизация порядковых номеров;
- m6: FIN: у отправителя больше нет данных.

Построение дерева ДДГС представлено на рисунке 2.

При анализе ДДГС можно выделить следующие состояния: в ДДГС есть вершины (на ярусах 14 и 15), из которых не выполняется передача/прием сообщений из-за соблюдения требований, ограничивающих ёмкость канала. При увеличении объёма очереди увеличится и глубина дерева, но подобные состояния останутся. На ярусе 13 есть состояние, из которого осуществляется переход в другие состояния, которые находятся на других ярусах, что свидетельствует о возможности возникновения циклов, но в данном случае циклы не бесполезные – система всё равно переходит в конечные состояния. В результате построения ДДГС, получено 3 конечных состояния, которые соответствуют диаграмме работы TCP.

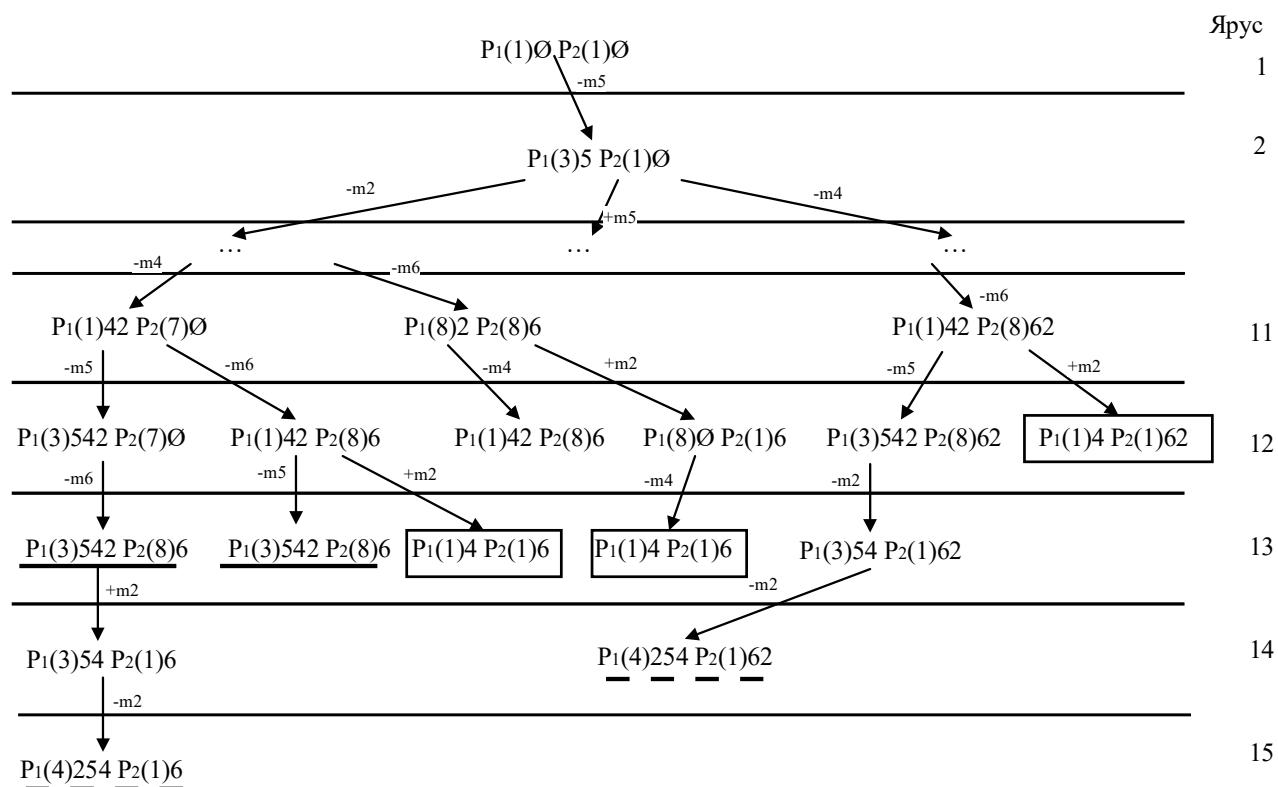


Рис.2. Дерево ДДГС

Предложена система для исследования несостоятельности сетевых протоколов на основе взаимодействия систем, представленных конечно-автоматной моделью. Перечислены характеристики протоколов, несоблюдение которых говорит о несостоятельности. Исследовано конечно-автоматное представление протокола TCP. Построено дерево ДДГС. Время обработки дерева равно 15 условным единицам. Выявлены несостоятельности рассмотренного протокола – заикливание и переполнение емкости канала.

Библиографический список

1. Lowe G. Breaking and fixing the Needham-Schroeder public-key protocol using FDR. // Software - Concepts and Tools, 17: pp. 93-102, 1996
2. Апраксин Ю.К. Моделирование поведения взаимодействующих объектов распределённых систем / Ю.К.Апраксин // АСУ и приборы автоматики. — Харьков, 1999. — Вып. 110. — С. 3-6.
3. Experience with Formal Specification Using an Extended State Transition Model/ G.V. Bochmann, E.Cerny, V.Gague E.A. //IEEE Trans. – 1982. – Vol. COM-30, N12. – P.2506-2512.
4. Программная спецификация протокола DARPA INTERNET [Электронный ресурс]: – Электрон. текстовые дан. – Advanced Research Projects Agency, Sep.1981. Режим доступа: <http://rfc.com.ru/rfc793.htm>, свободный.

УДК 004.7

А.А. Сукачев, магистрант**Научный руководитель: И.В. Кудрявченко, канд. техн. наук, доц***Севастопольский государственный университет***АНАЛИЗ САМООРГАНИЗУЮЩИХСЯ MESH СЕТЕЙ**

Аннотация: Проведен анализ состояния самоорганизующихся сетей на примере MESH сетей. Приведено сравнение характеристик основных технологий доступа к беспроводной среде.

Ключевые слова: самоорганизующиеся сети, MESH сети, технологии доступа

Annotation: The analysis of a condition of the self-organizing networks on the example of MESH networks have been carried out. Comparison of characteristics of the main technologies of access to the wireless environment have been given.

Key words: self-organizing network, MESH networks, access technology

В настоящее время количество мобильных телекоммуникационных устройств заметно увеличивается. К ним относятся смартфоны, планшеты, ноутбуки и т.п. Эта тенденция будет сохраняться в будущем. В таком случае технологии для построения беспроводных самоорганизующихся сетей займут лидирующие позиции.

Понятие самоорганизация представляет собой процесс упорядочения элементов одного уровня в системе за счет внутренних факторов, без внешнего специфического воздействия [1]. Следовательно, самоорганизующиеся сети – это такие сети, которым не требуется никакой дополнительной инфраструктуры кроме самих узлов [3].

В данной статье рассмотрен пример с самым распространённым видом самоорганизующихся сетей – MESH сетями. Их главной особенностью является ячеистая структура. Схема централизованной MESH сети приведена на рисунке 1.

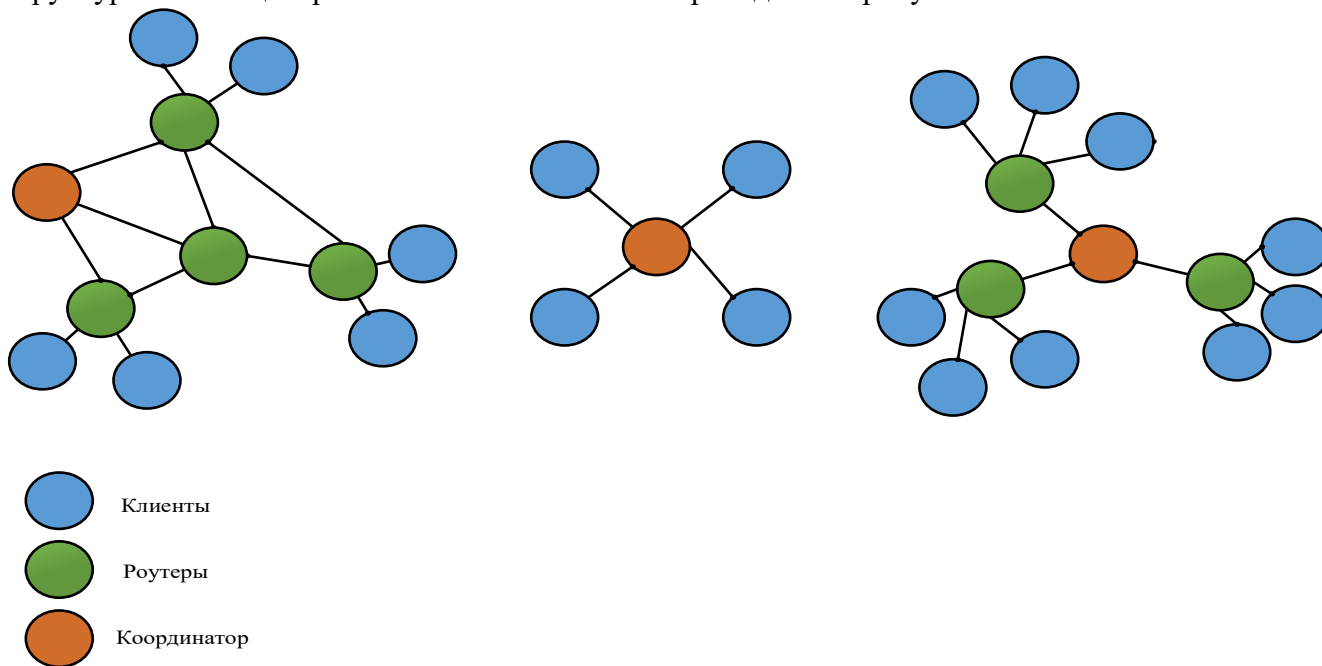


Рис. 1 Структурная схема MESH сети (Произвольная, Звезда, Кластерное дерево)

Существует три типа узлов в MESH сетях: клиент, роутер, координатор.

Клиенты – устройства конечного пользователя, такие как: ноутбук, смартфон, планшет и другие. Клиент может получить доступ в сеть для того, чтобы использовать ресурсы сети. Могут быть соединены с координаторами и/или роутерами.

Роутеры – устройства, которые находятся в сети, чтобы управлять сетевыми соединениями. Могут быть соединены с другими роутерами, координаторами и клиентами.

Координатор – главный узел в схеме сети, отвечает за построение начальной топологии и последующее её построение, и другие необходимые настройки для поддержания работоспособности сети.

Отличительной особенностью MESH сетей является динамически изменяющаяся топология сети, а небольшие размеры устройств накладывают свои ограничения на мощность передающего сигнала оборудования, небольшую емкость аккумуляторов и прочие. Поэтому очень важно максимально уменьшить влияние данных факторов за счет используемых оптимальных технологий доступа и алгоритмов построения.

Для построения самоорганизующихся сетей активно используют уже существующие технологии беспроводного доступа: IEEE 802.11 (Wi-Fi), IEEE-802.15.1 (Bluetooth), IEEE 802.15.4 (ZigBee) и др. [3]. Технологии выбираются в зависимости от назначения такой сети.

На рисунке 2 приведено сравнение технологий доступа по максимально доступной полосе пропускания и энергопотреблению [4].

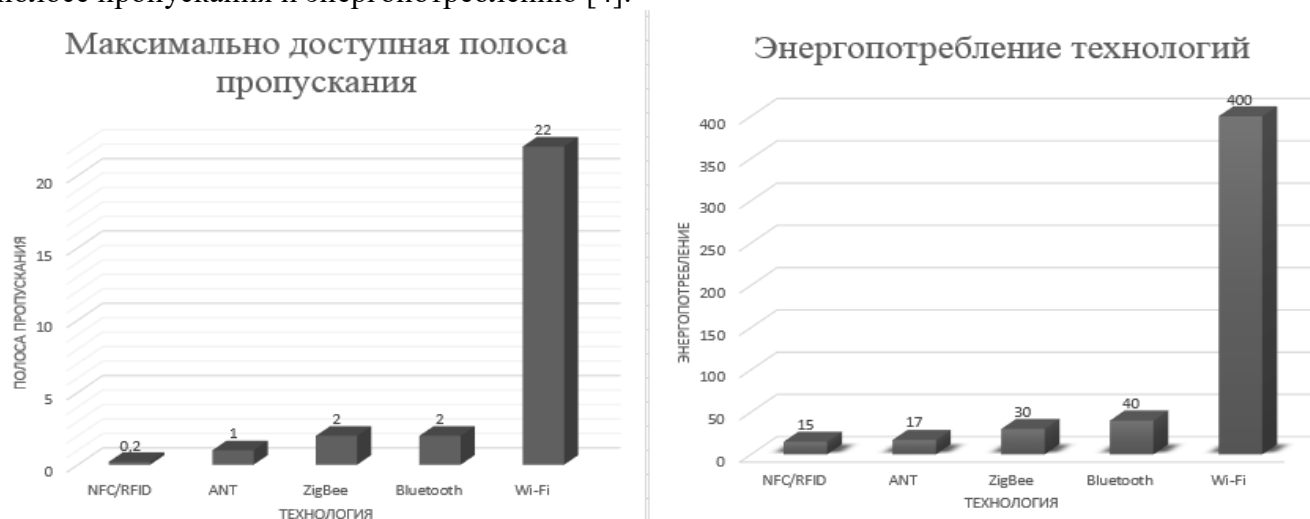


Рис. 2 Максимально доступная полоса пропускания технологий (МГц) и энергопотребление технологий (mA) в самоорганизующихся сетях

Опираясь на приведенные диаграммы, можно сделать вывод, что если главной характеристикой сети выступает максимальная полоса пропускания, то необходимо отдать предпочтение Wi-Fi. При этом не стоит забывать об обратной стороне медали, а именно энергопотреблении. Наиболее экономичными в этом плане являются технологии NFC, ANT, ZigBee. Важным аспектом при построении сетей является формирование топологии сети (установка физических связей между узлами), а также управление и координирование узлов в топологии. Решения этих задач содержатся в протоколах маршрутизации. Рассмотрение различных протоколов и их сравнение планируется в будущих публикациях.

Библиографический список

1. Евланов М.В., Неумывакина О.Е., Карамышева А.Ю. 2013. Анализ возможностей применения подходов к самоорганизации отдельных сервисов в сервис-ориентированных информационных системах. Вестник Национального технического университета. № 16. с 22-26.
2. Хоров Е.М. Знакомство с современными беспроводными технологиями. Многошаговые беспроводные сети: принципы построения и открытые задачи. Электронный ресурс. Режим доступа: <https://mipt.ru/drec/upload/d27/multihop-wireless.pdf>
3. Петров Д. 2010. Стандарты беспроводной связи диапазона ISM. Электронные компоненты. № 10: 28-32.
4. Pothuganti, K., Chitneni A. 2014. A Comparative Study of Wireless Protocols: Bluetooth, UWB, ZigBee, and Wi-Fi. Advance in Electronic and Electric Engineering. 4(6): 655–662

СЕКЦИЯ 7

Web – разработка



УДК 004.7

Д.В.Анисимова, Н.А. Дергачев, студенты 4-го курса**Научные руководители: В.С.Чернега, доцент, кандидат тех. наук; А.Ю. Дрозин, ст.преподаватель.***Севастопольский государственный университет***МОБИЛЬНОЕ ПРИЛОЖЕНИЕ ДЛЯ ПОСЛЕОПЕРАЦИОННЫХ ПАЦИЕНТОВ**

Аннотация: В данной статье описывается мобильное приложение для послеоперационных пациентов, находившихся в «Поликлинике Городской больницы №9» г. Балаклавы. Авторы описывают актуальность разрабатываемого мобильного приложения, подчеркивают его преимущества по отношению к другим существующим мобильным приложениям. Рассмотрены основные функции, которые будет выполнять приложение и рассматривается его интерфейс.

Ключевые слова: врач, пациент, приложение, напоминание.

Annotation: This article describes a mobile application for postoperative patients who were in the Polyclinic of City Hospital No. 9 in Balaklava. The authors describe the relevance of the mobile application being developed, emphasizing its advantages with respect to other existing mobile applications. The main functions that the application will perform and its interface are examined.

Keywords: doctor, patient, reminder.

На данный момент существует множество различных приложений для контроля своего здоровья. Современные приложения позволяют людям отслеживать свой вес, давление, считать расходуемые калории и пройденные километры, контролировать уровень сахара и качество сна. Однако не все приложения одинаково полезны, так как большинство из них можно больше отнести к развлекательным, а не медицинским. Много людей, выписавшихся из больницы, забывают соблюдать определенный режим, составленный ему врачом, а так же пренебрегают предварительной записью к врачу.

Для решения этих проблем разрабатывается мобильное приложение для пациентов, пребывавших на стационаре в урологическом отделении Городской больницы №9 Балаклавы. При выписке эти пациенты получили список рекомендаций врача и для строгого соблюдения выполнения всех этих рекомендаций им будет помогать наше приложение.

Сейчас существует множество приложений – напоминаний для приема таблеток и выполнения каких-либо процедур, которые для себя будет назначать сам пациент. Однако такие приложения имеют недостаток: действия пациента не будут контролироваться квалифицированным специалистом.

Наше приложение позволяет врачу дистанционно назначать список рекомендаций при выписке и обладает возможностью контролировать выполнение пациентом всех его рекомендаций. Для реализации этого функционала приложения будет существовать серверная часть, которая будет располагаться в больнице. В мобильном приложении для напоминания о приеме медикаментов, проведения процедур, а так же записи к врачу, пациенту, или пользователю, на дисплее экрана телефона будет отображаться уведомление, сопровождаемое звуковым сигналом.

Разрабатываемое приложение имеет интуитивно-понятный интерфейс, что делает его понятным для пользователя любого возраста и любой квалификации. Приложение адаптируется под различные форматы экранов мобильных гаджетов, но под платформу Android.

На главной странице приложения расположены основные пункты меню:

- Медикаменты
- Процедуры
- Анализы
- Визиты к врачу
- Рекомендации
- Настройки.

На странице «Медикаменты» пользователю отображается список принимаемых лекарств, так же их дозировка и время принятия.

Страница «Процедуры» имеет список процедур, рекомендуемых пациенту, время их проведения, а так же дополнительные сведения о проведении процедуры.

Страница «Анализы» включает в себя список анализов, которые будет необходимо пройти пациенту, а так же предоставлена возможность просмотреть расписание сдачи определенных анализов, подробной информации о сдаче анализов, возможность установки напоминания об их сдаче.

На странице «Визиты к врачу» пользователь имеет возможность видеть список всех рекомендуемых ему врачей, просматривать их расписание, а так же предварительно записываться на прием с помощью веб-сайта doctor92, на страницу которого перенаправляет приложение. Приложение также напоминает пациенту о дне посещения врача.

На странице «Рекомендации» расположен режим дня пациента, который будет составляться врачом и под который будет подстраиваться прием медикаментов.

В «Настройках» пациент может отключать/включать звуковые уведомления, выходить из своего аккаунта.

Приложение имеет двустороннюю связь: врач больницы получает данные о всех пациентах, о выполнении пациентом полученных рекомендаций. Все данные, которые будет получать врач, будут храниться на сервере либо хостинге. Сервер будет представлять некую программу, работающую на удаленном компьютере и реализующую функционал взаимодействия с приложением – клиентом, то есть нашим мобильным приложением. Разработанный Web-сайт будет инициализировать врача как администратора и он, в свою очередь, сможет заполнять рекомендации пациентам дистанционно.

В качестве формата взаимодействия сервера с мобильным приложением использовался формат JSON. Для разработки серверной части был выбран «Django REST framework» языка Python. В качестве сервера базы данных используется свободно распространяемая реляционная СУБД MySQL.

Для разработки мобильного приложения используется «Android sdk» версии 24. В качестве мобильной базы данных используется кроссплатформенная база данных «Realm». Для взаимодействия мобильного приложения с сервером используется библиотека «Retrofit».

В завершение можно сказать, что разрабатываемое приложение актуально, так как оно будет высылать пациенту-пользователю напоминания о приеме лекарств, проведении процедур, а так же запланированном посещении врача, что поможет пациенту более строго соблюдать рекомендации, следить за своим здоровьем и, в свою очередь, экономить время врача.

УДК 004.7

А.И. Батраченко, студент 4-го курса**Научный руководитель: В.А. Строганов, старший преподаватель***Севастопольский государственный университет***WEB-ОРИЕНТИРОВАННАЯ СИСТЕМА ПОСТРОЕНИЯ И АНАЛИЗА
РАСПИСАНИЯ ЗАНЯТИЙ**

Аннотация: В данной статье рассмотрена web-ориентированная система, целью которой является построение и анализ расписания занятий. Разработанная система позволяет не только эффективно создать расписание для любой группы учащихся на семестр, но и предотвратить множество ошибок и недочетов, допускаемых при ручном конструировании расписаний. Результат достигается за счет удобного визуального представления, а также автоматического анализа учебного плана преподавателей и проверки на наличие свободных аудиторий.

Ключевые слова: Расписание, web-сервис, анализ данных

Annotation: This article is devoted to a web-based system, the purpose of which is to build and analyze timetable. The developed system allows not only to efficiently create a schedule for any group of students for a semester, but also to prevent a lot of mistakes and shortcomings done in the manually constructed schedules. The result is achieved through a convenient visual presentation, as well as automatic analysis of the curriculum of teachers and checking classrooms availability.

Key words: Timetables, web-service, data analysis

Составление расписания в высших учебных заведениях – длительный, трудоемкий и энергозатратный процесс. В связи с необходимостью обработки большого количества информации в сжатые сроки, на составление расписания оказывает непосредственное влияние человеческий фактор. В результате чего, из года в год процесс сведения общего расписания и расчета почасовой нагрузки на членов преподавательского состава несколько затягивается. Работники деканата вынуждены решать организационные вопросы дольше запланированного срока, а преподаватели и студенты периодически сталкиваются с неточностями в расписании занятий.

Идея создания web-ориентированной системы для построения и анализа расписания занятий заключается в решении этой проблемы. Цель проекта заключается в том, чтобы упростить и автоматизировать этот процесс для высших учебных заведений. Особого внимания заслуживает аспект простого доступа к расписанию для преподавателей и студентов.

Апробация данной системы планируется на базе кафедры информационных систем.

Для реализации проекта были использованы следующие технологии:

Web-фреймворк Ruby on Rails — представляет собой многоуровневый MVC-фреймворк для построения web-приложений, использующих реляционные и NoSQL базы данных (например, MySQL, MariaDB, PostgreSQL, MongoDB). Фреймворк написан на языке программирования Ruby. Благодаря активно развивающемуся сообществу программистов Ruby on Rails существует большая база решений для использования различных технологий, что позволяет существенно сократить время на разработку веб-приложений.

Ruby — интерпретируемый, объектно-ориентированный язык программирования «сверхвысокого уровня» с динамической типизацией и встроенным «сборщиком мусора». Одним из принципов, положенных в основу его проектирования, заключается в том, что компьютер должен работать на человека, а не наоборот.

RSpec — это фреймворк для тестирования написанный на Ruby. Он предоставляет специальный DSL (Domain Specific Language — предметно ориентированный язык) для написания тестов — спецификаций. RSpec — это инструмент для BDD. BDD (Behavior-Driven Development) — разработка с опережающим описанием поведения программы.

Vue.js — фреймворк для разработки пользовательских интерфейсов, реализующий MVVM паттерн (ViewModel). На Vue.js значительно повлияли Angular, Knockout, React и Rivets. Несмотря на сходства, Vue.js может предложить ценную альтернативу этим существующим библиотекам, в поисках золотого сечения между простотой и функциональностью.

В ходе разработки системы были выделены роли пользователей данной системы – гость и администратор. Гость может просмотреть общую информацию о системе (статические страницы), а также ознакомиться с расписанием по запросу для группы/потока, преподавателя, аудитории.

В свою очередь среди возможностей администратора выделяются:

- просмотр списка групп, предметов, аудиторий с возможностью добавления, редактирования, удаления, поиска; просмотр списка преподавателей с возможностью удаления;
- отправка пригласительного письма преподавателям со ссылкой на форму регистрации, по указанному заранее почтовому ящику;
- просмотр списка администраторов с возможностью удаления. Кроме того он может отправлять пригласительное письмо со ссылкой на форму регистрации новым администраторам;
- создавать расписание для конкретной группы исходя из наличия свободных аудиторий, совместимости пар преподавателей и учетом их нагрузки.
- выгружать расписание группы в Excel или PDF файл.

Такому набору функций можно поставить в соответствие концептуальную схему базы данных, изображенную на рисунке 1.



Рис. 1. Концептуальная схема базы данных

Заключение

Благодаря реализации вышеперечисленных функций достигается цель данного проекта, которая заключается в рационализации процесса составления расписания для высшего учебного заведения. Кроме того, система предоставляет постоянный доступ к электронной версии расписания для сотрудников учебного заведения и учащихся. Также предполагается дальнейшая работа по разработке данного проекта, связанная с улучшением интерфейса и частичным перепроектированием архитектуры приложения.

УДК 004.7

Д.Н.Спорыш, студент 4-го курса

Научный руководитель: В.А.Строганов, ст. преподаватель.

Севастопольский государственный университет

ИНФОРМАЦИОННАЯ СИСТЕМА ДЛЯ МЕДИЦИНСКОГО ПЕРСОНАЛА И ПАЦИЕНТОВ

Аннотация: В данной статье описывается информационная система для «Поликлиники Городской больницы №9» г. Балаклавы. Автор описывает актуальные проблемы текущего рабочего процесса в медицинском учреждении и способы их решения с помощью разрабатываемой системы. Рассмотрены функции, которая обеспечивает система. Рассматривается общая структура информационной системы и используемые технологии. Было доказано, что выбранный архитектурный стиль RESTful API позволяет успешно реализовывать функционал системы как для приложения на ПК, так и под мобильные приложения. Также описываются перспективы развития проекта, возможность его расширения и ввод в эксплуатацию в других больницах.

Ключевые слова: медицина, веб-приложения, информационная система, медицинские учреждения, пациенты, мобильное приложение, медицинская информационная система.

Annotation: This article describes the information system for Polyclinics of City Hospital No. 9 in Balaklava. The author describes the actual problems of the current workflow in a medical institution and the ways to solve them with the help of the developed system. The functions that the system provides are considered. The general structure of the information system and the technologies used are considered. It was proved that the chosen architectural style of the RESTful API allows to successfully implement the system functionality both for the application on the PC and for mobile applications. It also describes the prospects for the development of the project, the possibility of its expansion and commissioning in other hospitals.

Key words: Medicine, web applications, information system, medical institutions, patients, mobile application, medical information system.

В настоящее время в медицине существует ряд проблем, из-за которых сильно тормозится рабочий процесс и снижается качество лечения пациентов в целом. Одной из основных проблем является необходимость в оперативном доступе ко всей необходимой информации (о пациентах, динамике их лечения, анализах, процедурах, операциях, осмотрах и т.д.). И, что не мало важно, поддержание актуальной информации для всех пользователей системы. Как пояснили нам сами сотрудники данной больницы, они большую часть времени работы бегают по отделению, операционным, палатам и различным кабинетам. Поэтому для них иметь оперативный доступ к необходимой актуальной информации очень важно.

Отправляясь на ежедневную процедуру обхода пациентов лечащим врачом и медсестрами, медицинские работники должны иметь полную информацию по проведенным за день обследованиям, по готовым анализам и прочие данные. При этом, одной из основных проблем является обилие бумажной документации в виде историй болезней пациентов, к которым невозможно обеспечить оперативный доступ должным образом. Кроме этого, если часть каких-либо анализов или рентген-снимков уже сделана, то они могут быть еще не подклеены в истории болезней пациентов. Отсюда вытекают проблемы поддержания актуальности информации.

Даже при наличии разработанного сайта для стационарных компьютеров, уходит много времени на перенос информации из бумажных документов в электронную систему. Таким образом, происходит продлевание одной и той же работы два раза.

Проанализировав эту и прочие проблемы, было установлено, что одна из основных задач – реализация централизованной системы, как для клиентского приложения на стационарных компьютерах, так и для мобильных приложений на планшетах и смартфонах.

Функции, которые может обеспечивать разрабатываемая система [1]:

- взаимодействие различных структур больницы и мед. персонала;
- структурированное и наглядное представление информации (списка пациентов, работников, процедур, анализов, операций и т.д.);
- автоматизированный прием, выписку, лечение и уход за пациентами;
- возможность оперативного доступа и изменения информации о пациенте, его динамике лечения, анализах, процедурах, операциях, назначениях, осмотрах и т.д.;

оперативное информирование о состоянии больного (на текущий момент и за определенный период);

информирование о диагнозе, результатах предварительного обследования, процессе лечения;

автоматизация учета нахождения в лечебном учреждении (составление истории болезни, выписного эпикриза и т.д.).

Общая структура взаимодействия компонентов разрабатываемой информационной системы представлена на рисунке 1.

Для серверной части данной системы необходимо было реализовать функционал независимо от реализации пользовательского интерфейса. Для этого необходимо предоставить некоторый веб-интерфейс, посредством которого клиентская часть системы сможет взаимодействовать с её серверной частью. Для обеспечения указанного требования в качестве стиля архитектуры разрабатываемой системы использовался RESTful API. Благодаря такому подходу, можно реализовать отдельно клиентское приложение на ПК, отдельно приложения под смартфон, и при этом все они смогут взаимодействовать с серверной частью системы и использовать ее функционал независимо друг от друга.

В качестве формата взаимодействия с системой пользовательского интерфейса использовался формат JSON. Для разработки серверной части был выбран "MVC-Framework Laravel" языка PHP. В качестве сервера базы данных используется свободно распространяемая реляционная СУБД MySQL. Основным инструментом для разработки клиентской части выступает AngularJS – фреймворк с открытым исходным кодом.

При успешной реализации проекта и дальнейшем сотрудничестве нашей кафедры с данной поликлиникой, мы планируем обеспечить дальнейшую поддержку и развитие проекта.

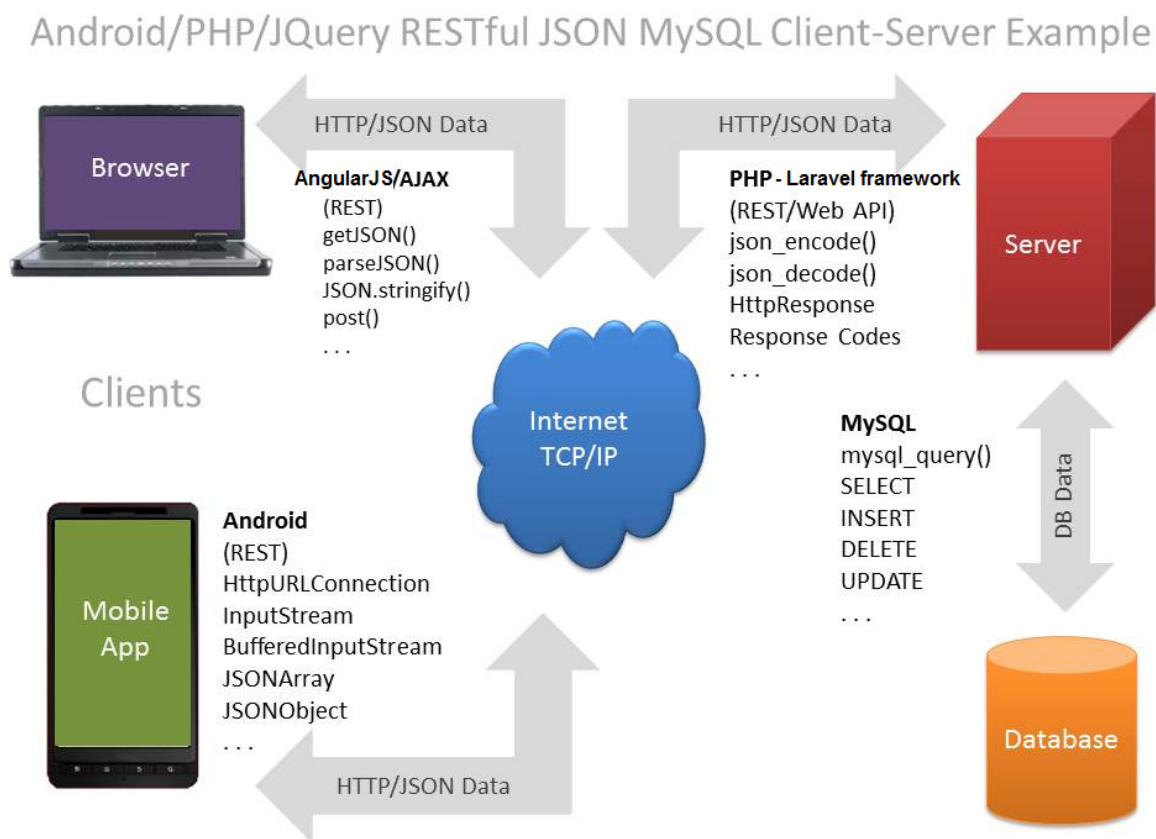


Рис.1. Общая структура взаимодействия компонентов медицинской информационной системы

Перечь функций, который мы планируем внедрить при развитии проекта, следующий:

1. Реализация графического плана отделений со всеми палатами. В каждой палате визуально будут отображаться койки различных цветов, в зависимости от ее занятости. При наведении указателя на койку будет выводиться информация о пациенте и возможная дополнительная информация.
2. Возможность конфигурирования расположения коек и их количества с помощью графического редактора посредством мыши.
3. Голосовое управление мобильным приложением для обеспечения быстрого поиска пациентов и различной необходимой информации.
4. Внедрение меж персональной переписки для более удобного и быстрого взаимодействия мед персонала.
5. Ведение мед документации в электронном формате.
6. Построение различных отчетов по запросам персонала.
7. Дополнительные графики, диаграммы и таблицы различных статистик работы больницы.
8. Дополнительные функции по желанию заинтересованных сторон и техническое обслуживание.

Следует также отметить, что данную систему можно интегрировать и в другие больницы. Поэтому мы нацелены на сотрудничество с теми, кто в этом заинтересуется.

Рассматриваемая тема актуальна, так как такая система позволяет ускорить процесс взаимодействия медперсонала и различных подразделений медучреждения, улучшить качество лечения и ухода за пациентами, а также составлять различного рода отчёты по проведенной работе, что позволит сэкономить много времени и бумажных ресурсов.

Библиографический список

1. Назаренко Г.И., Гулиев Я.И., Ермаков Д.Е. Медицинские информационные системы: теория и практика / Под ред. Г.И. Назаренко, Г.С. Осипова. – М.: ФИЗМАТЛИТ, 2005. – 320с.

УДК 681.3

С.С. Сысоев, студент 4-го курса, В.Д. Полетаева, студентка 4-го курса**Научный руководитель: А.Л. Овчинников, ст. преподаватель***Севастопольский государственный университет***СИСТЕМА КОНТРОЛЯ УСПЕВАЕМОСТИ И ПОСЕЩАЕМОСТИ СТУДЕНТОВ**

Аннотация: современная web-ориентированная система предоставляет удобный интерфейс и инструментарий для отслеживания состояния учебного процесса студентов, выставления оценок и пропусков, создания сводных таблиц и ведомостей.

Ключевые слова: успеваемость, документооборот, учебный процесс, web-приложение.

Annotation: modern web-based system provides useful interface and tools for monitoring educational process of students, making assessment and looking after their absence, it also provides compiling summaries for student groups.

Key words: academic progress, document, educational process web-application.

В эпоху компьютеризации и информатизации очень важную роль на любом предприятии играют удобные инструменты для ведения электронной отчётности и документооборота. Наличие подобного инструментария помогает значительно сократить время, затрачиваемое на выполнение повседневной рутинной работы по составлению отчётов, заполнению ведомостей и бланков, позволяя тем самым направить освободившееся время на выполнение непосредственных обязанностей сотрудниками, что приводит к повышению производительности предприятия в целом.

Сотрудники университета, будучи работниками государственного учреждения, вынуждены вести строгую отчётность по успеваемости и посещаемости студентов. На данный момент в большинстве высших и средне-специальных учебных заведений традиционно пользуются бумажными носителями для хранения оценок и пропусков занятий студентами, что далеко не всегда является удобным, особенно при оценивании студентов в рамках одной дисциплины разными преподавателями.

Проведя опрос среди преподавателей и сотрудников кафедры "Информационные системы" было принято решение разработать автоматизированную информационную систему, решающую следующие функциональные задачи:

- ввод и редактирование учебных планов кафедры, для получения списка дисциплин и видов работ по дисциплинам;
- управление профилями студентов и сотрудников кафедры;
- разграничение полномочий (сотрудник деканата, преподаватель, студент);
- ввод, редактирование и отображение посещаемости студентов;
- управление профилями студентов и сотрудников кафедры;
- разграничение полномочий (сотрудник деканата, преподаватель, студент);
- автоматическая генерация итоговых ведомостей;
- инфографика сдачи работ студентами.

Приложение, имеющее подобные функции, является отчасти стандартной задачей современного разработчика, для успешной реализации которой необходимо в соответствии с результатом анализа предметной области выбрать:

- язык или языки программирования и средства разработки;
- наиболее подходящую СУБД;
- архитектуру приложения;
- платформы, на которых планируется запускать приложение.

Проанализировав поставленные требования было принято решение разработать web-ориентированное приложение, так как современные web-приложения являются кросс-платформенными (что уменьшит затраты на разработку под конкретную платформу), также современные языки web-программирования предоставляют весь необходимый функционал для быстрой и качественной разработки, а также масштабируемости.

Для хранения данных была выбрана реляционная СУБД MySQL, так как разрабатываемая система не предполагает необходимости очень быстрого доступа к данным, а MySQL поддерживается большинством платформ по развёртыванию web-приложений.

Языком программирования для серверной части приложения был выбран PHP. Это один из самых популярных языков для разработки web-ориентированных приложений, имеет все необходимые функции и множество библиотек для расширения функциональности. Языком разработки логики клиентской части был выбран традиционный JavaScript с использованием библиотеки "jQuery", препроцессором для каскадных таблиц стилей был выбран SCSS, так как он позволяет упростить написание CSS-правил и ускорить процесс разработки.

В качестве каркаса будущего web-приложения был выбран один из наиболее популярных фреймворков "Laravel", так как он имеет хорошую документацию, множество расширений и предоставляет весь необходимый базовый функционал, не ограничивая свободы действий. В базовой конфигурации присутствует шаблонизатор "blade", который помогает упростить разработку шаблонов. Он использует помимо стандартных HTML-тегов разметки документа собственные (например операторы циклов или условные операторы).

Исходя из выбранного инструментария, был применён архитектурный шаблон проектирования "MVC" (Model-view-controller) для жёсткого разделения логики и представления web-приложения. Данный шаблон является одним из самых удачных для применения в web-разработке, так как с подобным разделением логики и представления намного ускоряется процесс разработки за счёт хорошо структурированного кода.

Было принято решение использовать подход к разработке, предполагающий вынесение всей бизнес-логики приложения в отдельные библиотеки, что позволило избавиться от написания большого количества программного кода в контроллерах и моделях. Таким образом, все логические части шаблона проектирования выполняют только те действия, которые им делегированы: маршрутизатор обращается к заданному контроллеру, который передаёт данные модели методу соответствующей библиотеки, и возвращает результат, который выводится на представление.

Также, было принято решение придерживаться парадигмы REST, что позволит использовать данное web-приложение в качестве поставщика данных для сторонних приложений (например, полноценного мобильного приложения).

Исходя из современных тенденций web-разработки был выбран ряд пакетных менеджеров: для серверной части – "composer", для клиентской – "npm" и "bower". Они позволяют устанавливать необходимые компоненты одной командой, не сохраняя файлы в репозитории системы контроля версий, а только название пакета и его версию, что уменьшает общий размер проекта.

Для сборки всей клиентской логики в один файл и для компиляции SCSS-файлов был выбран инструмент "Gulp", так как это один из самых быстродействующих сборщиков. Более того, он имеет очень простой синтаксис и интеграцию со средой разработки, что делает написание программного кода ещё более удобным и быстрым.

В качестве платформы разработки использовалась операционная система семейства "Linux", так как изначально это семейство разрабатывалось для работы с сетевыми ресурсами, а web-ориентированное приложение таковым и является.

Определив необходимое программное обеспечение и инструментарий, была спроектирована база данных, позволяющая масштабировать проект, не прибегая к значительным изменениям структуры базы или изменению уже разработанных программных компонентов. Это достиглось путём проведения нормализации, введения избыточности, определения независимых сущностей системы, данные о которых дополняются введением соответствующих таблиц, которые также являются независимыми. Спроектированная база данных позволяет уменьшить степень связности системы, прибегнуть к полноценному использованию компонентного подхода разработки приложений.

Исходя из спроектированной таким образом базы, разработка велась с максимально возможным использованием модульной структуры приложения, каждая сущность описывалась отдельной моделью данных, а доступ к полной информации о необходимой сущности осуществляется с помощью внешних ключей и определения отношений между независимыми моделями.

Благодаря следованию REST-парадигме и применению шаблона проектирования "MVC", разработанное приложение не имеет жёсткой привязки к интерфейсу пользователя, что также является примером модульной структуры. Представление только отображает полученные данные с серверной части, что позволяет изменять его без угрозы нарушения логики работы всего проекта. Представление может являться и вовсе отдельным сторонним приложением, разработанным с использованием любого другого языка программирования, технологий и методологий, а получать данные в соответствии со спецификацией внешних интерфейсов web-приложения.

Результатом работы является web-ориентированное приложение, в котором реализованы следующие функции для преподавателя и сотрудника деканата:

- создание институтов, профилей, учебных планов;
- создание академических групп;
- назначение учебных планов группам;
- создание произвольного количества типов оценивания учащихся;
- выставление оценок по каждой работе;
- просмотр истории изменения оценок по конкретной работе;
- управление профилями пользователей и разграничением прав доступа;
- назначение преподавателям дисциплин.

Для студента реализован просмотр оценок по каждой изучаемой или изученной дисциплине по каждой сданной работе (расчётно-графическое задание, курсовой проект, лабораторная работа и так далее).

В настоящий момент разработанное приложение тестируется локально, исправляются ошибки и добавляется новый функционал. В ближайшем времени планируется произвести тестирование на сервере кафедры, обучить сотрудников и провести исследование о целесообразности внедрения подобного приложения.

Библиографический список

1. Laravel [Электронный ресурс] // <http://laravel.com>: официальный сайт Laravel. 2006-2016. Дата обновления 2016. URL: <http://laravel.com> (дата обращения: 24.03.2017)
2. Архитектура REST руководство по проектированию архитектуры REST [Электронный ресурс]. URL: <https://habrahabr.ru/post/38730/> (дата обращения: 24.03.2017)

УДК 004.7

П.Е. Шушпан, студент 4-го курса**Научный руководитель: В.А. Строганов, старший преподаватель***Севастопольский государственный университет***WEB-ОРИЕНТИРОВАННАЯ СИСТЕМА ЭЛЕКТРОННОГО ЖУРНАЛА
ПРЕПОДАВАТЕЛЯ**

Аннотация: В данной статье рассмотрена web-ориентированная система персонализированного журналирования для преподавателей вуза. Разработанная система позволяет создавать списки групп, рассчитывать индивидуальную нагрузку преподавателя, автоматизировать процесс аттестации студентов и анализа посещаемости. Электронный журнал оснащен удобным пользовательским интерфейсом и автоматизацией всех необходимых расчетов.

Ключевые слова: журналирование, персонализированный электронный журнал, автоматизация

Annotation: This article is devoted to a web-based system of personalized logging for university teachers. The developed system allows creating lists of groups, calculating the individual load of the teacher, automating the process of attestation of students and the analysis of attendance. The electronic log is equipped with a user-friendly interface and automation of all necessary calculations.

Key words: Logging system, personalized e-zine, automation

Ведение журнала является неотъемлемой частью учебного процесса. Каждый преподаватель вынужден регулярно заполнять журналы групп – вносить туда темы занятий, отмечать количество часов, фиксировать посещаемость, выставять баллы. На практике оказывается, что ведение данной документации становится весьма обременительным. Кроме того, чтобы систематизировать информацию о каждой из дисциплин, групп, программ, каждому преподавателю необходимо вести дополнительные записи – вносить в личные ежедневники списки студентов, тем, расписание занятий. Учитывая влияние человеческого фактора, в ходе семестра часть данных может быть утеряна либо субъективно интерпретирована.

Цель web-ориентированной системы персонализированного журналирования для преподавателей вуза позволяет решить данные проблемы путем автоматизации ведения необходимой документации и личных записей. Электронный журнал позволит структурировать и систематизировать информацию о ходе учебного процесса в конкретной группе и свести количество спорных моментов в период аттестации к минимуму.

Разработана система на основе web-ориентированной системы построения и анализа расписания занятий. Внедрение системы электронного журнала запланировано на базе кафедры вуза.

Для реализации проекта были использованы перечисленные технологии:

Высокоуровневый язык Ruby, используемый для удобного и быстрого объектно-ориентированного программирования. Его характеризуют рефлексивность и динамичность.

Полноценный многоуровневый web-фреймворк Ruby on Rails. Данный web-фреймворк используется для построения веб-приложений, которые используют базы данных. Он базируется на архитектуре Модель-Представление-Контроллер (Model-View-Controller, MVC). Его преимуществом является динамичный AJAX-интерфейс, возможность обработки запросов и выдача данных в контроллерах, предметная область и отраженная в базе данных.

Библиотека Sidekiq для Ruby on Rails. Она используется для обработки задач, занимающих длительное время. Что позволяет пользователю не завершать запущенные ранее задачи. Библиотека используется в связке с сетевой журналируемой нереляционной базой данных, которая хранит данные в оперативной памяти. Кроме того она оснащена механизмом обмена сообщениями в фоновых процессах.

Решение RSpec — удобное, обладающее «человекопонятным» синтаксисом, решение позволяет применить в создании проекта принципы разработки через тестирование, тем самым документируя его и значительно упрощая отладку. Данный подход позволяет обойти подробное проектирование системы, так как сам по сути и декларирует ее поведение.

Данная система позволит преподавателю просматривать расписание на текущий день и неделю; формировать рабочий план по предмету, путем добавления форм работы (лабораторных работ, расчетно-графических заданий, рефератов, контрольных работ и т.д.); выставлять оценки и отмечать отсутствующих студентов для конкретной пары; формировать аттестации; просматривать статистическую информацию по каждой группе. (оценки, пропуски). Также он может вычислять итоговые баллы для каждого студента группы и отмечать в расписании выходные дни (праздники).

Кроме того, преподаватель может загружать excel файл либо создавать вручную список студентов для конкретной группы, куратором которой он является.

В ходе разработки была спроектирована концептуальная схема базы данных, которая изображена на рисунке 1.

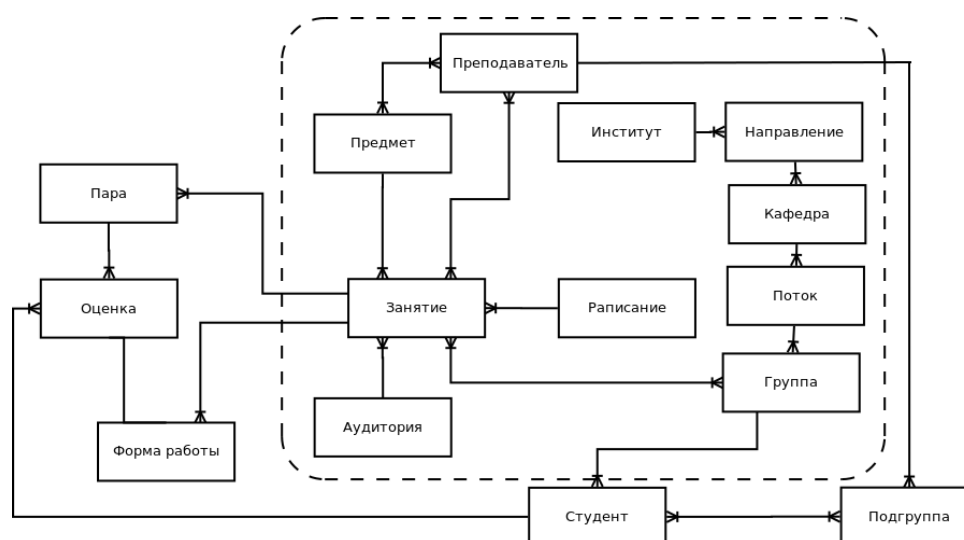
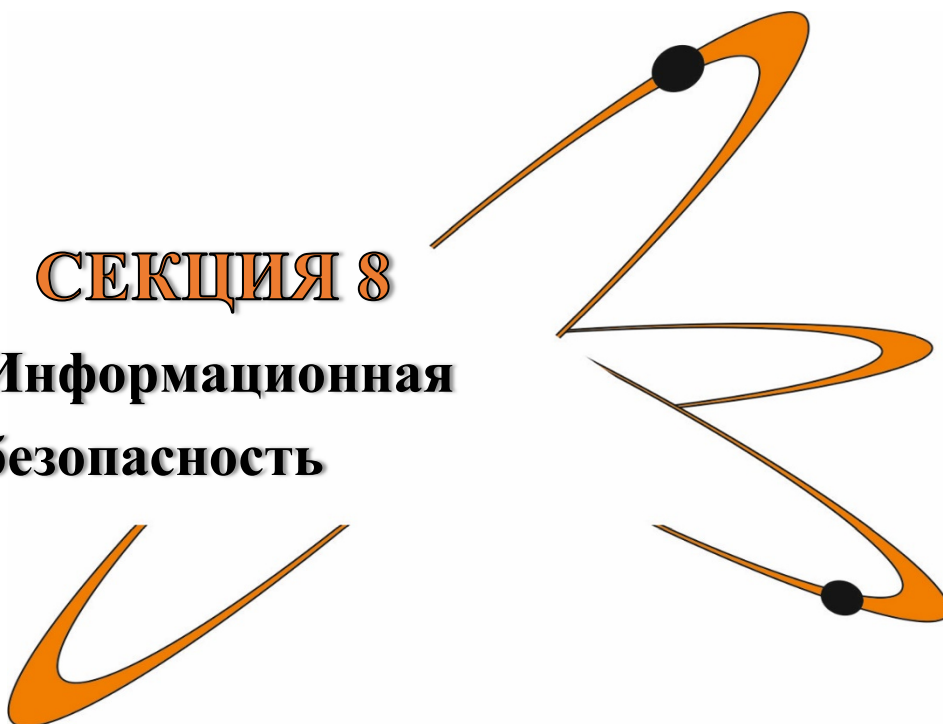


Рис. 1. Концептуальная схема базы данных

Итак, благодаря функциям, доступным в рамках данной web-ориентированной системы, полностью реализуется цель данного проекта. Удобство и простота использования электронного журнала позволят полностью автоматизировать процесс аттестации студентов и систематизировать информацию о ходе учебного процесса. В ходе внедрения данной системы в практику вуза, будут внесены необходимые коррективы для максимально рационального и комфортного использования электронного журнала.

СЕКЦИЯ 8
**Информационная
безопасность**



УДК 681.3

Р.Г.Вихров, студент 2-го курса магистратуры**Научный руководитель: С.Н.Фисун, канд. техн. наук, доцент***Севастопольский государственный университет***ИССЛЕДОВАНИЕ БЫСТРОДЕЙСТВИЯ СИММЕТРИЧНЫХ АЛГОРИТМОВ БЛОЧНОГО ШИФРОВАНИЯ**

Аннотация: Для обеспечения безопасности данных используются симметричные и асимметричные алгоритмы шифрования, которые преобразуют данные в формат, непригодный для чтения. В данной статье представлен сравнительный анализ быстродействия шифрования симметричных алгоритмов AES, Blowfish, CAST5 и 3DES. Рассматриваются основные понятия теории шифрования информации, преимущества использования симметричных алгоритмов, описаны достоинства и недостатки программной реализации криптографических алгоритмов.

Ключевые слова: информационная безопасность, защита информации, криптография, алгоритм шифрования, симметричные алгоритмы, блочные алгоритмы, Triple-DES, 3DES, TDES, TDEA, AES, Blowfish, CAST5, CAST128.

Annotation: To ensure data security, symmetric and asymmetric encryption algorithms are used, which convert the data into a format that is not readable. This article presents a comparative analysis of symmetric algorithms AES, Blowfish, CAST5 and 3DES in terms of their encryption speed. Explains the basic concepts of the information encryption, the advantages of using symmetric algorithms, describes the advantages and disadvantages of software implementations of cryptographic algorithms.

Key words: information security, encryption, cryptography, algorithm, symmetric, block cipher, Triple-DES, TDES, TDEA, 3DES, AES, Blowfish, CAST5, CAST128.

В связи с быстрым развитием средств вычислительной техники, сетей передачи данных и постоянным внедрением компьютерных технологий в различные сферы человеческой деятельности становится актуальным вопрос безопасного хранения и передачи данных, циркулирующих в компьютерных системах.

Наиболее эффективным способом обеспечения конфиденциальности данных, их целостности и подлинности является применение криптографических методов. В основе криптографических методов лежит понятие криптографического преобразования данных, производимого по определенным математическим законам, с целью преобразования исходной информации в форму, недоступную для понимания потенциальным злоумышленником.

На рисунке 1 изображена традиционная схема криптосистемы.

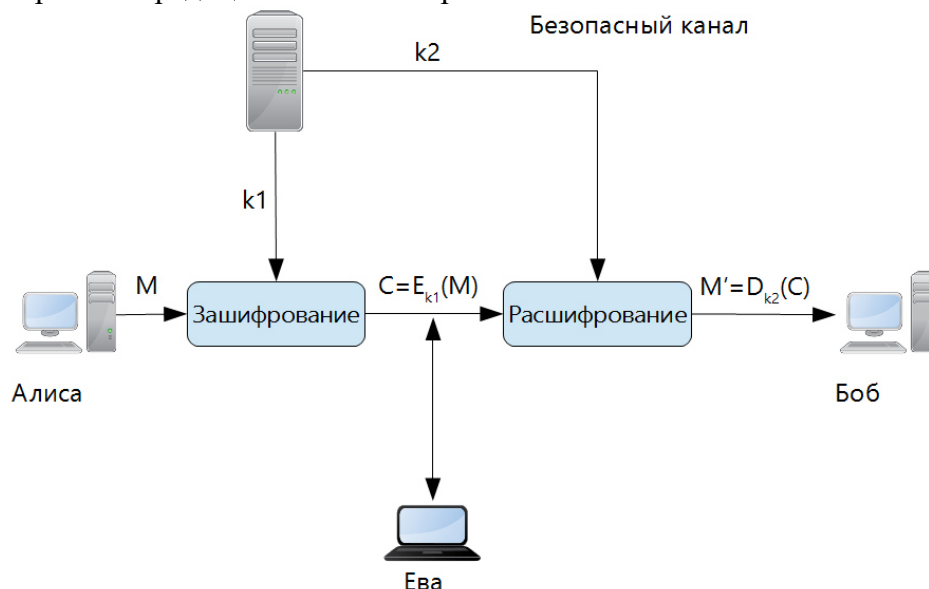


Рис.1. Традиционная криптосистема

Задача криптосистемы, приведенной на рисунке 1 состоит в том, чтобы Алиса смогла передать Бобу сообщение M , называемое открытым текстом. Для этого, при помощи ключа $k1$, Алиса сначала преобразует открытый текст M в так называемый шифротекст (криптограмму) C . Этот процесс называется зашифрованием. Если обозначить процесс зашифрования символом E , то в общем виде его можно записать формулой (1):

$$C = E_{k_1}(M), \quad (1)$$

Полученный шифротекст Алиса передает Бобу. Для того, чтобы Боб смог прочитать сообщение Алисы, он должен расшифровать полученный им шифротекст при помощи специального ключа расшифрования k_2 и функции расшифрования D . Обозначив информацию, полученную в результате расшифрования, символом M' , запишем процесс расшифрования в виде формулы (2):

$$M' = D_{k_2}(C), \quad (2)$$

Предполагается, что канал передачи между Алисой и Бобом является незащищенным. Поэтому злоумышленник Ева, присоединенный к линии связи, предпринимает попытки найти исходное сообщение M .

Под ключом понимают некоторую информацию, позволяющую выбрать один из множества возможных для данного алгоритма вариантов криптографических преобразований. Таким образом, ключ – это уникальный элемент, с помощью которого можно варьировать результат работы алгоритма шифрования: один и тот же открытый текст при использовании различных ключей будет зашифрован по-разному. Перед отправкой открытого текста Алиса и Боб должны знать конкретный выбор ключей и согласовать их значения посредством безопасного канала. Этот канал может быть курьером, но возможно также, что Алиса и Боб заблаговременно согласовали выбор ключей.

Алгоритмы симметричного шифрования — это такие алгоритмы шифрования, в которых для процессов зашифрования и расшифрования используется один и тот же ключ [1], то есть применительно к схеме, изображенной на рисунке 1: $k_1 = k_2$.

Симметричное шифрование бывает двух видов: блочное и потоковое. Именно симметричные алгоритмы блочного шифрования получили наибольшее распространение в компьютерном мире, поскольку имеют ряд преимуществ, среди которых следует отметить:

- возможность эффективной реализации;
- высокую скорость зашифрования/расшифрования как при аппаратной, так и при программной реализации;
- высокую гарантированную стойкость [2].

На эффективность функционирования систем шифрования влияют два основных параметра: скорость шифрования информации и временная сложность вскрытия алгоритма (криптостойкость).

Криптостойким считается алгоритм шифрования, вскрытие которого практически невозможно при текущем уровне развития вычислительной техники, или атакующему необходимо обладать недостижимым объемом перехваченных пар открытого и зашифрованного текстов, либо взлом алгоритма требует настолько значительных затрат времени, что к его моменту интересующая противника информация утратит свою актуальность.

Алгоритмы AES, Blowfish, CAST5 и 3DES считаются криптостойкими. На сегодняшний день неизвестно случаев успешных атак на эти алгоритмы.

Исследуем временные характеристики криптопреобразований следующих симметричных алгоритмов шифрования информации: AES, Blowfish, CAST5, 3DES. Алгоритмы для исследования были выбраны на основе следующих критериев:

- а) открытость структуры и принципа работы алгоритма;
- б) отсутствие успешных (на сегодняшний день) попыток взлома алгоритма;
- в) возможность бесплатного использования алгоритма;
- г) наличие программных реализаций в свободных криптографических библиотеках.

При выборе алгоритмов шифрования для исследования нельзя обойти стороной установленный в Российской Федерации единый стандарт криптографического преобразования текста для информационных систем, который носит обязательный характер для государственных органов, организаций, предприятий, банковских и иных учреждений, чья деятельность связана с обеспечением информационной безопасности государства [3]. В частности, стандарт ГОСТ Р 34.12-2015 «Информационная технология. Криптографическая

защита информации. Блочные шифры» описывает блочные алгоритмы шифрования «Магма» (он же ГОСТ 28147-89), и «Кузнечик».

Алгоритм Магма имеет богатую историю, он подвергался анализу со стороны большого количества отечественных и зарубежных специалистов в области криптографии. В 2010 году алгоритм был выдвинут на включение в международный стандарт блочного шифрования ISO/IEC 18033-3:2010. Однако, в процессе исследований было выявлено, что алгоритм не обеспечивает уровня стойкости, требуемого организацией ISO. Известный криптоаналитик, основоположник алгебраического криптоанализа Николя Куртуа в своей статье «Security Evaluation of GOST 28147-89 In View Of International Standardisation» (URL: <http://eprint.iacr.org/2011/211.pdf>) описывает проведенную атаку на GOST, которая оказалась эффективнее полного перебора в 28 раз. Атака на Магму была представлена на конференции FSE 2011 и более подробно описана в работе Н. Куртуа. Это первый случай в истории, когда серьезный стандартизированный блочный шифр, созданный для защиты секретов военного уровня и предназначенный для защиты документов государственной тайны для правительств, крупных банков и других организаций, оказался взломан математической атакой[4].

Описанный случай наглядно демонстрирует негласный принцип, формировавшийся на протяжении всей истории криптографии: чем старше алгоритм, тем больше шансов на то, что он будет взломан в ближайшем будущем (если уже не взломан).

Что касается алгоритма «Кузнечик», то он является сравнительно новым и малоизученным. Он не имеет очевидных недостатков, но всё же Кузнечик ещё мало распространен и на сегодняшний день не имеет программных реализаций в популярных криптографических библиотеках.

Алгоритмы «Магма» и «Кузнечик» заметно проигрывают в быстродействии AES, Blowfish и CAST, что подтверждается стандартным бенчмарком при помощи программного продукта VeraCrypt.

Исходя из всего вышеизложенного было принято решение отечественные стандарты шифрования в исследование не включать.

Параметры исследуемых алгоритмов шифрования представлены в таблице 1. С целью максимально объективных результатов исследования за эталонное значение была принята длина ключа, равная 128 битам. Длина ключа для алгоритма 3DES является фиксированной, что обусловлено структурой самого алгоритма.

Таблица 1. Параметры исследуемых алгоритмов шифрования

Алгоритм	Структура	Размер блока, бит	Длина ключа, бит	Длина ключа, используемая в эксперименте, бит
AES	Подстановочно-перестановочная сеть	128	128/192/256	128
Blowfish	Сеть Фейстеля	64	128-448	128
CAST5	Сеть Фейстеля	64	40-128	128
3DES	Сеть Фейстеля	64	168	168

Алгоритмы были реализованы с использованием двух подходов:

1. Разработка программного обеспечения на основе опубликованных в свободном доступе криптографических алгоритмов;
2. Разработка программного обеспечения с использованием свободной криптографической библиотеки Bounce Castle (язык программирования Java).

В криптографических библиотеках, как правило, проводится тщательный аудит программного кода, структурное и функциональное тестирование всех компонентов,

оптимизация программных процедур. Поэтому второй подход является более оптимальным, не только с точки зрения минимизации времени разработки, но и с точки зрения эффективности реализации.

Тестирование криптографических алгоритмов выполнялось на программных платформах Windows 10 Pro и Linux Mint Rosa. Эксперименты проводились на компьютере с процессором Intel Core i3 M330 @ 2.13GHz x 4 и объемом оперативной памяти 3 Гб.

Данные для зашифрования представлены сгенерированными случайным образом байтовыми массивами размерностями: 10 Мб, 50 Мб, 100 Мб, 200 Мб и 500 Мб. Результаты тестирования скоростей шифрования/расшифрования разработанных систем шифрования оцениваются Мбайт/с и представлены в табл. 2, 3.

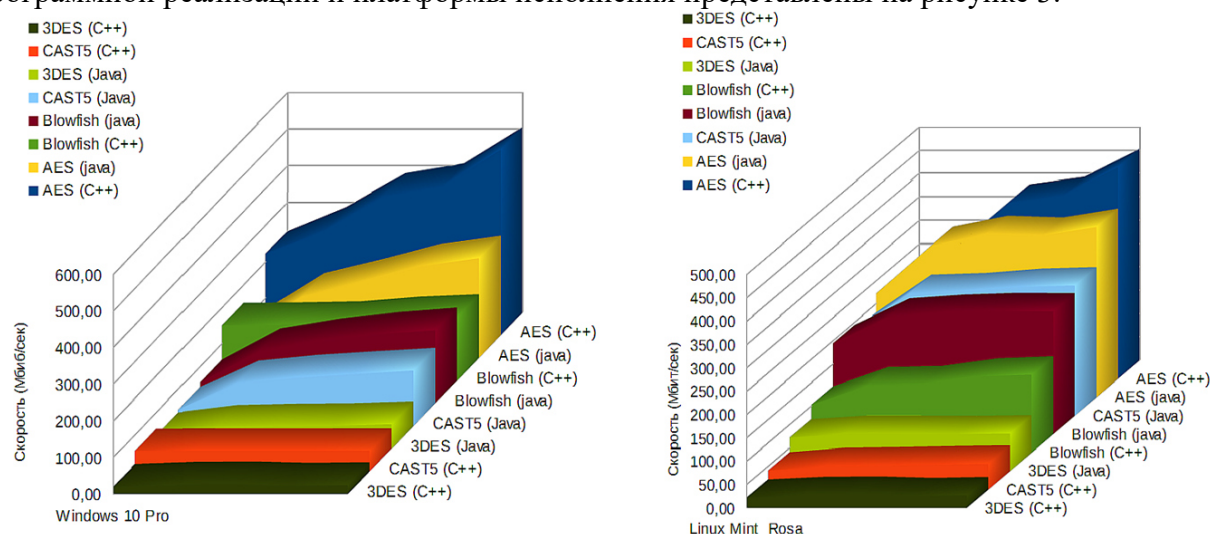
Таблица 2. Быстродействие алгоритмов шифрования на Java

Алгоритм	Windows 10						Linux Mint					
	Размер файла, Мб											
	10	50	100	200	500	средняя скорость	10	50	100	200	500	средняя скорость
	Быстродействие, Мбит/с											
AES	70,8	169,49	207,79	250,39	272,29	194,15	222,2	325,20	349,34	345,57	364,63	321,39
Blowfish	57,97	142,35	168,78	188,24	199,80	151,43	195,12	251,57	256,94	263,59	262,64	246,53
CAST5	44,69	117,30	132,45	142,48	152,21	117,83	181,82	263,16	266,67	275,86	279,13	253,33
3DES	37,74	56,74	60,61	62,11	66,90	56,82	72,07	79,68	79,60	79,44	80,96	78,35

Таблица 3. Быстродействие алгоритмов шифрования на C++

Алгоритм	Windows 10						Linux Mint					
	Размер файла, Мб											
	10	50	100	200	500	средняя скорость	10	50	100	200	500	средняя скорость
	Быстродействие, Мбит/с											
	AES	222,22	287,77	379,15	406,09	506,33	360,31	222,22	325,20	349,34	345,57	364,63
Blowfish	150,94	152,09	155,04	167,89	173,91	159,98	195,12	251,57	259,74	263,59	262,64	144,67
CAST5	56,34	56,02	55,98	56,00	56,34	56,13	181,82	263,16	266,67	275,86	272,13	50,72
3DES	20,51	26,49	27,78	23,19	25,43	24,68	72,07	79,68	79,60	79,44	80,96	26,03

Для удобного и наглядного анализа полученных результатов, содержимое таблиц 1,2 представлено визуально на рисунке 2. Усредненные скорости шифрования со указанием программной реализации и платформы исполнения представлены на рисунке 3.



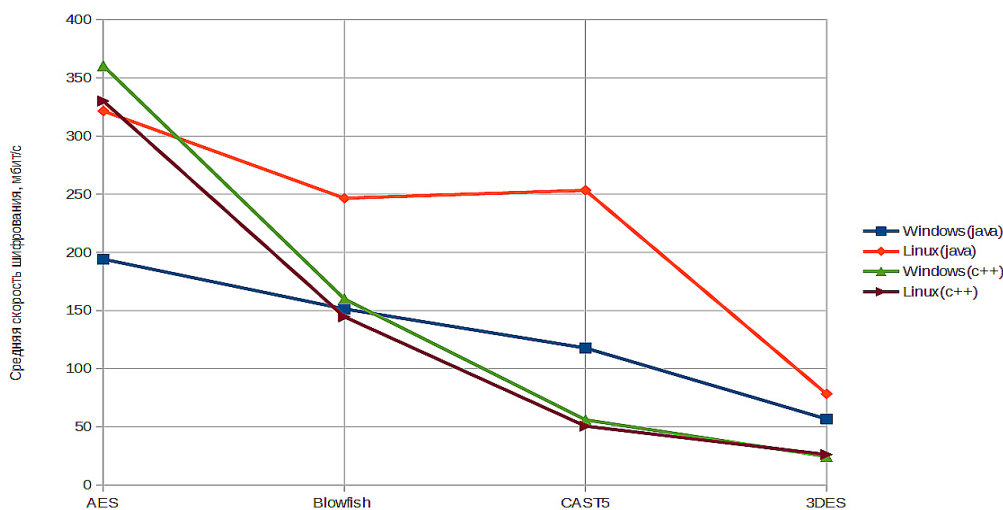


Рис.3. Средние скорости шифрования алгоритмов

Из полученных результатов, можно сделать вывод, что наибольшим быстродействием обладает реализация алгоритма AES на языке C++. Аутсайдером по скорости шифрования является алгоритм 3DES. Быстродействие алгоритмов, реализованных на языке программирования Java существенно зависит от выбора платформы — ОС Linux Mint Rosa здесь явно предпочтительна, чем Windows 10. Также стоит отметить, что с увеличением объема данных, скорость шифрования алгоритмов также повышается.

При выборе симметричного алгоритма блочного шифрования следует принимать во внимание, что в условиях современных информационных технологий рекомендуется применять алгоритмы с размером блока не менее 128 бит. Из рассмотренных алгоритмов только AES удовлетворяет данной рекомендации. Это не значит, что алгоритмы с размером блока 64 бит «плохие», однако из-за возможных коллизий этими алгоритмами небезопасно зашифровывать на одном ключе файлы размером более 32 Гб.

Также следует обратить внимание на то, являются ли алгоритмы стандартизованными. Алгоритмы AES, CAST5 и 3DES включены в международный стандарт ISO/IEC 18033-3:2010 «Информационные технологии. Методы и средства обеспечения безопасности. Алгоритмы шифрования. Часть 3. Блочные шифры» (обновлен в 2016 году).

Все рассмотренные алгоритмы обладают высокой криптостойкостью и могут быть эффективно реализованы программным способом.

К достоинствам программной реализации алгоритмов можно отнести их гибкость и переносимость, что позволяет модифицировать разработанный алгоритм под любой тип операционной системы. Кроме того, обновление программного обеспечения требует меньшего вложения финансовых и временных затрат. К недостаткам программных средств криптографической защиты следует отнести возможность несанкционированного вмешательства в работу алгоритмов и получения секретной информации, хранящейся в общедоступной памяти[2].

Полученные результаты представляют большой интерес и практическую ценность в перспективе применения их в будущем.

Библиографический список

1. Панасенко С.П. Алгоритмы шифрования. Специальный справочник. – Спб.: БХВ-Петербург, 2009. – 276 с.: ил.
2. Петров А. А. Компьютерная безопасность. Криптографические методы защиты. / А. А. Петров – М.: ДМК, 2000. – 448 с
3. Баричев, С.Г. Основы современной криптографии / С.Г. Баричев, Р.Е. Серов. – М.: Горячая линия – Телеком, 2006. – 152 с.
4. Блочный шифр ГОСТ 28147-89 взломан в разгар усилий по его международной стандартизации: [Электронный ресурс] // Журнал «Information Security», URL: <http://www.itsec.ru/articles2/crypto/gost-28147-89-vzloman>. (Дата обращения: 27.03.2017)

УДК 004.92

А.С. Рязанов, магистрант 2-го курса**Научный руководитель: Е.Н. Машенко, к.т.н., доцент****Севастопольский государственный университет****ИССЛЕДОВАНИЕ ХАРАКТЕРИСТИК КАЧЕСТВА СИСТЕМЫ ОБНАРУЖЕНИЯ
ВТОРЖЕНИЙ В ОБЛАЧНОЙ ИНФРАСТРУКТУРЕ**

Аннотация: Предложена имитационная модель системы обнаружения вторжений как компонента системы информационной безопасности облачной инфраструктуры. Проведены планирование и реализация экспериментов с целью повышения эффективности функционирования системы обнаружения вторжений.

Ключевые слова: облачная инфраструктура, информационная безопасность, система обнаружения вторжений, имитационный эксперимент.

Annotation: A simulation model of the intrusion detection system is proposed as a component of the information security system of the cloud infrastructure. Planning and implementation of experiments to improve the efficiency of the intrusion detection system.

Keywords: cloud infrastructure, information security, intrusion detection system, simulation experiment.

Введение. Быстро развивающиеся компьютерные информационные технологии вносят значительные изменения в нашу жизнь. В настоящее время понятие “информация” используется как обозначение специального товара, который можно приобрести, продать, обменять на что-то иное и т.п. При этом стоимость информации довольно часто может превышать стоимость компьютерной системы, в которой она находится. Поэтому возникает необходимость в защите информации от несанкционированного доступа, уничтожения, кражи, умышленного изменения и других неправомерных действий. Для защиты информации от вредоносного воздействия необходимо развернуть систему информационной безопасности, которая предотвратит возможность проникновения в информационную систему. Особенно важны вопросы обеспечения безопасности при применении современных облачных технологий и развертывании облачных инфраструктур.

Система обнаружения вторжений (IDS – Intrusion Detection Systems) - один из основных элементов систем информационной безопасности при реализации концепции многоуровневой защиты облачной инфраструктуры [1-4]. Системой обнаружения вторжений является множество различных аппаратных и программных средств, которые объединены одним общим свойством - они проводят анализ использования компьютерных систем. В случае обнаружения каких-либо подозрительных или просто нетипичных событий, которые могут нарушить безопасность компьютерной системы, способны предпринимать определенные самостоятельные действия по обнаружению, идентификации и устранению их причин, которые впоследствии могут являться сетевыми атаками, если они не были замечены системой. К подозрительной и нетипичной активности относятся: сетевые атаки против уязвимых систем, атаки, которые направлены на повышение привилегий, неавторизованный доступ к важным файлам, а также действия вредоносного программного обеспечения, к которым относятся компьютерные вирусы.

Системы обнаружения вторжений также являются системами, которые собирают информацию из различных точек защищаемой компьютерной системы (вычислительной сети) для дальнейшего анализа этой информации как для выявления попыток вторжений, так и для реальных нарушений защиты (вторжений в компьютерную систему). Поведение после обнаружения указывает на реакцию IDS на атаки. Реакция может быть активной – IDS предпринимает корректирующие действия (устраняет лазейки) или действительно активной (закрывает доступ для возможных нарушителей, делая недоступными сервисы). Если IDS только выдаёт предупреждения о попытках вторжения, либо об уже осуществленных сетевых атаках, её называют пассивной. По методам анализа IDS делят на две группы: IDS, которые сравнивают информацию с предустановленной базой сигнатур атак и IDS, контролирующие частоту событий или обнаружение статистических аномалий.

Первым методом, который был применен для обнаружения вторжений, является анализ сигнатур. Анализ сигнатур базируется на простом понятии совпадения определенной

последовательности с образцом. Во входящем трафике пакеты просматриваются байт за байтом и сравниваются с сигнатурой (подписью) – характерной строкой программы, которая указывает на характеристику или свойство вредоносного трафика. Такая подпись может содержать правило, команду или ключевую фразу которая связана с сетевой атакой. В случае обнаружения совпадения объявляется тревога. Основным достоинством метода анализа сигнатур является высокая скорость обработки данных. Также этот метод точно и быстро регистрирует подозрительную деятельность и принимает необходимые меры. К недостаткам этого метода можно отнести то, что со временем скорость работы IDS будет замедляться в связи с тем, что база данных сигнатур будет увеличиваться из-за появления новых видов атак. Фактически, каждая новая атака увеличивает базу данных проверяемых сигнатур. Также к недостаткам данного метода можно отнести то, что система может выявить только уже известные атаки, сигнатуры которых имеются. Однако этот недостаток компенсируется тем, что большинство атак происходит по давно известным алгоритмам.

Второй метод анализа подозрительной активности состоит в рассмотрении строго форматированных данных трафика сети - протоколов. У этого метода также имеются свои достоинства и недостатки. Из-за предпроцессов, требующих тщательной экспертизы протоколов, анализ протокола может быть довольно медленным. Кроме того, правила проверки для системы протокола трудно написать и понять. На первый взгляд, IDS, которые используют метод на основе анализа протокола работают медленнее, чем IDS на основе сигнатуры, однако, они более надежны в смысле масштабности и результатов.

Постановка задачи.

Объектом исследования являются системы обнаружения вторжений как компонент многоуровневой архитектуры системы информационной безопасности облачной инфраструктуры.

Предметом исследования является процесс функционирования IDS при отражении различных типов сетевых атак.

Целью исследования является повышение эффективности функционирования IDS и обеспечение достаточного уровня безопасности облачных сервисов.

Для достижения цели исследования необходимо решить следующие задачи:

- 1) построение имитационной модели процесса функционирования IDS на основе концепции многоподходного моделирования;
- 2) планирование и проведение экспериментов с моделью;
- 3) анализ результатов экспериментов и выработка рекомендаций для систем поддержки принятия решений по управлению IDS.

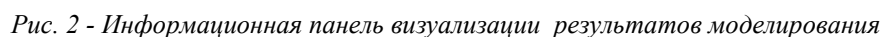
Описание модели системы обнаружения вторжений.

Для построения модели IDS как компонента системы информационной безопасности использовались современные технологии имитационного моделирования: агентное моделирование, дискретно-событийное моделирование, средства визуализации.

Модель разработана с использованием среды AnyLogic [5]. При построении модели использовалась концепция сигнатурной IDS. Интерфейсы модели приведен на рисунках 1, 3; информационные панели визуализации результатов моделирования показана на рисунках 2,4.

Для организации и проведения экспериментов в ходе исследований были определены целевые функции (отклики) системы обнаружения вторжений, описаны входные переменные (факторы) и уровни факторов. Откликами системы в данной группе экспериментов являются количество заблокированных атак и количество пропущенных атак.

В качестве факторов выбраны типы сетевых атак, в качестве уровней факторов - количество атак за моделируемый период работы IDS. Выделенные факторы и уровни факторов приведены в таблице 1.



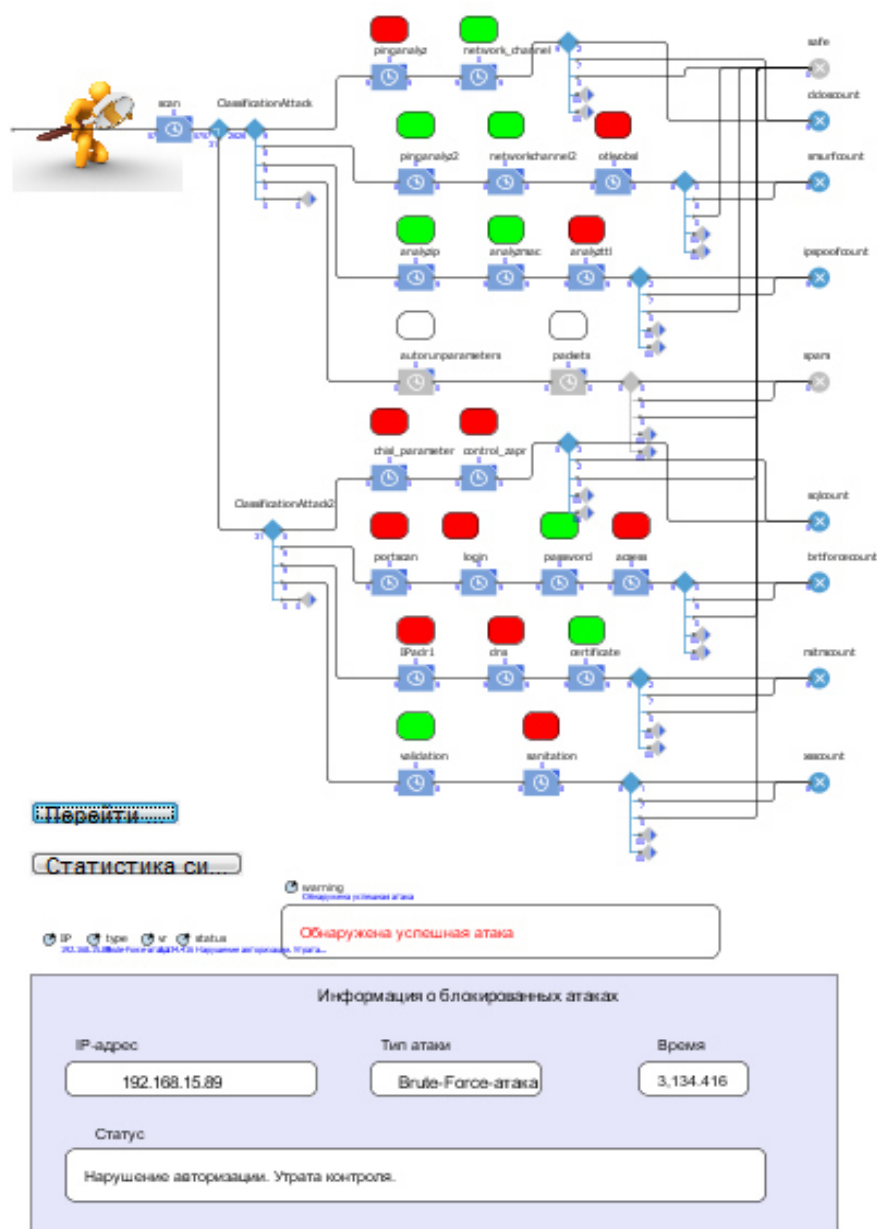


Рис. 3 – Интерфейс анализатора состояния системы

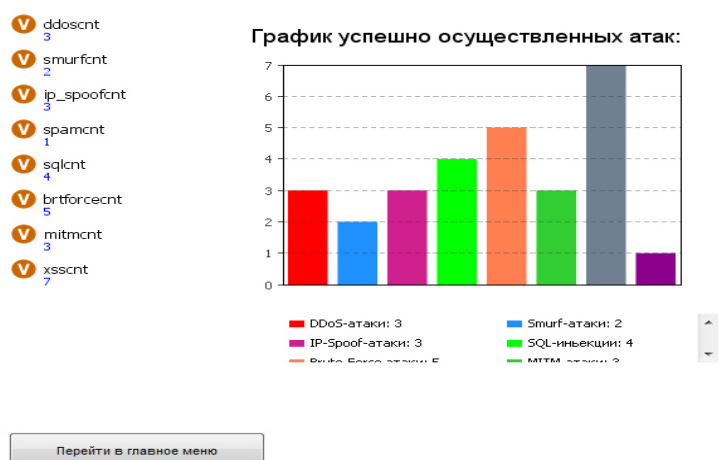


Рис. 4 - Информационная панель визуализации результатов моделирования для анализатора состояния системы

Таблица 1 - Факторы эксперимента и их уровни

Факторы		Уровни	
		Max	Min
A	DDoS атака	20	10
B	Smurf атака	20	10
C	Ip-Spoof атака	30	10
D	Spam атака	50	10
E	SQL-инъекция	50	10
F	Brute-Force атака	40	10
G	MITM атака	30	10
H	XSS атака	50	10

Было проведено планирование экспериментов с моделью IDS согласно частичному факторному плану [6], план экспериментов приведен в таблице 2. В данной таблице значение 1 показывает присутствие фактора (определенного типа атаки) в эксперименте, а значение 0 – отсутствие.

Таблица 2 - Планирование факторного эксперимента.

№п/п	Факторы								
1. Влияние одиночных факторов									
1.1	A	1	0	0	0	0	0	0	0
1.2	B	0	1	0	0	0	0	0	0
1.3	C	0	0	1	0	0	0	0	0
1.4	D	0	0	0	1	0	0	0	0
1.5	E	0	0	0	0	1	0	0	0
1.6	F	0	0	0	0	0	1	0	0
1.7	G	0	0	0	0	0	0	1	0
1.8	H	0	0	0	0	0	0	0	1

Фрагмент результатов экспериментов для группы экспериментов 1 (Влияние одиночных факторов), когда система подвергается не более чем одной атаке каждого типа одновременно, приведен в таблице 3.

Таблица 3 – Результаты экспериментов.

№п/п	Обозначение	Тип атаки	Количество атак	Блокировано	Пропущено
1.1	A	DDoS	15	13	2
1.2	B	Smurf	15	14	1
1.3	C	IP-Spoof	20	17	3
1.4	D	Spam	40	40	0
1.5	E	SQL инъекция	40	38	2
1.6	F	Brute-Force	30	25	5
1.7	G	MITM	20	16	4
1.8	H	XSS	40	33	7

Выводы. В результате проведения экспериментов с имитационной моделью IDS в среде Anylogic была получена информация, которую в дальнейшем можно будет использовать для оптимизации системы обнаружения вторжений с целью повышения эффективности обнаружения атак при минимальных затратах ресурсов и времени. В частности, необходимо ввести дополнительные параметры для сканирования XSS – атак, MITM атак, Brute-Force атак в связи с тем, что на основе проведенных экспериментов была выявлена уязвимость IDS системы перед этими атаками.

Разработанная имитационная модель может быть использована как часть системы поддержки принятия решений по управлению качеством работы IDS, с целью снижения трудоемкости и сокращения времени обнаружения атак и, следовательно, повышения эффективности системы информационной безопасности облачной инфраструктуры в целом.

Библиографический список

1. The 2016 Global Cloud Data Security Study. Ponemon Insitute LLC, 2016. – 40 с.
2. Беккер, М.Я. Информационная безопасность при облачных вычислениях: проблемы и перспективы / М.Я. Беккер, Ю.А. Гатчин, Н.С. Кармановский, А.О. Терентьев, Д.Ю. Федоров // Научно-технический вестник Санкт-Петербургского государственного университета информационных технологий, механики и оптики. – 2011. – №1(71). – С. 97-102.
3. CSA Security Guidance for Critical Areas of Focus in Cloud Computing v.4. [Электронный ресурс]. – Режим доступа: <https://cloudsecurityalliance.org/guidance/> Дата доступа - 10.03.2017.
4. NIST Guide to Intrusion Detection and Prevention Systems (IDPS). [Электронный ресурс]. URL: http://csrc.nist.gov/publications/drafts/800-94-rev1/draft_sp800-94-rev1.pdf Дата доступа - 10.03.2017.
5. AnyLogic - инструмент многоподходного имитационного моделирования [электронный ресурс]. Режим доступа: <http://www.anylogic.ru/overview>. Дата доступа: 10.03.17.
6. Налимов В.В. Теория эксперимента. М.: Наука, 1971. — 208 с.

СЕКЦИЯ 9

Системное и прикладное программное обеспечение



УДК 004.9

И.А. Безпалый, студент 3-го курса**Научный руководитель: Т.А.Абрамов, ст. преп.***Севастопольский государственный университет***КОНЦЕПТУАЛЬНАЯ МОДЕЛЬ ИГРЫ «ОСКАЛОР»***Аннотация: статья предлагает компьютерную игру на основе предложенной альтернативной вселенной, придуманной автором**Ключевые слова: альтернативная вселенная, игра, концепт, поколение, стратегия, головоломка**Annotation: Article offers a computer game based on the proposed alternative universe, invented by the author**Key words: alternative universe, game, concept, generation, strategy, puzzle***Предисловие**

Оскалор – это альтернативная фэнтези вселенная в трехмерном пространстве, главными особенностями которой является её цикличность и бесконечное количество вариаций объектов в замкнутом пространстве.

Оскалор (игра) – это игра в основу которой легли фундаментальные правила одноименной вселенной, действие которой были переведены на 2D плоскость. Жанр игры - головоломка. В игре перед игроком стоит сложная задача - следить за вселенной и подводить её к концу её цикла. Игрок всеми силами должен стараться не дать ни одной части вселенной погаснуть и избавляться от всех преград, вставших на его пути.

Постановка задачи.

Необходимо описать: 1) основные принципы и правила вселенной, выделить из этих принципов правила для игры; 2) проблемы, с которыми может встретиться игрок во время игры; 3) примерный интерфейс игры; 4) модель игры; 5) варианты прохождения. Предположить дальнейшие перспективы развития игры и возможное применение в других областях науки.

Краткое описание вселенной.

Будут описаны правила вселенной без привязки к реализации на компьютере.

Вселенная представляет собой замкнутое пространство в виде шара. Вначале всё пространство шара представляет собой один объект под названием Единство. Внутри Единства находится Ядро. Ядро – это существо, содержащее в себе гигантское количество энергии. Энергию, которое оно содержит, можно разделить на два вида: магическую и жизненную энергию. Жизненная энергия напоминает по своей природе крепкий камень, магия же похожа на воздух или легкую вату. Также магическая энергия содержит в себе разные типы магии (эти типы пока не будем описывать). Ядро – единственное существо способное конвертировать жизненную энергию в магическую.

Ядро всегда создает вокруг себя сферу под названием Мир. Создавая его, Ядро затрачивает энергию. В Мире Ядро создает новый ландшафт, атмосферу, флору и фауну. Все созданное состоит из смеси жизненной и магической энергии в разных пропорциях, но при этом ни один созданный объект не может содержать только один вид энергии, «везде где есть жизнь, есть толика магии и везде где есть магия, всегда есть крупинка жизни». Количество той или иной энергии влияет на все параметры и характеристики создаваемого объекта, особенно у флоры и фауны. Если взять ландшафт, то в случае с большим количеством жизненной энергии, это будут горы, холмы и т.п., в случае с большим количеством магической энергии это будут моря, реки, озера и т.п. Такой же принцип пропорций работает и с флорой и фауной (только в случае с ними вступают также и типы магии). Важный момент, каждое живое существо (подразумевается флора и фауна) имеет неразрывную связь с Ядром, по которому оно вытягивает из существ их жизненную энергию (что является причиной старения существа). Любое новое живое существо в Мире, в момент зачатия всегда обретает связь с Ядром и создает в себе жизненную и магическую энергию, повторяя пропорции родителей. То есть каждое новое поколение флоры и фауны – это новая порция жизненной энергии для Ядра.

После создания Мира и всех его составляющих, Ядро засыпает и единственной его деятельностью является вытягивание жизни из флоры и фауны. Как только Ядро заполнит себя энергией, оно сворачивает Мир – переводит весь мир обратно в энергию, которую использовала для его создания, после чего переполняется и взрывается.

После взрыва Единство разбивается на Осколки (отдельные участки/сегменты), важно понять, что после взрыва ничто никуда не разлетается, все остается на месте. Это напоминает хрустальный треснувший шар (*рис.1*). Только в случае со вселенной каждая трещинка, проходящая в пространстве шара, образует сегмент – Осколок. В 2D это будет выглядеть как на *рисунке 2*



Рис.1. Хрустальные шарики;

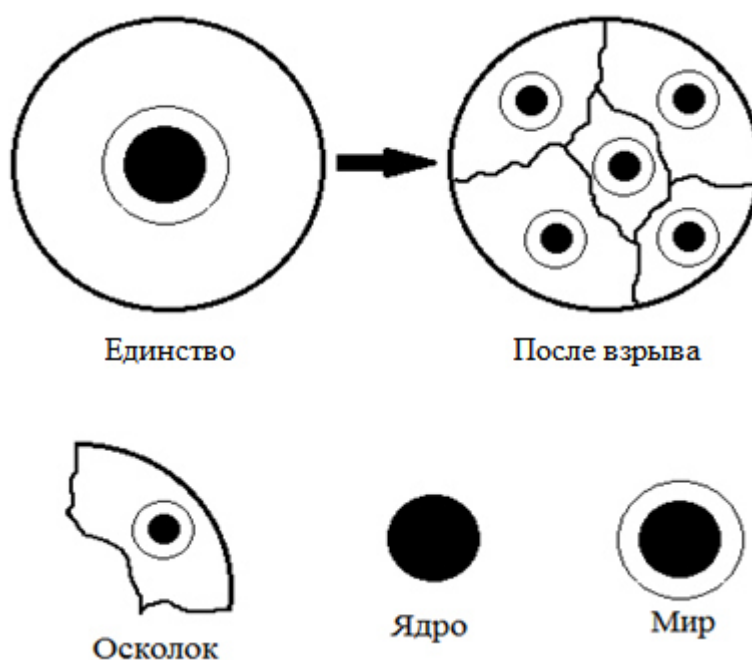


Рис.2. Простейшее представление вселенной после взрыва единства

В Осколках зарождается Ядра, которые обладают частичной памятью, фрагментами того что создавал их взорвавшийся предок. При создании Мира, пробелы заполняются новыми деталями, таким образом во вселенной создается разнообразие форм ландшафта, атмосферы, флоры и фауны, одновременно разных и в чем-то похожих.

После создания Мира, Ядра будут продолжать накапливать энергию для взрыва, таким образом разбивая свой Осколок (сегмент) на новые сегменты, в которых будут снова

зарождаются Ядра, которые будут повторять действия предка. Этот процесс деления будет продолжаться до тех пор, пока Осколки (сегменты) станут столь малы что их размеры не будут позволять зарождать миры. После чего все Осколки (сегменты) и ядра сливаются воедино образуя вновь единый объект – Единство с которого все и началось. В этом заключается жизненный цикл вселенной (Рисунок 3).

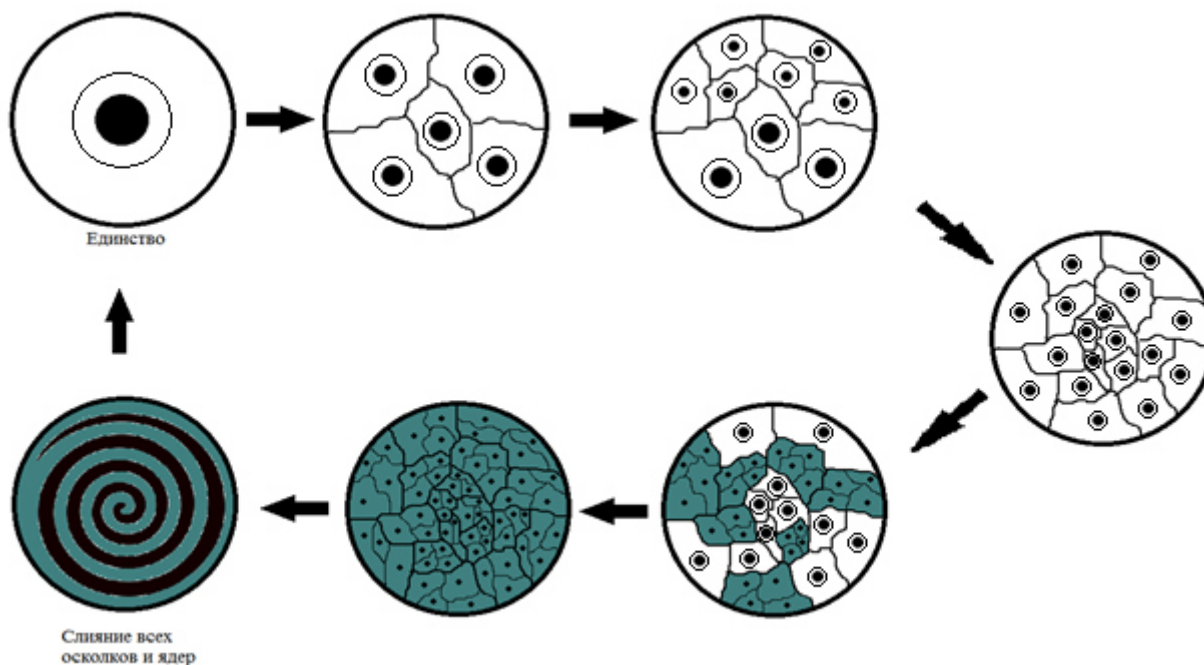


Рис.3. Простейшее представление цикла вселенной
(осколки серого цвета – это осколки в которых (условно) невозможно создать мир)

Однако на пути цикла могут появиться проблемы, такие как потухшие Ядра. Потухшие Ядра – это те ядра, количество энергии в которых достигло критического минимума. Они вынуждены свернуть мир для поддержания существования и уйти в спячку, ожидая момента, когда в них появится достаточно энергии для создания мира, который их будет питать. Проблема заключена в том, что потухание ядра происходит из-за самого мира, который оно создает. Флора и фауна, которая должна питать мир начинает, наоборот, вытягивать энергию из ядра (причину опустим), что и приводит Ядро к критическому состоянию (Рисунок 4).

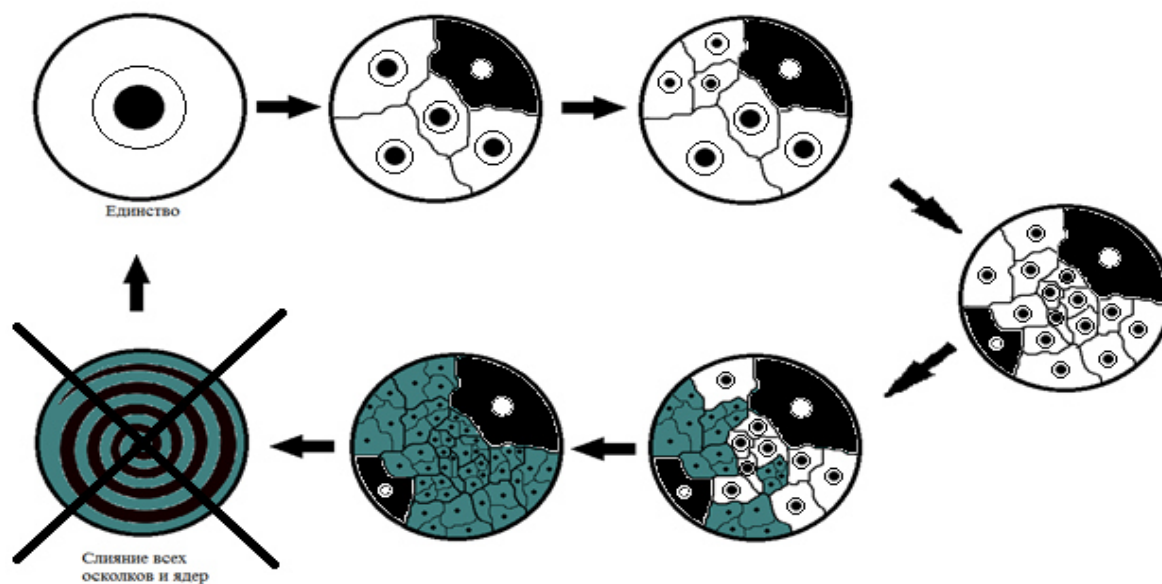


Рис.4. Невозможность завершить цикл из-за потухших осколков

Описание игры:

Прежде всего, выделим то, что игра будет 2D и относится к жанру головоломки. Платформой для игры выбраны операционные системы Win/Mac/Linux.

Правила игры:

На основе описания вселенной можно выделить такие правила:

1. Место действия игры – это замкнутая плоскость в виде круга разделенная на сегменты (участки) неоднородной формы.
2. Каждый сегмент содержит в себе два объекта: ядро, мир.
 - Ядро обладает емкостью (C) в которой хранится энергия, $C \in [20;100]$. Размер зависит от поколения, которому относится сегмент;
 - Мир способен поглощать и генерировать энергию (e) каждую секунду, количество $e \in [-10;10]$. Делится на три типа который определяется случайным образом:
 - *Полезный*;
Производить (+) энергию или вовсе ничего не делать (0);
 - *Вредный*;
Поглощает энергию (-);
 - *Паразит*;
Маскируется под полезный мир. Только производит (+) энергию, но не дает ядру заполниться и заражает ядра сливая в них энергию из своей емкости.
 - Слив происходит если в следующую секунду должно произойти заполнение ядра мира паразита. Слив производится в ядро соседнего сегмента, выбранное случайным образом. Сегмент, в который была слита энергия, помечается как зараженный;
3. Игрок может:
 - Хранить энергию. Игрок обладает собственной емкостью (хранилищем), размер которой не является постоянным числом, в отличие от ядер. Размер емкости игрока (C_i) зависит от количества единиц энергии (e), затрачиваемых каждую секунду для её поддержания и рассчитывается по таким правилам:
 - если количество e равно 0, то C_i будет равен 10;
 - если количество e равно 1, то C_i будет равен 50;
 - если количество e равно 2, то C_i будет равен 100;
 - если количество e равняется $[3,10]$, то $C_i = 100 + 10 * e$;
 - Брать или давать энергию ядрам из своей емкости. Эти взаимодействия осуществляются в двух режимах:
 - *единичный*, взаимодействие совершается один раз, совершение взаимодействия в этом режиме обходится в 5 e . Используется для оперирования большим количеством энергии;
 - *постоянный*, действие повторяется каждую секунду до тех пор, пока емкость ядра не упадет до 0 или не будет заполнено. Чтобы взаимодействие было повторяющимся нужно заплатить в 2 раза больше оперируемой энергией, т.е. если игрок хочет брать или давать каждую секунду 3 единицы энергии ядру, ему нужно один раз потратить 6 единиц энергии для этого;
 - Проводить исследование мира для выявления является ли он паразитом и ядра, заражено оно или нет. Стоимость исследования 10 e ;

4. В начале игры всю плоскость покрывает один сегмент (Единство), нулевое поколение. Каждое следующее поколение рассчитывается по такому правилу:
 - Как только заполняется емкость ядра, оно «взрывается», после чего, сегмент разбивается на части (меньшие сегменты). Число частей, на которые разбивается сегмент случайно, в пределах от 3 до 5. Сегмент может иметь от 2 до 16 соседей.
5. Если емкость ядра упадет до 0, сегмент станет выключенным (потухшим), а мир уснет.
 - Чтобы включить (зажечь) сегмент, нужно влить энергию в ядро, в размерах половины размера емкости этого ядра. После этого сегмент включится, мир проснется, а емкость ядра будет заполнена лишь на четверть.
 - Если сегмент был заражен, то количество потухших потомков равняется большей части потомков (3 потомка – 2 потухших, 4 потомка – 2 потухших, 5 потомков – 3 потухших). Потомки взрыва не помечены как зараженные;
6. Если сегмент относится к третьему поколению, то игрок или паразит не способен взаимодействовать с объектами сегмента, а сегмент всегда будет включен.
7. Если все сегменты являются третьим поколением, то все сегменты сливаются воедино, образуя один единый сегмент (Единство) покрывающий всю плоскость
8. Игрок проигрывает если:
 - На поле выключились все сегменты;
 - Энергия в хранилище игрока упала до 0;
9. Игрок выигрывает если:
 - Игрок дошел до Единства;

Проблемы, с которыми может встретиться игрок во время игры:

Цели игрока:

1. довести все сегменты к третьему поколению,
2. не растерять всю энергию в хранилище.

Чтобы эти цели достигнуть, игрок должен не давать выключаться или включать сегменты, и поддерживать количество энергии в хранилище больше 0.

Миры вредного типа и паразиты мешают достижению первой цели игроку. Те и другие создают выключенные сегменты (хотя в одном случае паразит может помочь игроку, тогда, когда сливает энергию в ядро сегмента второго поколения, который породит сегменты третьего поколения).

В случае с вредным миром стоит просто постоянно вливать число энергии большее, чем число, которое мир вытягивает из ядра. В случае с паразитом, единственный способ избавиться от паразита – это единичным образом заполнить емкость ядра сегмента с паразитом, чтобы тот дал новое поколение.

Чтобы поддерживать количество энергии в емкости больше нуля, игрок должен правильно распределить приход энергии от ядер и его расходы на избавление от вредных миров и миров паразитов. В случае недосмотра, когда приход энергии каждую секунду будет меньше расходов, то количество энергии в хранилище упадет до нуля.

Интерфейс игры. Главное меню (Рисунок 5): 1 – Фон; 2 – Новая игра; 3 – Загрузить сохраненную игру; 4 – Выход;

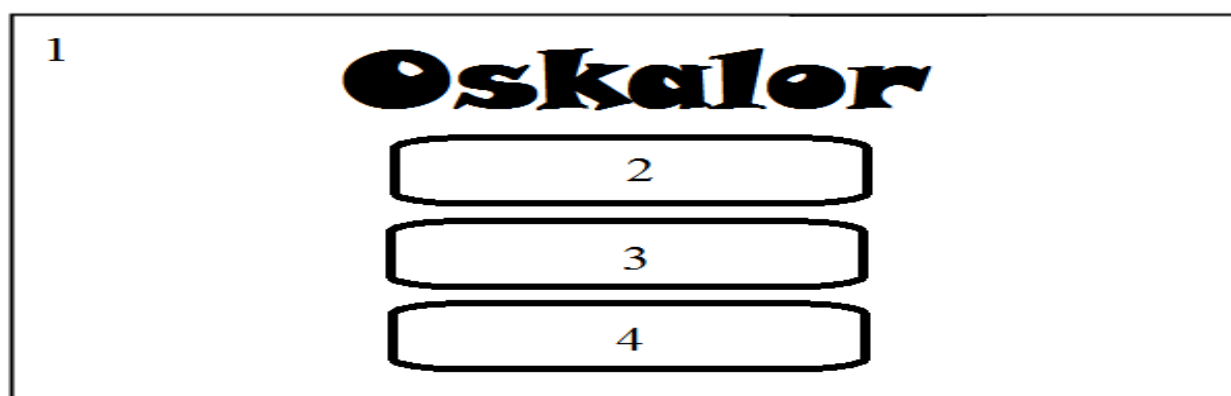


Рис.5. Примерное расположение форм в Главном меню

Игровой процесс (Рисунок 6):

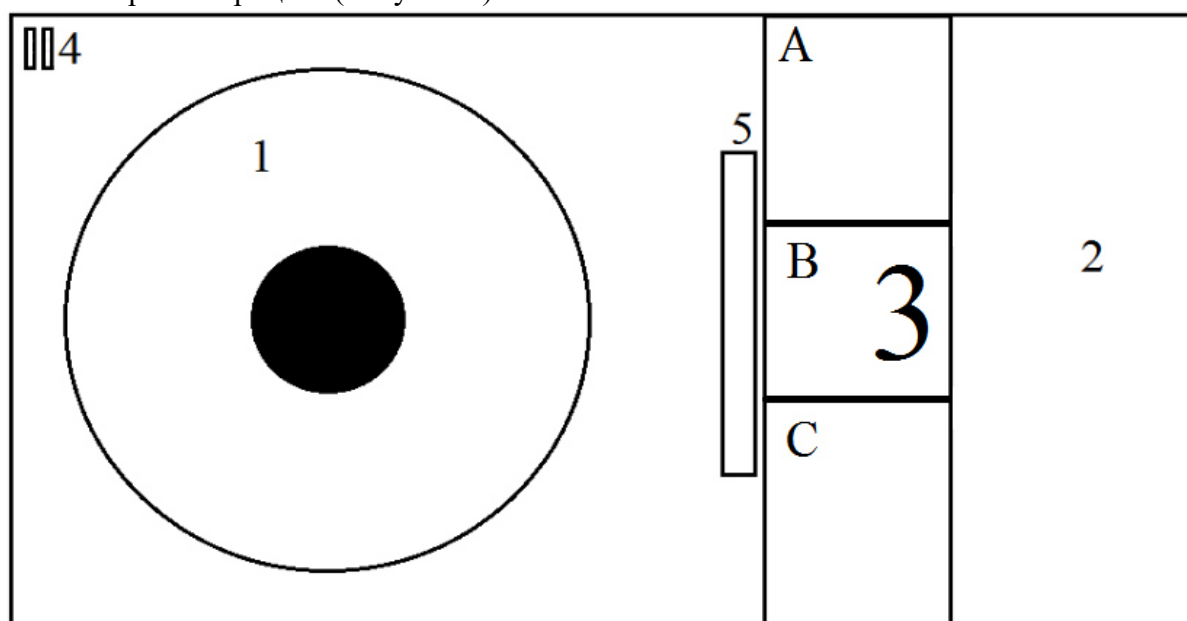


Рис.6. Примерное расположение форм в интерфейсе игры

1 – Карта сегментов;

Здесь игрок выбирает сегмент для просмотра. На карте визуальное показано состояние сегмента и тип мира. Если сегмент темный, то он выключен, а если светится, то включен. Цвет свечения показывает какой в сегменте мир: светится голубым – полезный, красным – он вредный, зеленым – паразит, белым – закрытый сегмент третьего поколения;

2 – Общая статистика;

Состоит из:

1. Список просмотренных сегментов и краткая по ним сводка (вкл., выкл.);
2. Общий прирост и затраты энергии в емкости игрока за секунду времени;
3. Общее количество выделяемой энергии во всей вселенной (+, -, 0);
4. Процент завершенности для следующего слияния в Единство;
5. Размер хранилища игрока и количество энергии в ней содержащееся;

3 – Сводка по рассматриваемому осколку

Выплывающее окно

Разделено на три поля:

А. Сегмент;

Состояние Сегмента: включен, выключен;

Поколение: номер, поколения открыт, закрыт;

В. Ядро;

Емкость: кол-во энергии/размер емкость

С. Мир;

Тип мира: полезный, вредный, паразит

Количество выделяемой энергии: некоторое число

4 – Пауза;

Остановка времени. Предназначена для того чтобы Игрок мог назначить свои действия на следующую единицу времени.

5 – Полоса емкости игрока;

Визуальное представление количества энергии в емкости Игрока

Меню паузы (Рисунок 7):

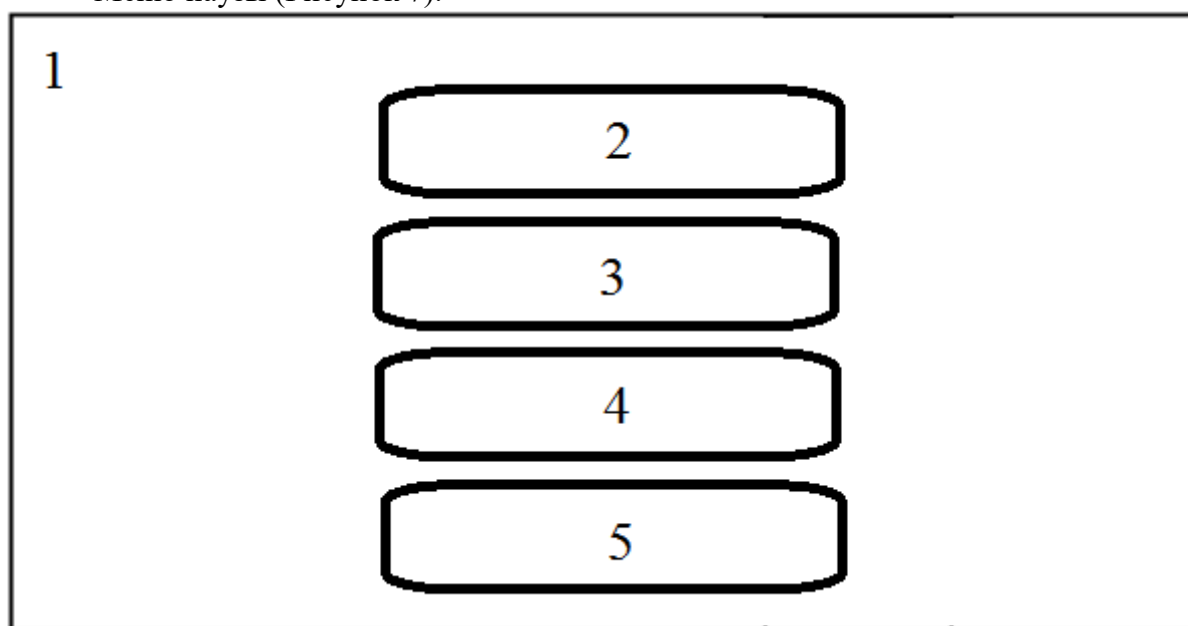


Рис. 7. Примерное расположение форм в Меню паузы

1 – Фон. Затемнение игрового процесса.

2 – Продолжить;

3 – Сохранить игру;

4 – Загрузить игру;

5 – Выход;

Логические модели игры (Рисунки 8-11):

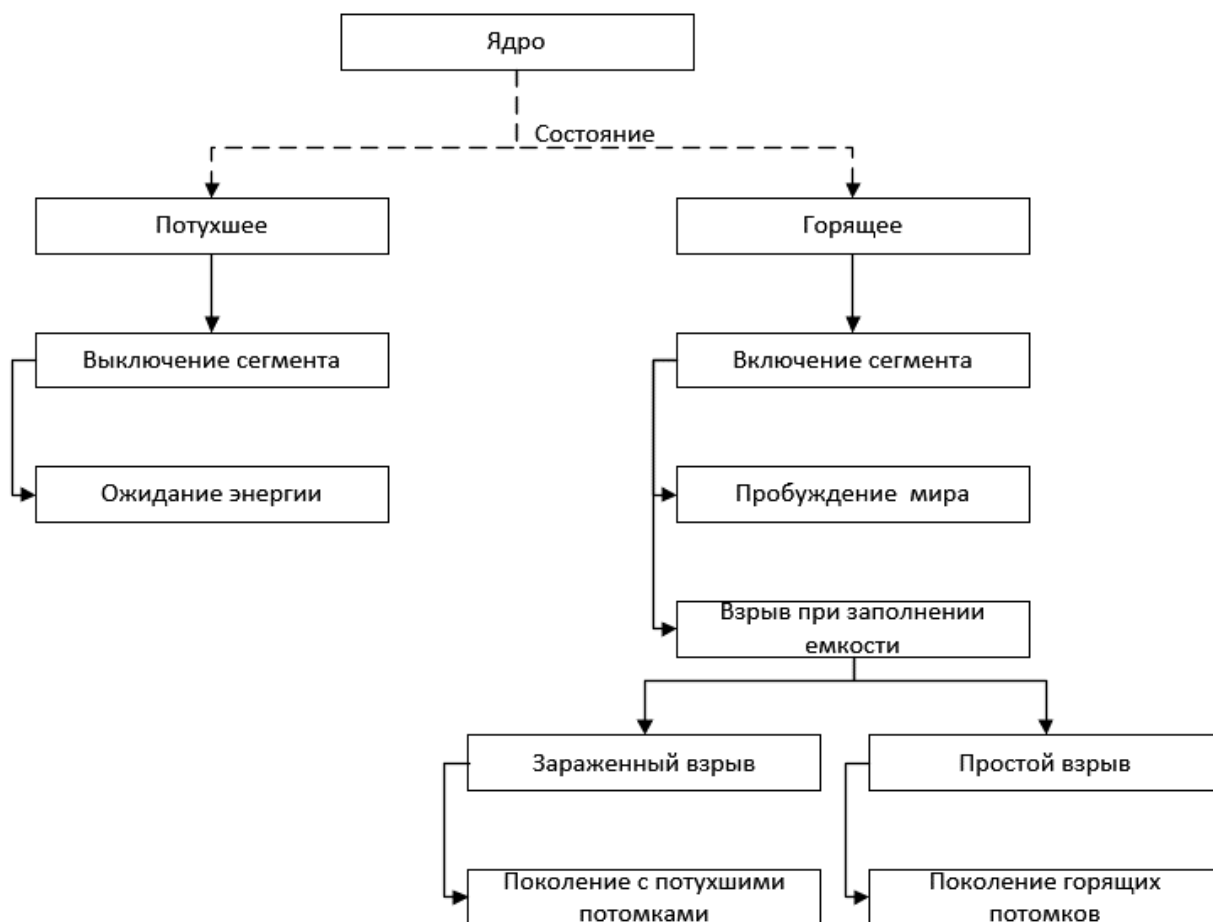


Рис.8. Логическая модель ядра

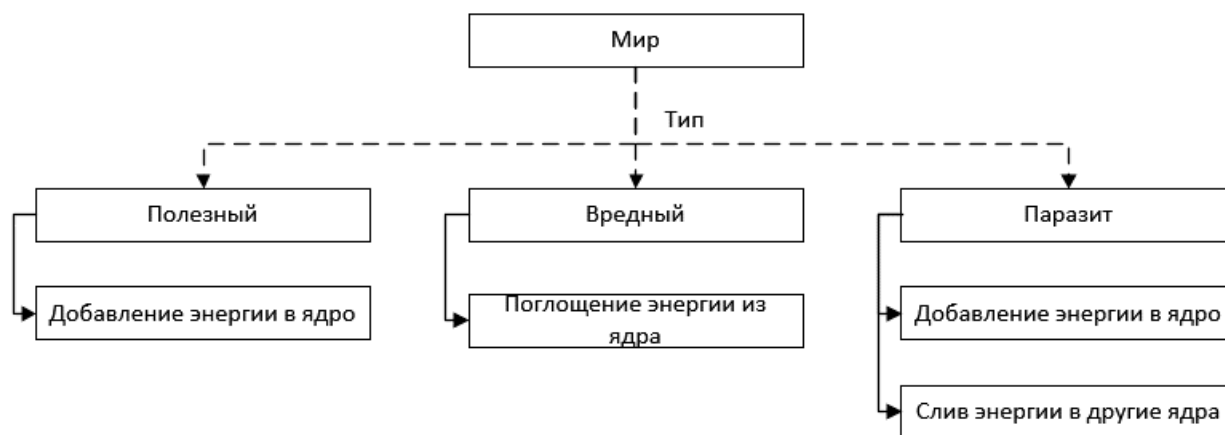


Рис.9. Логическая модель мира

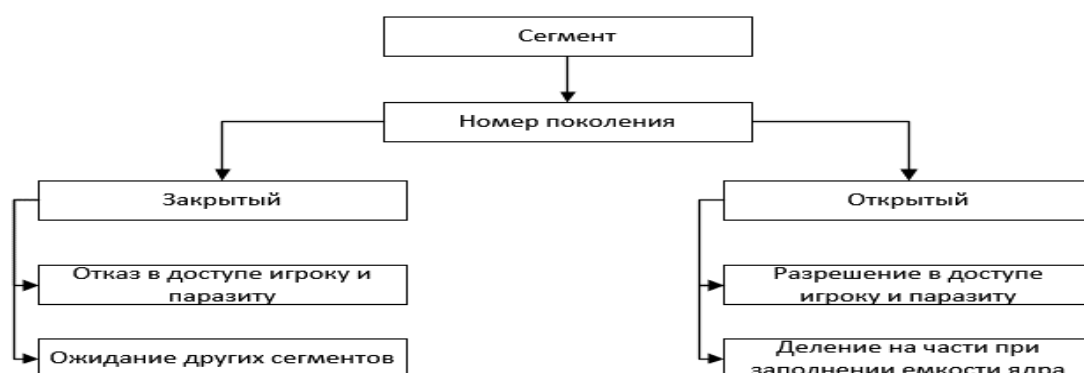


Рис.10. Логическая модель сегмента

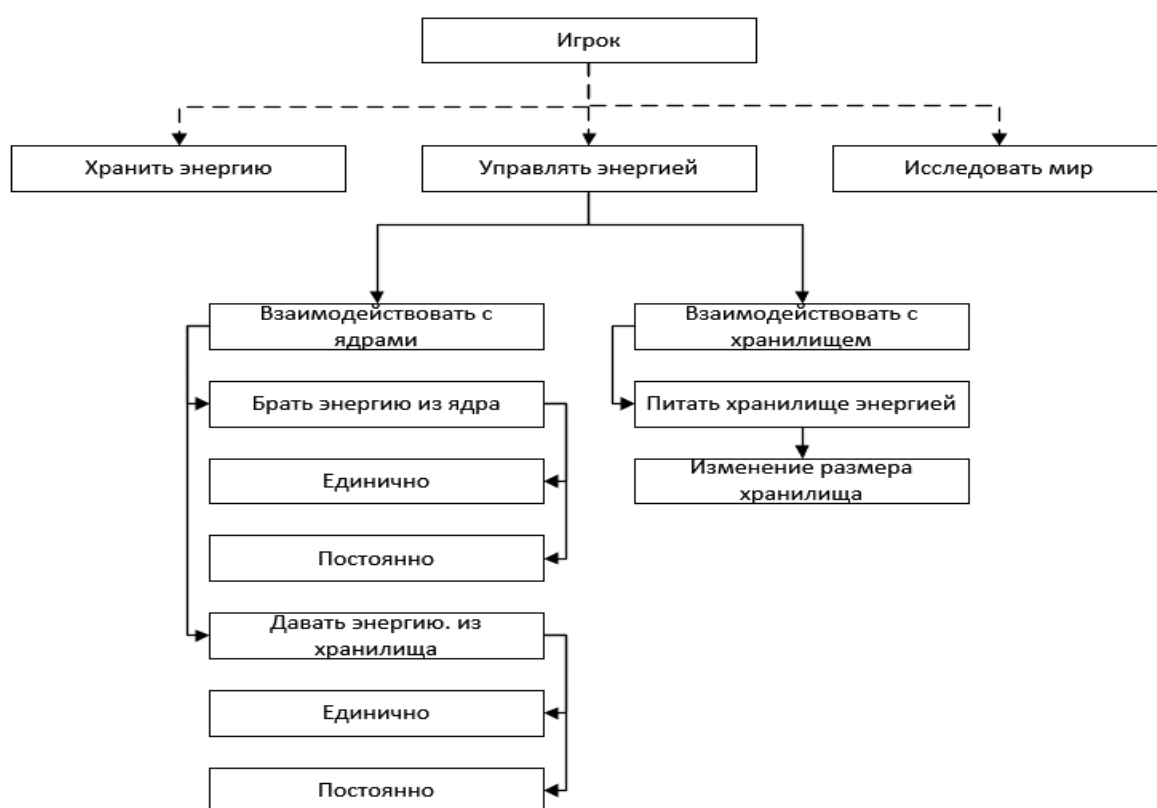


Рис.11. Логическая модель игрока

Варианты прохождения

Первый вариант. Если игрок везучий и за весь цикл ни разу не был сгенерирован вредный мир и мир паразит, то весь игровой процесс сводится к тому, чтобы игрок не растратил энергию в хранилище. Игрок будет следить за вселенной, постоянно брать из ядер энергию. Игрок выигрывает.

Второй вариант. В игре были сгенерированы миры паразиты и вредные миры. Перед игроком появилось такое препятствие как выключенные сегменты. Игроку предстоит постоянно давать энергию вредному ядру, чтобы нейтрализовать поглощение энергии и заполнить емкость ядра. Кроме того, игрок должен искать паразита, интуитивно с помощью исследований. Паразита можно вычислить не только с помощью исследования, но и с помощью наблюдения. Если какой-то сегмент дал потухших потомков, то значит, у этого сегмента один из соседей является паразитом. Или же, если в каком-то сегменте с полезным миром, без вмешательства игрока, ядро никак не заполнится, постоянно теряя часть энергии в

емкости, то этот сегмент содержит не полезный мир, а мир паразит. При обнаружении мира паразита необходимо влить достаточное количество энергии в его ядро, чтобы тот взорвался.

Третий вариант. В игре много выключенных сегментов, игроку приходится тратить много энергии, чтобы их включить, но миры в этих сегментах вредные. Игроку нужно приложить все усилия для того, чтобы правильно распределить получаемую и затрачиваемую энергию

Дальнейшие перспективы развития и возможное применение в других областях науки.

На данный момент в игре реализованы лишь основные, упрощенные правила вселенной в 2D формате. В будущем игру можно будет перевести в 3D, и перенести в игру полноценно все правила вселенной, с модернизацией интерфейса и добавлением новых возможностей в игре. Например, добавление процедурного генерирования для сегментов, ядер и миров, процедурного текстурирования для миров и т.п.

На данном этапе развития, поведение объектов в игре напоминает поведение клеток в цитологии. Поведение полезных сегментов сравнивают с делением обычной клетки, а поведение вредных или паразитов сегментов сравнивают с поведением онкологических клеток. Таким образом, игра может быть использована для моделирования поведения клеток.

УДК 004.357

Ю.О. Воронецкий, студент 4-го курса,

С.С. Журба, студент 4-го курса,

Абрамов Т.А., старший преподаватель.

Севастопольский государственный университет

г. Севастополь, Российская Федерация

z3xxjv@gmail.com

РАЗРАБОТКА АВТОМАТИЗИРОВАННОЙ СИСТЕМЫ «УМНЫЙ ДОМ»

Аннотация: В данной статье рассматривается проект разработки автоматизированной системы управления бытовой техникой и взаимодействия с пользователем «Умный дом» и основные принципы её функционирования.

Ключевые слова: умный дом, датчик, Raspberry, голосовое управление, Linux, Raspbian.

Annotation: This article reviews a project of an automated system for household appliances control and user interaction development ("Smart home"). Also, the main principles of its functioning are being described.

Keywords: smart home, sensor, Raspberry, voice control, Linux, Raspbian.

Введение

Ежегодно мировой спрос на автоматизированные системы, способные облегчить жизнь рядовому современному человеку, растёт. По мере развития научно-технического процесса людям хочется сделать свою жизнь комфортнее и безопаснее. Таким требованиям и соответствует рассматриваемая система «Умный дом», которая представляет собой совокупность датчиков и переключателей, объединённых единым центром приёма, обработки и передачи информации [1,2].

Описываемый аппаратно-программный комплекс должен обладать достаточным быстродействием, а также соответствовать запросам конечного пользователя, а, следовательно, быть универсальным, гибким и удобным в настройке, а также обладать интуитивно понятным управлением и однозначно понимаемой обратной связью.

Целью настоящей работы является разработка подобной системы – готового и конкурентоспособного продукта (в первую очередь, за счёт дешевизны реализации и удобства использования), способного стать альтернативой уже имеющимся решениям на современном рынке РФ и популяризировать подобные технологии среди населения.

Следовательно, приоритетом разработки являлись небольшая стоимость относительно аналогов, представленных на российском рынке и простота установки и первичной настройки. Именно простота установки, которая невозможна без унификации элементов подобных проектов, считается краеугольным камнем при проектировании таких систем в Российской Федерации. В то время как, например, в Европе подобные комплексы являются вполне обыденной вещью, для наших соотечественников они всё ещё в диковинку. Именно поэтому компании, занимающие данную нишу в нашей стране, преимущественно выбирают строго индивидуальный подход к клиенту вместо применения типовых решений, что, несомненно, влияет на сроки и стоимость реализации. Система должна также быть легко расширяема, что невозможно без уже упомянутой унификации её составных компонентов.

Программная часть

Система основывается на одноплатном микрокомпьютере Raspberry Pi 3 Model B, на котором производится анализ текущего состояния электрических приборов помещений, приём, распознавание и обработка информации с датчиков, и дальнейшая передача соответствующих команд на органы управления техникой. Операционная система – дистрибутив Linux под названием Raspbian.

Управление может производиться как голосовыми командами, так и с использованием дублирующего веб-интерфейса, доступного с любого современного гаджета (такое решение было принято для избежания проблемы кроссплатформенности клиентских приложений).

В качестве системы распознавания речи будет использована бесплатная (open-source) CMU Sphinx разработки университета Карнеги-Меллон, а конкретнее, её реализация под названием PocketSphinx, которая может работать в системе, базирующейся на процессорах архитектуры ARM (именно такой установлен в используемом нами микрокомпьютере

Raspberry). Данная система обладает широким инструментарием, позволяющим ей не просто распознавать последовательности произносимых слов, но и обучаться при помощи записи и анализа образцов голоса пользователя.

Помимо указанных выше средств, на микрокомпьютере (далее – сервер) работает скрипт, написанный на языке Python, реализующий общение сервера и датчиков и выключателей (далее – клиенты) по протоколу TCP/IP. Узлом связи в таком случае выступает маршрутизатор Wi-Fi, который также обеспечивает доступ сервера в сеть Интернет.

Доступ во Всемирную паутину необходим для получения своевременных обновлений программного обеспечения, а также для реализации вспомогательных функций, таких, как поиск информации в разнообразных поисковых сетях, что, несомненно, повышает интерактивность и способствует практически безграничному расширению сценариев взаимодействия пользователя с данной системой.

Система поддерживает как пользовательские, так и предустановленные режимы функционирования. Из предустановленных стоит отметить «стандартный», «режим экономии энергии», «режим охраны».

Режим экономии энергии работает следующим образом: если с датчиков движения во всех помещениях жилплощади не поступает сигнал, система автоматически отключает большинство электроприборов кроме тех, отключение которых может привести к неблагоприятным последствиям (например, холодильник). Также, в этом режиме локально отключается свет в помещениях, в которых не находится ни один жилец.

Присутствует также режим охраны помещения от несанкционированного доступа. Устроен он по такому принципу: для его отключения необходимо на мобильном устройстве выбрать пункт меню управления системой умного дома «Отключить защиту» и ввести пароль, заданный при постановке системы на защиту. Данное действие можно выполнять с любого устройства, авторизованного на сервере.

В стандартном режиме указанные выше меры по экономии энергии не предпринимаются.

Аппаратная часть

Как было указано выше, проектируемая система будет управляться одноплатным компьютером Raspberry Pi 3 Model B с модулями Wi-Fi и Bluetooth. Данный микрокомпьютер будет взаимодействовать с множеством датчиков:

- Микрофонный;
- Инфракрасный;
- Фотодатчик;
- Электрический датчик;

Датчик - измерительный прибор, предназначенный для создания сигнала о поступающей на него информации в необходимой форме, для дальнейшей передачи, анализа, преобразования или обработки без восприятия различных наблюдателей.

Микрофонный датчик - это датчик, принимающий и передающий аналоговый сигнал на АЦП для преобразование в цифровой сигнал.

Инфракрасный датчик движения - датчик, основанный на инфракрасном излучении тел или предметов, при изменении излучения приводится в действие.

Фотодатчик - прибор или устройство, которое способно позиционировать различные объекты.

Фотодатчики будут реагировать на освещение как в помещении, так и за его пределами.

Электрические датчики в данном проекте будут использоваться для управления бытовой техникой и электрическими приборами в доме.

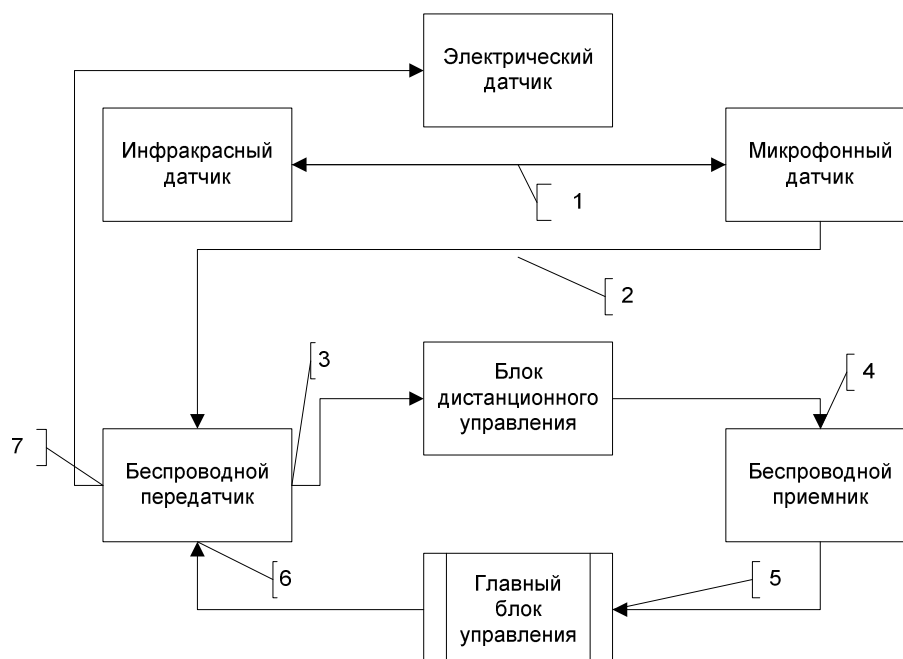


Рис.1. Упрощенная схема работы разрабатываемой системы

При отсутствии жильцов в целях экономии электроэнергии по сигналам от инфракрасных датчиков будут выключаться энергопотребляющие приборы.

Голосовой датчик будет состоять из микрофона и реле передатчика. Аудиопередатчик будет настроен на приемник, который непосредственно подключен к управляющему одноплатному компьютеру, где и будет происходить распознавание голоса для выполнения определенной команды.

По всему дому будут расположены динамические головки широкополосного звукового спектра для озвучивания работы компьютера, например для вопроса об авторизации при входе в дом, будет озвучен данный вопрос.

В качестве блока питания для одноплатного компьютера будет использоваться компьютерный блок питания 2.5А 5В.

В целях безопасности, а также для обеспечения автономности системы будет использоваться запасной внешний аккумулятор.

Таким образом, в статье был рассмотрен проект разработки автоматизированной системы управления бытовой техникой и взаимодействия с пользователем «Умный дом» и основные принципы её функционирования.

Библиографический список

1. Asadi A. Raspberry Pi for Beginners: All you need to know to get started with your Raspberry Pi / A. Asadi - Bournemouth Dorset: "Imagine Publishing Ltd.", 2014. - 180 с.
2. Шило В.Л. Популярные цифровые микросхемы: Справочник. – М.: Радио и связь, 1987 – 352с.

УДК 681.3.053

В.В.Голованев, А.Г.Шлемко студент 3-го курса

Научный руководитель: С.Н.Фисун, канд. техн. наук, доцент

Севастопольский государственный университет

РАЗРАБОТКА ЛЕКСИЧЕСКОГО И СИНТАКСИЧЕСКОГО АНАЛИЗАТОРОВ ПРИ ПРОЕКТИРОВАНИИ КОМПИЛЯТОРА

Аннотация: В статье демонстрируется значение лексического и синтаксического анализаторов при разработке компиляторов для компьютерных инженеров. Рассматриваются методы построения блоков лексического и синтаксического анализатора. Основное внимание уделяется автоматной модели при разработке лексического анализатора и методу нисходящего синтаксического анализа LL(1).

Ключевые слова: лексический анализатор, управляющий автомат, генератор лексических анализаторов Lex, синтаксический анализатор, метод LL(1).

Annotation: The article demonstrates the importance of scanners and parsers development for the computer engineers. Methods for constructing blocks of lexical and syntactical analyzers were considered. The main focus is on the model of automaton in the development of the scanner and the method of top-down parsing LL(1).

Key words: scanner, control automat, lexical analyzer generator Lex, parser, method LL(1).

Лексический анализатор представляет собой первую фазу компилятора, цель которого состоит в переводе исходной программы на стандартный входной язык компилятора и преобразовании ее к виду, удобному для дальнейшей обработки на этапах синтаксического и семантического анализа[1].

Блок лексического анализа предназначен для решения следующих задач:

- 1) перекодировка исходной программы, рассматриваемой как входная строка, и приведение её к стандартному виду (формирование списка лексем);
- 2) выделение идентификаторов и служебных слов;
- 3) выделение и перевод в машинную форму числовых констант;
- 4) формирование таблиц идентификаторов и констант;
- 5) удаление пробелов и комментариев.

При проектировании компилятора возможны случаи объединения лексического и синтаксического анализаторов, так как спецификации лексем можно внести при помощи регулярных выражений внутри описываемой грамматики языка. Однако есть несколько серьезных доводов в пользу отделения лексического анализа от синтаксического:

1. Значительная часть времени компиляции тратится на сканирование литер. Выделение же позволяет нам сконцентрировать внимание на сокращении времени.

2. Синтаксис символов можно описать в рамках очень простых грамматик. Если отделить сканирование от синтаксического распознавания, то можно разработать эффективную технику разбора, которая наилучшим образом учитывает особенности этих грамматик.

3. Развитие языков высокого уровня требует внимательного отношения как к лексическим, так и синтаксическим свойствам языка. Разделение этих двух свойств позволит нам исследовать их независимо друг от друга.

Сейчас можно выделить 3 основных метода реализации лексического анализатора.

1. Использование генератора лексических анализаторов, такого как Lex, для создания лексического анализатора по спецификациям, основанным на регулярных выражениях. В этом случае генератор предоставляет функции для чтения и буферизации ввода.

2. Написание лексического анализатора на подходящем языке программирования с использованием возможностей ввода-вывода этого языка для чтения входной информации.

3. Написание лексического анализатора на языке ассемблера и явное управление процессом чтения входной информации.

Первый способ самый легко реализуемый из всех перечисленных, так как порядок определения лексем мы задаем при помощи регулярных выражений и специального языка Lex, который достаточно просто понять. Однако получаемая в результате диаграмма переходов получается избыточной, а регулярными выражениями не всегда просто определить возможные наборы лексем языка, что в итоге дает нам неэффективный блок лексического анализатора.

Третий способ, наоборот, дает нам самый эффективный лексический анализатор, но процесс создания требует отличного знания ассемблера и процесс отладки усложнен отсутствием дружелюбного к пользователю интерфейса.

В итоге был выбран второй способ, который представляет собой «золотую середину», а в качестве языка высокого уровня – Java. Полученная программа представляет собой распознаватель языка, который использует таблицу переходов для определения служебных слов и идентификаторов (Рисунок 1).

	_	H	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11	S12	S13	S14	S15	S16	S17	S18	S19
V	H, L3	e	e	e	e	e	e	e	e	e	e	e	e	e	e	e	e	e	e	e	e
Y	H, L9	e	e	e	e	e	e	e	e	e	e	e	e	e	e	e	e	e	e	e	e
A	e	e	S10	e	e	e	e	e	e	e	e	e	e	e	e	e	e	e	e	e	e
B	S1	e	e	e	e	e	e	e	e	e	e	e	e	e	e	e	e	e	e	e	e
C	S2	e	e	e	e	e	e	e	e	e	e	e	e	e	e	e	e	e	e	e	e
D	S3	e	e	e	e	e	e	e	e	e	e	H, L6	e	e	e	e	e	e	e	e	e
E	S4	S9	e	e	e	e	e	e	e	e	e	e	S16	e	e	e	e	e	e	H, L11	e
L	e	e	e	e	e	e	e	e	e	e	S15	e	e	e	e	H, L4	e	S19	e	e	e
G	e	e	e	e	e	e	e	e	e	S14	e	e	e	e	e	e	e	e	e	e	e
I	S5	e	e	e	e	e	e	e	e	e	e	e	e	S17	S18	e	e	e	e	e	e
N	H, L1	e	e	e	S11	e	e	e	e	e	e	e	e	e	e	e	H, L10	e	H, L5	e	e
F	e	e	e	e	e	H, L8	e	e	e	e	e	e	e	e	e	e	e	e	e	e	e
T	S6	e	e	e	e	e	e	e	e	e	e	e	e	e	e	e	e	e	e	e	e
H	e	e	e	e	e	e	S12	S13	e	e	e	e	e	e	e	e	e	e	e	e	e
W	S7	e	e	e	e	e	e	e	e	e	e	e	e	e	e	e	e	e	e	e	e
O	e	e	e	H, L12	e	e	e	e	e	e	e	e	e	e	e	e	e	e	e	e	e
;	H, L7	e	e	e	e	e	e	e	e	e	e	e	e	e	e	e	e	e	e	e	e
:	S8	e	e	e	e	e	e	e	e	e	e	e	e	e	e	e	e	e	e	e	e
=	e	e	e	e	e	e	e	e	H, L2	e	e	e	e	e	e	e	e	e	e	e	e
\$	K	e	e	e	e	e	e	e	e	e	e	e	e	e	e	e	e	e	e	e	e

Рисунок 1. Пример управляющего автомата

В результате мы получаем 3 таблицы: таблица лексем, таблица идентификаторов и таблица определения условных обозначений лексем. В дальнейшем они используются как входные данные для этапа синтаксического анализа.

Синтаксический анализ - процесс сопоставления линейной последовательности лексем (слов, токенов) естественного или формального языка с его формальной грамматикой. Можно выделить два типа алгоритмов синтаксического анализа:

1. Нисходящий разбор - правила грамматики раскрываются, начиная со стартового символа, до получения требуемой последовательности лексем. Наибольшее распространение получили метод рекурсивного спуска и LL(1).

2. Восходящий разбор - правила восстанавливаются из правых частей, начиная с лексем и кончая стартовым символом. Наибольшее распространение получили методы LR(1) и предшествования[1, 2]

В процессе разработки компилятора был выбран метод LL(1). Синтаксический анализатор для данного метода просматривает входной поток лексем только на один символ вперед при принятии решения о том, какое правило грамматики необходимо применить.

Преимуществами данного метода являются: высокое быстродействие и сравнительно небольшой размер управляющей таблицы. Основным недостатком метода LL(1) является необходимость устранения в правилах исходной грамматики левой рекурсии и выполнения левой факторизации для альтернативных правил, которые начинаются с одинаковых символов.

Перед применением метода LL(1) строится управляющая таблица, с помощью которой задается однозначное соответствие правил грамматики для символа, находящегося на вершине рабочего стека, и текущего анализируемого входного символа. Строкам таблицы соответствуют нетерминальные символы преобразованной грамматики, а столбцам — терминальные символы. В результате работы блока синтаксического анализа формируется выходной стек или дерево разбора.

Рассмотрим пример разбора входной цепочки «U+C\$» для заданных правил (см. рис 2 и 3).

L	U	L	C	+	-	(	*	D	M	)	\$
	1	1	1	1	1	1	E	E	E	E	E
<F>	5	5	5	5	5	5	E	E	E	E	E
<P>	10	11	12	13	14	15	E	E	E	E	E
<X>	E	E	E	2	3	E	E	E	E	4	4
<Y>	E	E	E	9	9	E	6	7	8	9	9

Рисунок 2. Пример управляющей таблицы синтаксического анализатора

```

<B>: <F><X>
<X>: +<F><X>
<X>: -<F><X>
<X>: e
<F>: <P><Y>
<Y>: *<P><Y>
<Y>: D<P><Y>
<Y>: M<P><Y>
<Y>: e
<P>: U
<P>: L
<P>: C
<P>: +C
<P>: -C
<P>: (<B>)

```

Рисунок 3. Пример списка правил синтаксического анализатора

1) Стартовая вершина (соответствует первой строчке рисунка 2); идет поиск столбца, который соответствует терминалу 'U'; найден номер правила (см. рисунок 3) на пересечении строки с столбца;

2) Из списка правил (см. рис 3) выбирается правило, номер которого был только что найден (это правило номер 1); из стека выталкивается нетерминал и записывается правая часть первого правила в обратном порядке;

3) Для верхушки стека (нетерминала <F>) из управляющей таблицы выбирается номер очередного правила, который находится на пересечении строки соответствующего нетерминала и искомого терминала; это правило 5;

4) Нетерминал <F> выталкивается из стека и заталкивается правая часть пятого правила в обратном порядке; для очередной верхушки стека ищется номер правила в управляющей таблице; это номер 10;

5) По 10 правилу был найден первый терминал входной цепочки; аналогично для всех оставшихся в стеке нетерминалов.

В конечном счете будет достигнут конец просматриваемой цепочки, а в стеке останется только символ '\$'. Это будет означать, что входная цепочка не содержит синтаксических ошибок (см. рисунок 4).

```

M(<B>, U) = 1;
M(<F>, U) = 5;
M(<P>, U) = 10;
M(U, U) = C;
M(<Y>, +) = 9;
M(<X>, +) = 2;
M(+, +) = C;
M(<F>, C) = 5;
M(<P>, C) = 12;
M(C, C) = C;
M(<Y>, $) = 9;
M(<X>, $) = 4;
Analizing stopped: $ char was encountered.
=====
Input chain is correct;
Tracer: 1 5 10 9 2 5 12 9 4
=====
Output Stack:
<X>, 6
<Y>, 8
C, 9
<P>, 8
<F>, 6
+, 6
<X>, 1
<Y>, 2
U, 3
<P>, 2
<F>, 1
<B>, 0

```

Рисунок 4. Результат синтаксического анализа входной цепочки

Библиографический список

1. Лебедев В.Н. Введение в системы программирования/В.Н. Лебедев. – М.: Статистика, 1975. – 312 с.
2. Ахо А. Компиляторы: принципы, технологии и инструменты/А. Ахо, Р. Сети, Д. Ульман: Пер. с англ. М.: Вильямс, 2003. – 768 с.

УДК 681.3.053

А.В. Гримута, студент 4-го курса

Научный руководитель: В.И. Шевченко, канд. техн. наук, доцент

Севастопольский государственный университет

СИСТЕМА АВТОМАТИЗИРОВАННОГО ДОКУМЕНТООБОРОТА КАФЕДРЫ НА ПЛАТФОРМЕ «1С: ПРЕДПРИЯТИЕ»

Аннотация: В статье на примере автоматизации процесса «Повышение квалификации сотрудников» рассматриваются возможности разработанной системы автоматизации документооборота кафедры. Приводятся результаты проектирования подсистемы «Повышение квалификации сотрудников» в виде IDEF0 и DFD-диаграмм, полученных с помощью CASE-средства RAMUS. В виде диаграммы UML описан алгоритм взаимодействия пользователя с системой АИС «Кафедра». Рассматривается визуальный интерфейс и экранные формы подсистемы «Повышение квалификации сотрудников», созданные на платформе «1С:Предприятие».

Ключевые слова: автоматизация документооборота, управление бизнес-процессами, повышение квалификации, «1С:Предприятие», АИС «Кафедра».

Annotation: The article is about application of the automated document management system of the university department. The results of designing subsystem «Employee skill improving» in the form of IDEF0 and DFD diagrams using RAMUS CASE-tools are shown. The algorithm of the «user»-AIS «Department» interaction is presented in the form of UML-diagram. «Employee skill improving» was created on the basis of «1С:Enterprise» platform.

Key words: Automation of workflow, management of business processes, employee skill improving, «1С:Enterprise», AIS «Department».

Введение

На сегодняшний день задача совершенствования процессов принятия решений при управлении образовательным процессом ВУЗа тесно связано с развитием его информационной инфраструктуры в целом и, в частности, решением задачи внедрения системы электронного документооборота кафедр. Кафедра ВУЗа имеет значительную по объемам базу бумажных и архивных документов, связанных с сопровождением учебного процесса, научной работой, организационно-воспитательной деятельностью и процессами повышения компетенций профессорско-преподавательского состава, путем повышения квалификации.

В данной работе на примере автоматизации процесса «Повышение квалификации сотрудников» рассматриваются возможности разработанной системы автоматизации документооборота кафедры АИС «Кафедра» на платформе 1С: 8.3, представленная статья является продолжением работы [1].

В настоящее время непрерывность профессионального образования и развития профессиональных компетенций является обязательным условием востребованности сотрудников высших учебных заведений как специалистов. Это связано с постоянными изменениями актуальности содержания учебных материалов (особенно в ИТ-сфере), способов и методик преподавания, общих принципов воспитания и обучения, и т.п. что приводит к необходимости проведения работ по повышению педагогической квалификации сотрудников образовательных учреждений. Основная задача повышения квалификации сотрудников образовательных учреждений связана с необходимостью установления соответствия между уровнем подготовленности специалистов к реализации поставленных перед учреждением, задач и уровнем актуальных научных знаний, формирующих ядро управленческих процессов в системе образования.

Поскольку процесс повышения квалификации ограничен определенными регламентными рамками, метрики этого процесса входят в вектор показателей деятельности ВУЗа, актуальной является задача автоматизации документооборота в данном направлении с целью сокращения временных рамок формирования отчетности кафедры и повышения эффективности процессов управления.

Концептуальная модель подсистемы автоматизации документооборота для процесса «Повышение квалификации сотрудников»

Согласно регламентам, процесс повышения квалификации проводится в следующих формах:

- курсы повышения квалификации;
- профессиональная переподготовка по соответствующему направлению деятельности;
- стажировки в сторонних предприятиях и организациях;
- участие в работе тематических и проблемных семинаров.

Курсы повышения квалификации направлены на обновление теоретических и практических знаний специалистов в связи с возрастанием требований к уровню преподавания и необходимостью современных методов решения профессиональных задач в сфере образования.

Целью *профессиональной переподготовки* специалистов является получение ими дополнительных знаний, умений и навыков по образовательным программам, предусматривающим изучение отдельных дисциплин, разделов науки, техники и технологии, необходимых для выполнения нового вида профессиональной деятельности.

Основной целью *стажировки* является формирование и закрепление на практике профессиональных знаний, умений и навыков, полученных в результате теоретической подготовки. Стажировка осуществляется также в целях изучения передового опыта, приобретения профессиональных и организаторских навыков для выполнения обязанностей по занимаемой или более высокой должности. Стажировка может быть, как самостоятельным видом дополнительного профессионального образования, так и одним из разделов учебного плана при повышении квалификации профессорско-преподавательского состава.

Процессы повышения квалификации регламентируются соответствующим пятилетним и ежегодными планами по кафедре. По результатам повышения квалификации каждый преподаватель формирует отчет о результатах внедрения. На рис. 1 и 2 приведена концептуальная модель подсистемы «Повышение квалификации сотрудников» для АИС «Кафедра» в виде IDEF0 и DFD-диаграмм, которые получены с использованием CASE-средства RAMUS [2].

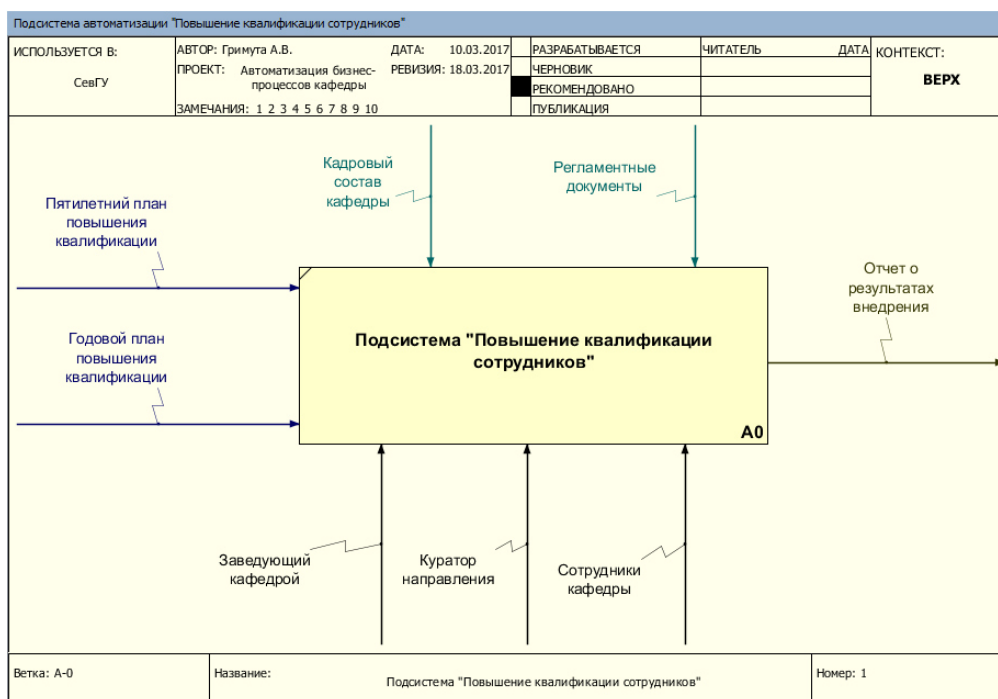


Рис. 1 – Контекстная диаграмма модели процесса «Повышение квалификации сотрудников» в нотации IDEF0

Диаграмма, представленная на рис. 2, отражает потоки данных, которые образуются в подсистеме, функциональные блоки и хранилища данных. В первом хранилище находится список-перечисление всех доступных видов стажировок. Во втором хранилище формируется актуальный список сотрудников кафедры, путем ввода данных о сотрудниках в

соответствующих экранных формах администратором системы. Третье хранилище содержит журнал электронных документов «Повышение квалификации», упорядоченных в хронологическом порядке, согласно данным о начале стажировки сотрудников. Четвертое и пятое хранилища формируются на основе данных о сертификатах и свидетельствах повышения квалификации государственного образца, загруженных пользователями системы.



Рис. 2 – Диаграмма потоков данных для подсистемы «Повышение квалификации сотрудников»

При разработке подсистемы «Повышение квалификации сотрудников» на платформе «1С:Предприятие 8.3» каждый блок диаграммы (рис. 3) находит свое отражение в создаваемой конфигурации [3, 4], а именно: хранилища данных представляются в виде объектов конфигурации типа «Справочник» или регистр сведений, а функциональные блоки – в виде программных обработчиков событий, электронного документа и формы отчета.

Автоматизация процесса в программной среде АИС «Кафедра»

Далее рассмотрим работу системы «АИС: Кафедра» на примере автоматизации процессов повышения квалификации. Для этого в конфигурации системы используется документ «Повышение квалификации сотрудников», в котором отображается информация о результате прохождения курса повышения квалификации. По проведению данного документа заполняется соответствующий регистр сведений. На его основании в последствии можно строить отчеты по сотрудникам, которым необходимо пройти курс повышения квалификации. Процесс автоматизации введения документооборота по данному направлению включает следующие этапы:

1. Ответственный за организационное направление (куратор направления) формирует задачу о необходимости прохождения курса (рис. 3), при этом указывая список сотрудников (подлежащих повышению квалификации).
2. По прохождению курса сотрудник кафедры должен ввести данные в документ «Повышение квалификации сотрудников» и провести его (рис. 4). При этом система «АИС: Кафедра», позволяет прикрепить копию сертификата или свидетельства государственного образца в данный документ при их наличии.
3. После чего пользователь должен закрыть данную задачу, тем самым проинформировав ответственное лицо (рис. 5).
4. На основе введенных документов формируется соответствующий журнал, в котором ответственное лицо (куратор проекта) может проконтролировать или откорректировать вводимую пользователями информацию (рис.6).

Рис.3 – Создание новой задачи

Рис.4 – Заполнение документа «Повышении квалификации»

Рис.5 – Отображение задачи перед закрытием.

Повышение квалификации сотрудников					
Создать		Поиск (Ctrl+F)			
Дата	Номер	Сотрудник	Дата начала ста...	Дата окончания стажир...	Вид стажировки
11.03.2017 23:51:38	000000002	Владимирова Елена Серг...	20.08.2016	11.10.2016	Курсы повышения квали...
12.09.2016 0:00:00	000000005	Мащенко Елена Николае...	01.09.2016	09.09.2016	Курсы повышения квали...
12.03.2017 10:46:20	000000006	Шевченко Виктория Игор...	01.09.2016	09.09.2016	Курсы повышения квали...
11.03.2017 23:55:22	000000003	Моисеев Дмитрий Влади...	29.10.2016	30.10.2016	Курсы повышения квали...

Рис.6 – Журнал документов «Повышение квалификации сотрудников»

Схема процесса взаимодействия пользователей с системой «АИС: Кафедра» на примере вышеописанного бизнес-процесса в виде диаграммы на языке UML приведена на рис. 7.

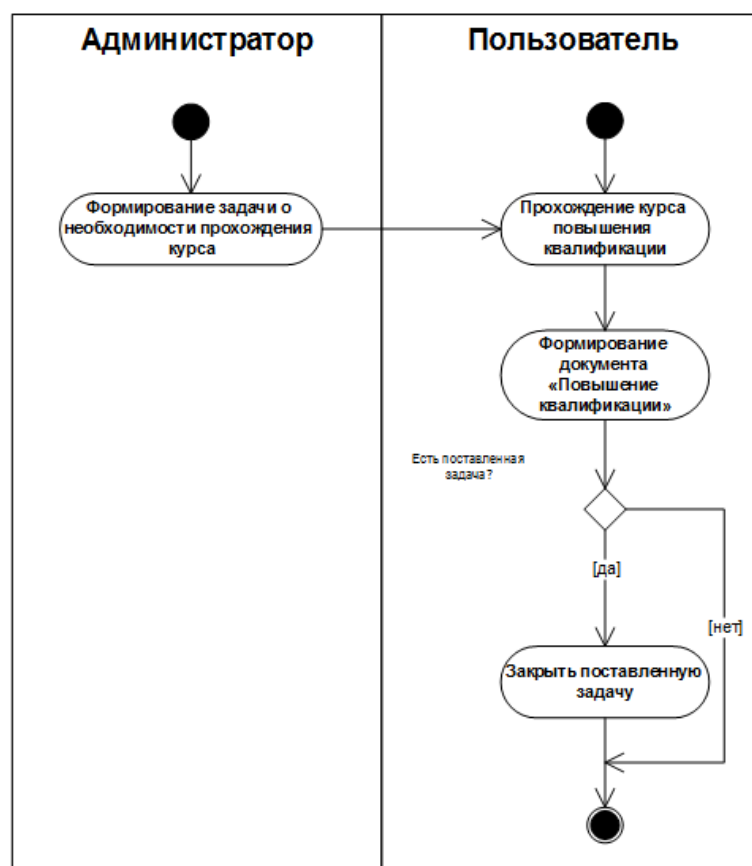


Рис.7 – Схема процесса взаимодействия пользователей с системой «АИС:Кафедра»

Выводы

Повышение квалификации является обязательным условием для развития навыков и способностей сотрудника. Однако данный бизнес-процесс не должен являться «бумажной проблемой». Потому использование предложенного программного продукта АИС: «Кафедра» позволит оптимизировать управленческую составляющую и перейти к безбумажному документообороту. Проиллюстрированный в данной статье фрагмент является лишь небольшим этапом автоматизации, при этом в целом данное программное решение разрабатывается с учетом пожеланий преподавателей, что делает ее более универсальной и адаптивной к задачам развития университета.

Библиографический список

1. Гримута А. В., Шевченко В.И. Автоматизация бизнес-процессов кафедры с использованием платформы «1С: Предприятие 8» // Материалы всероссийской научно-технической конференции молодых ученых, аспирантов и студентов «ИСУМ – 2016», Севастополь 19-21 мая 2016 г. – Севастополь: Изд-во СевГУ, 2016. – С. 342–345.
2. Кроссплатформенная система моделирования и анализа бизнес-процессов Ramus [Электронный ресурс]. – Режим доступа: <http://ramussoftware.com/> (дата обращения: 10.03.2017)
3. 1С:Информационно-технологическое сопровождение [Электронный ресурс]. – Режим доступа: <http://its.1c.ru/db/bspdoc> (дата обращения: 10.03.2017)
4. Радченко М.Г. Практическое пособие разработчика 1С: Предприятие 8.3 / М.Г. Радченко – М.: "1С-Паблишинг", 2013. – 964 с.

УДК 681.3.053

С.Н. Дзюба, студент 4-го курса

Научный руководитель: С.Н. Фисун, канд. техн. наук, доцент

Севастопольский государственный университет

ПРОГРАММНАЯ СИСТЕМА ДЛЯ АВТОМАТИЗАЦИИ РАЗРАБОТКИ СИНТАКСИЧЕСКИХ АНАЛИЗАТОРОВ*Аннотация: В статье рассматриваются принципы разработки синтаксического анализатора LL(1), а также инструмент разработки синтаксического анализатора ANTLR 4.**Ключевые слова: компилятор, синтаксический анализ, ANTLR, Теренс Парр.**Annotation: The article considers the principles of development of LL(1) parser and ANTLR 4.**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. Кроме того, 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 [1].

Синтаксический анализ является одним из начальных процессов компиляции программы. Он определяет, принадлежит ли некоторая последовательность лексем порождаемому грамматикой языку. Для любой контекстно-свободной грамматики может быть построен синтаксический анализатор, сложность которого не будет превышать $O(n^3)$ для входной строки длиной 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 входит 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.

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, приведем наиболее распространенные: ANTLR[2], YACC/BISON[3], LEX/FLEX и др.

Рассмотрим синтаксический анализатор, представленный недетерминированным конечным автоматом со стековой (магазинной) памятью. Основой функционирования такого автомата является управляющая таблица M , столбцам которой ставятся в соответствие все терминальные символы и символ конца строки ("λ"), строкам – все нетерминальные и терминальные символы и маркер конца стека ("⋄").

Таблица заполняется по следующим правилам:

1. Если $\langle A \rangle \rightarrow r$ - правило номер i заданной грамматики, то $M(\langle A \rangle, a) = i$ для всех a (кроме " λ "), являющихся терминальными префиксами цепочек, выводимых из r . Если таким префиксом может быть " λ ", то $M(\langle A \rangle, b) = i$ для всех b , являющихся терминальными символами, которые могут встречаться непосредственно справа от $\langle A \rangle$.

2. $M(a, a) = \text{"сдвиг"}$ для всех a , принадлежащих T (множеству терминальных символов).

3. $M(\$, \lambda) = \text{"допуск"}$.

4. Оставшиеся незаполненные элементы таблицы M получают значение "ошибка" (пустые клетки таблицы).

Анализатор работает по следующему алгоритму, представленному в таблице 1.

Таблица 1- Алгоритм работы анализатора.

№ п/п	Значение элемента таблицы	Интерпретация алгоритмом разбора
1	Номер i порождающего правила грамматики	Удаление символа из рабочего стека; запись этого символа в выходной стек; запись правой части правила номер i в рабочий стек справа налево, начиная с последнего символа
2	"сдвиг"	Удаление символа из рабочего стека; запись его в выходной стек; считывание следующего символа входной цепочки
3	«допуск»	конец работы
4	«ошибка»	Вывод сообщения об ошибке; конец работы

ANTLR (от англ. ANother Tool for Language Recognition- «еще одно средство распознавания языков»)- генератор нисходящих анализаторов для формальных языков, разработанный профессором Теренсом Парром из Университета Сан-Франциско. ANTLR преобразует 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 ANTLR является свободное распространение, использование единой нотации, предоставление сообщений об ошибках и восстановление после них, наличие визуальных средств разработки, например, ANTLR Works для IDE NetBeans.

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 символом "|" и должны заканчиваться символом ";". Также, существуют и другие элементы языка.

Для исследования зависимости времени работы анализатора от длины входной цепочки будет рассмотрен следующий небольшой фрагмент грамматики учебного языка программирования ASPLE-3:

1. $\langle S \rangle \rightarrow \text{While } V \text{ Do } \langle \text{OpSequence} \rangle \text{End}$
2. $\langle \text{OpSequence} \rangle \rightarrow \langle \text{Op} \rangle | \langle \text{Op} \rangle ; \langle \text{OpSequence} \rangle$
3. $\langle \text{Op} \rangle \rightarrow O | \langle \text{OpSequence} \rangle$

Фрагмент представляет оператор цикла с предусловием. В данной грамматике: $\langle S \rangle$ -оператор цикла, $\langle \text{OpSequence} \rangle$ -последовательность операторов, $\langle \text{Op} \rangle$ -оператор. Введем следующие сокращения: "W"-«While», "D"-«Do», " $\langle \text{OpSequence} \rangle$ "-" $\langle \text{Seq} \rangle$ ".

Для LL(1) синтаксического анализатора грамматика не должна содержать леворекурсивных правил, поэтому необходимо преобразовать второе правило следующим образом:

1. $\langle S \rangle \rightarrow WVD \langle \text{Seq} \rangle E$
2. $\langle \text{Seq} \rangle \rightarrow \langle \text{Op} \rangle \langle X \rangle$
3. $\langle X \rangle \rightarrow \varepsilon$
4. $\langle X \rangle \rightarrow ; \langle \text{Seq} \rangle$
5. $\langle \text{Op} \rangle \rightarrow O$
6. $\langle \text{Op} \rangle \rightarrow \langle S \rangle$

Для ANTLR 4 исходную грамматику необходимо представить в соответствии с правилами ее построения в файле <Имя грамматики>.g4

Файл грамматики Example.g4 выглядит следующим образом:

```
grammar Example; //название грамматики
options
{
    language = Java; //целевой язык, во втором случае будет Java
}
@parser::namespace { Generated }
@lexer::namespace { Generated }
prog : s+
s    : 'w' 'v' 'd' seq 'e';
Seq  : op
      | op 'seq';
Op   : 'o'
      | s;
WS   : (' ' | '\r' | '\n') -> channel(HIDDEN);
```

Входными данными для тестов является последовательность wvd[o;e]\$ (o;o повторяется от 1 раза до 131070 раз, каждый раз количество повторов увеличивается вдвое).

Результаты тестирования:

Кол-во символов	Время (нс)
8	1900805
10	450774
14	734869
22	1268983
38	2715319
70	5177333
134	9089379
262	13924320

Кол-во символов	Время (нс)
518	22126110
1030	41378843
2054	78232293
4102	33695978
8198	16195024
16390	22681983
32774	61467027
65542	138794898
131070	240205106

В сравнении с анализатором ANTLR, время работы LL(1) анализатора при длине входной цепочки приблизительно от 20 тысяч символов фрагмента грамматики начинает очень быстро возрастать, что проиллюстрировано на рисунке 1.

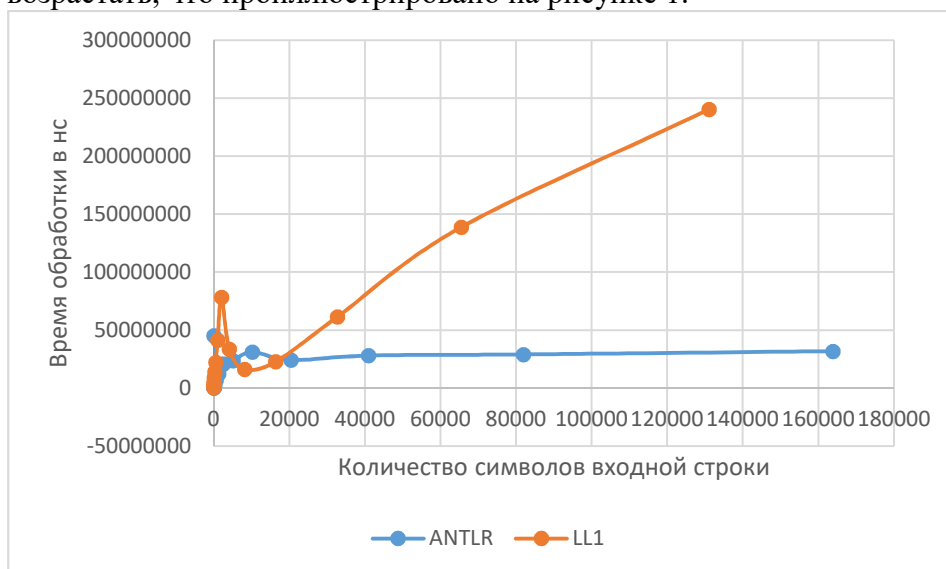


Рис. 1 - Графики зависимости времени работы синтаксического анализатора ANTLR 4 и LL(1) от длины входной цепочки

Исходя из полученных данных, можно сделать вывод, что инструмент ANTLR 4 позволяет значительно улучшить качество анализатора, а также сократить время на его разработку.

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. - 193 с.

УДК 004.415

А.В. Какаев, магистрант 1-го курса**Научный руководитель: В.И. Шевченко, к.т.н., доцент***Севастопольский государственный университет***АВТОМАТИЗИРОВАННОЕ ТЕСТИРОВАНИЕ ПО. ОБЗОР ИНСТРУМЕНТОВ НЕПРЕРЫВНОЙ ИНТЕГРАЦИИ**

Аннотация: В данной статье рассматривается проблема эффективности запуска автоматизированных тестов графического пользовательского интерфейса. Дается сравнение инструментов непрерывной интеграции. Обосновывается идея использования платформы виртуализации docker. Даются рекомендации по использованию инструментов для повышения эффективности автоматизированного тестирования.

Ключевые слова: непрерывная интеграция, разработка через тестирование, автоматизированные тесты, виртуализация, методологии проектирования ПО, эффективное программирование

Annotation: The article deals with problem of effective UI tests running. Continuous integration tools are compared in the article. It is proposed to use docker as virtualization platform. Recommendations of tools usage to increase automated testing efficient are given.

Key words: docker, continuous integration, Jenkins, test driven development, automated testing, UI tests, virtualization, effective testing, programming methodologies

Существует большое количество практик, методологий проектирования и разработки программного обеспечения. Как правило, методология выбирается в зависимости от требований к продукту, объемов финансирования разработки и допустимых рисков. Исходя из текущего уровня развития технологий и изменчивости рынка, наиболее подходящей методологией является гибкая методология (*agile*). По статистическим данным *agile* используется в 58% проектов. Гибкая методология наилучшим образом подходит для проектов с длительным жизненным циклом, постоянно адаптируемых к условиям рынка. Соответственно в процессе реализации проекта по этой методологии требования часто изменяются и реализуются итеративно [1]. Как показано на рис. 1, в каждой итерации (спринте) необходимо производить тестирование всего проекта.

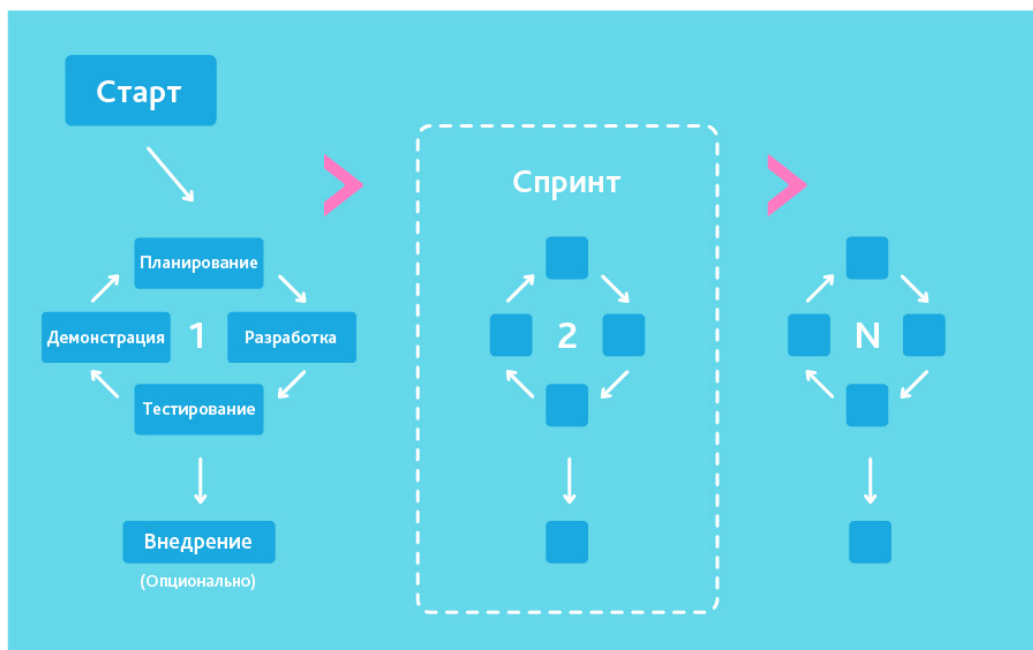


Рис. 1 – Жизненный цикл проекта по agile методологии

Для контроля качества и обеспечения автоматизации тестирования применяется практика непрерывной интеграции. Согласно этой практике несколько раз в день осуществляется сборка проекта и запуск автоматических тестов на специально выделенном сервере. Кроме того, согласно многим практикам разработки программного обеспечения, таким как TDD (*Test Driven Development*) запуск тестов осуществляется разработчиками в

самом процессе разработки программного обеспечения. При разработке через тестирование сначала разрабатываются тесты для проверки будущего функционала. После этого происходит написания кода, реализующего функционал, а затем рефакторинг написанного кода с постоянным запуском авто-тестов для проверки правильности [2].

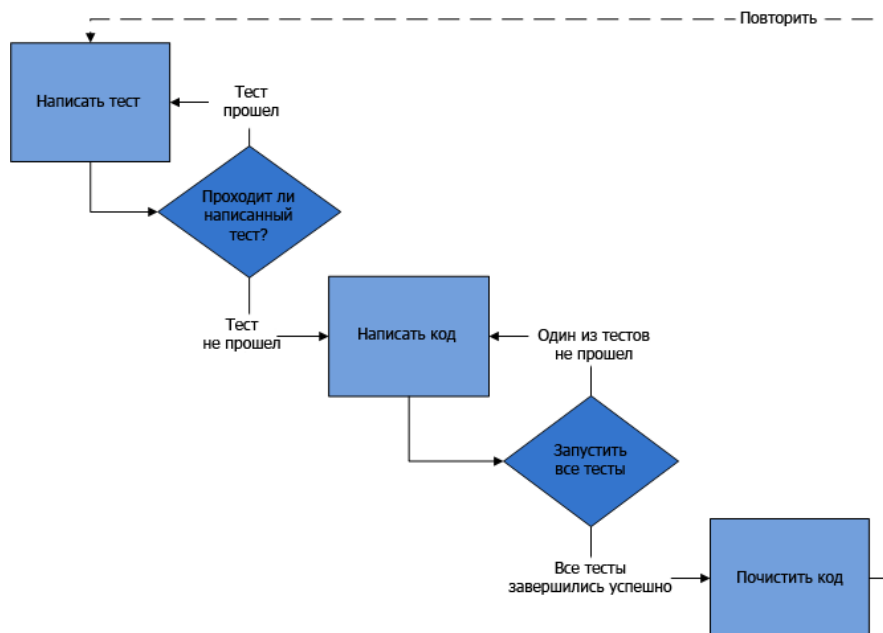


Рис. 2 – Разработка через тестирование

В основном при применении таких практик разрабатываются и запускаются модульные и функциональные тесты, но при разработке программного обеспечения с графическим пользовательским интерфейсом, в частности web-приложений, также применяются *UI* тесты. *UI* тесты имитируют реальные действия пользователя и выполняются значительно дольше модульных и функциональных тестов. Набор тестов, написанный для конкретной задачи может выполняться до нескольких минут, а тесты для всего приложения могут выполняться до нескольких часов в зависимости от размеров приложения. Кроме того, для проверки кроссбраузерности приложения необходимо запускать тесты на всех поддерживаемых браузерах, возможно разных версий. Так как в процессе разработки тесты запускаются очень часто, это значительно усложняет и замедляет процесс. Таким образом необходимо организовать автоматизированный запуск *UI* тестов по расписанию на различных браузерах для осуществления эффективной непрерывной интеграции. Также разработчикам необходимо предусмотреть возможность автоматического запуска тестов после каждого добавления нового кода в систему контроля версий.

Для осуществления непрерывной интеграции существует множество готовых решений, наиболее популярными из которых являются *TeamCity*, *Jenkins* – проект с открытым исходным кодом, написанный на *Java*, позволяет осуществлять запуск тестов по расписанию, при изменениях в системе контроля версий, по запросу на определенный *URL*, после завершения другой сборки в очереди. Возможности *Jenkins* можно расширять с помощью плагинов, поэтому они практически не ограничены [3]. *TeamCity* обладает несколькими преимуществами относительно *Jenkins*:

- позволяет произвести предварительное тестирование кода перед коммитом (сохранением, фиксацией). Предотвращает возможность сохранения программного кода, содержащего ошибки, нарушающего нормальную сборку проекта, путём удалённой сборки изменений перед коммитом.

- интеграция с системами оценки покрытия кода, инспекции кода и поиска дублирования кода [4].

Однако для использования *TeamCity* для больших проектов необходимо приобрести платную версию. Так как *Jenkins* удовлетворяет основным требованиям и является бесплатным продуктом, он вполне подойдет для организации непрерывной интеграции. *Jenkins* рекомендуется устанавливать на специально выделенный сервер, на котором и будет происходить запуск тестов. Для запуска тестов на *Jenkins* создается «задача». На рис. 3 показаны возможные виды задач в зависимости от целей и метода сборки проекта.

Item name

☐ **Freestyle project**
This is the central feature of Jenkins. Jenkins will build your project, combining any SCM with any build system, and this can be even used for something other than software build.

☐ **Maven project**
Build a maven project. Jenkins takes advantage of your POM files and drastically reduces the configuration.

☐ **External Job**
This type of job allows you to record the execution of a process run outside Jenkins, even on a remote machine. This is designed so that you can use Jenkins as a dashboard of your existing automation system. See [the documentation for more details](#).

☐ **Multi-configuration project**
Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.

☐ **Copy existing Item**
Copy from

OK

Рис. 3 – Создание задачи Jenkins

При настройке «задачи» указывается используемая система контроля версий для проекта, репозиторий с проектом, расписание запуска в указанном формате, скрипты сборки проекта, почта для рассылки результатов сборки и тестирования. С процессом настройки «задачи» и установкой необходимых плагинов можно ознакомиться на официальном сайте [3]. Для запуска тестов на различных браузерах можно создать несколько задач, которые будут запускать тесты на разных реальных и виртуальных машинах одновременно. Однако у такого подхода есть несколько недостатков:

- для каждой задачи необходимо создать, настроить и поддерживать виртуальную машину, что является достаточно трудоемким процессом;
- перед каждым запуском необходимо настраивать связь между *Jenkins* и машиной на которой будут запускаться тесты;
- каждая виртуальная машина потребляет реальные ресурсы и обладает недостаточным быстродействием;
- для каждой задачи необходимо иметь отдельную базу данных и возможно отдельную установку самого приложения, что значительно увеличит нагрузку на сервер.

Таким образом необходимо решить задачу изоляции приложения и базы данных для одновременного использования в нескольких тестах, минимизировать трудозатраты на настройку окружений. Для решения этих задач идеально подойдет *Docker*.

Docker – программное обеспечение для автоматизации развёртывания и управления приложениями в среде виртуализации на уровне операционной системы. Позволяет «упаковать» приложение со всем его окружением и зависимостями в контейнер, который может быть перенесён на любую *Linux*-систему с поддержкой *cgroups* в ядре, а также предоставляет среду по управлению контейнерами [5]. Чтобы создать окружение необходимо скачать необходимые контейнеры, содержащие необходимые приложения и связать их в один образ.

Так как контейнер является изолированной средой, однажды созданный контейнер может использоваться одновременно в нескольких более специфичных компонентах. То есть собранное из контейнеров окружение для запуска тестов физически хранится в одном образе, но может быть запущено параллельно для тестирования на различных браузерах.

Преимущества использования *Docker* сравнительно с виртуальными машинами:

- контейнеры с приложением и совсем необходимым окружением занимают значительно меньше места чем виртуальная машина;
- один образ с окружением можно использовать повторно;
- нет необходимости создавать отдельные базы данных для запуска тестов на разных браузерах;
- *Docker* контейнеры являются версируемыми в отличие от виртуальных машин, то есть они могут храниться в репозитории и если возникнет необходимость изменить окружение, достаточно сделать *pull* из репозитория и перезапустить образ.

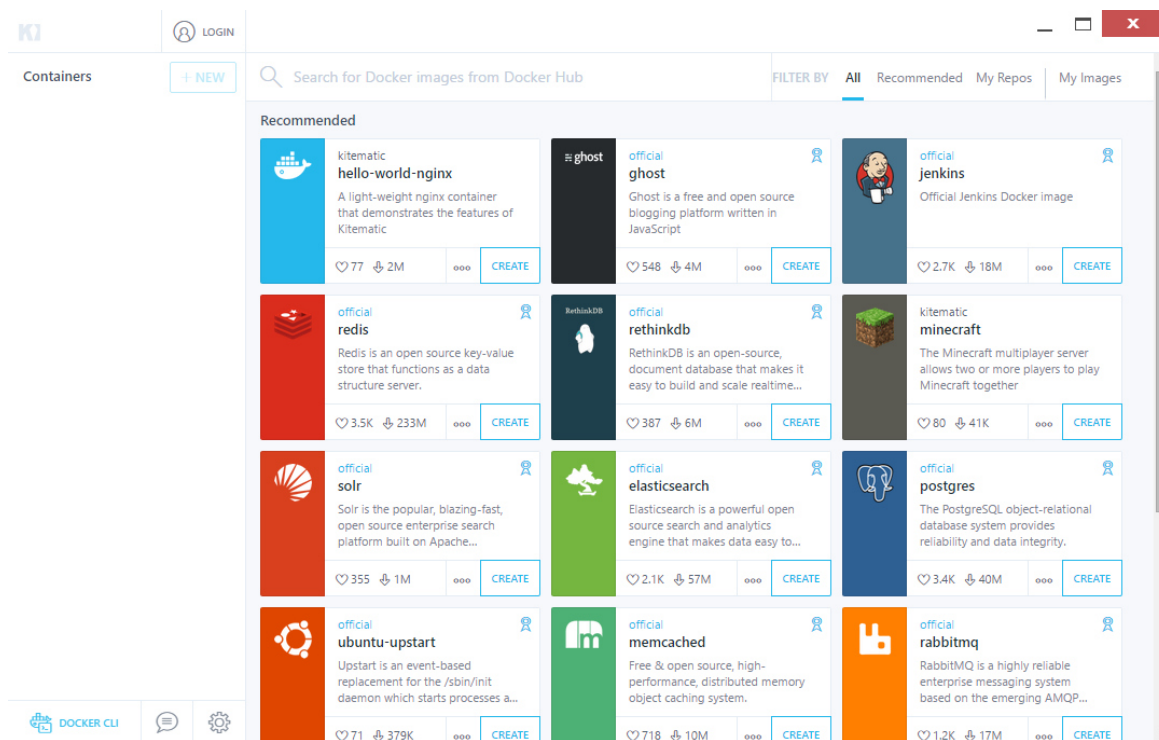


Рис. 4 – Готовые общедоступные контейнеры Docker

Таким образом, в работе был проведен обзор программных инструментов для организации непрерывной интеграции. Для организации сборок и запуска тестов программного обеспечения был выбран *Jenkins*. Для повышения эффективности запуска *UI* тестов рекомендовано использовать *Docker* как платформу виртуализации.

Библиографический список

1. Thomas Alva. Ещё раз про семь основных методологий разработки: сайт habrahabr. [Электронный ресурс]. 3 ноября 2015. URL: <https://habrahabr.ru/company/edison/blog/269789/> (дата обращения: 25.03.2017).
2. Test-driven development: Википедия : свободная энциклопедия. [Электронный ресурс]. 5 марта 2017. URL: https://en.wikipedia.org/wiki/Test-driven_development (дата обращения: 25.03.2017).
3. Jenkins: Википедия : свободная энциклопедия. [Электронный ресурс]. 14 марта 2017. URL: [https://en.wikipedia.org/wiki/Jenkins_\(software\)](https://en.wikipedia.org/wiki/Jenkins_(software)) (дата обращения: 25.03.2017).
4. TeamCity: Википедия : свободная энциклопедия.. [Электронный ресурс]. 9 февраля 2017. URL: <https://ru.wikipedia.org/wiki/TeamCity> (дата обращения: 25.03.2017).
5. Docker: Википедия : свободная энциклопедия. [Электронный ресурс]. 13 марта 2017. URL: <https://ru.wikipedia.org/wiki/TeamCity> (дата обращения: 25.03.2017).

УДК 681.3

М.В. Лысянский, студент 4-го курса**Научный руководитель: В.А.Строганов, ст. преподаватель***Севастопольский государственный университет***МОБИЛЬНОЕ ПРИЛОЖЕНИЕ ДЛЯ МЕДИЦИНСКОГО ПЕРСОНАЛА БОЛЬНИЦЫ**

Аннотация: в данной статье описывается мобильное приложения для медицинского персонала больницы. Автор описывает актуальные проблемы текущего процесса в медицинских учреждениях и способы их решения с помощью разрабатываемого мобильного приложения. В статье описаны технологии с помощью которых было создано мобильное приложение а также архитектура и функции которые обеспечивает мобильное приложение. Также описываются перспективы развития проекта.

Ключевые слова: медицина, мобильное приложение, медицинские учреждения, пациенты, медицинская информационная система.

Annotation: in this article narrates a mobile application for medical personnel in hospital. The author describes the actual problems of the current process in medical organisation and how to solve them using the developed mobile application. The article describes the technologies with which the mobile application was created and also the architecture and functions that provides the mobile application . Development prospects of the project are also depicted here.

Keywords: medicine, mobile application, medical organosations, patients, medical information system

На сегодняшний день одной из проблем в медицинских учреждениях является своевременный доступ к информации о пациентах, а также поддержка актуальной информации в электронных системах. Поиск соответствующих справок, историй болезней, анализов может занимать много времени, которое так необходимо для лечения пациентов. Заполнение данных в информационных системах для персональных компьютеров может быть утомительным и привязывает медицинского работника к стационарному компьютеру. Также одной из проблем является необходимость иметь при себе множество справок и историй болезней при обходе пациентов, что является утомительным для медицинского персонала.

Создание мобильного приложения для автоматизированной информационной системы позволит решить данные проблемы. Приложение сможет предоставить доступ к различным документам пациента с мобильного устройства (мобильный телефон, планшет), что уменьшит время на поиск документов и извлечения их из архивов, а также уберет необходимость медицинским работникам иметь при себе множество документов о пациентах. Помимо просмотра информации, приложение будет иметь возможность и редактировать ее, что упростит ведение документации в информационных системах и не будет привязывать работников в персональному компьютеру.

В рамках программы сотрудничества кафедры «Информационные системы» с «Городская больница №9» были выяснены основные требования работников больницы для подобного мобильного приложения:

- представление наглядной информации о пациентах;
- возможность изменять информацию;
- возможность оперативного доступа к информации;
- информирование о результатах лечения, процессе обследования пациента;
- голосовой ввод информации.

На основе выше собранных требований было принято решение реализовать приложение с клиент-серверной архитектурой. Серверная часть которого будет отвечать за хранение и предоставлять информацию мобильным клиентам, в то время как отображение самой информации и незначительная логика будет реализована на самом мобильном клиенте. Проанализировав информацию о популярности, распространенности мобильных операционных систем и их версий было принято решение о реализации мобильного приложения под операционную систему "Android" версии 4.0 и выше.

Разработка подобного мобильного приложения является инженерной задачей, с которой сталкиваются современные мобильные разработчики, основными вопросами в которой являются:

- выбор языка и средств разработки;
- выбор архитектуры мобильного приложения;
- выбор архитектуры клиент-серверного взаимодействия;
- выбор вспомогательных средств разработки (библиотек).

Для реализации приложения были выбраны стандартные средства разработки среда разработки программ "Android Studio"[1], ставшая официальным средством разработки "Android" приложений и язык программирования "Java".

Выбор архитектуры мобильного приложения является одной из важных задач, от которой зависят затраты на разработку системы, удобство поддержки, возможность тестирования, масштабируемость системы. Из множества вариантов была выбрана архитектура "MVP" ("Model-View-Presenter") + "rxJava", производная от "MVC" ("Model-View-Controller"). Данное архитектурное решение предлагает разбить приложение на следующие части:

- модель ("model"), является точкой входа к данным приложения;
- представление ("view"), являющиеся классом, устанавливающим состояние элементов графического интерфейса;
- предьявитель ("presenter"), класс устанавливающий связь между обработкой данных, получаемых из модели и вызовом методов представления, реализуя реакцию графических элементов на поступившие данные.

Стоит отметить, что данная архитектура не предполагает использование "rxJava", однако его использование обусловлено упрощением структуры приложения.

Для реализации клиент-серверного взаимодействия была выбрана "REST"-архитектура[2], преимуществами которой являются производительность, масштабируемость, простота интерфейсов, прозрачность системы взаимодействия, портативность компонентов, легкость внесения изменений. Для данной архитектуры, написание сетевых запросов не составит труда, если использовать библиотеку "Retrofit", которая предоставляет готовое решение для работы с сетевыми запросами, благодаря чему написание запросов не составит труда. Также стоит отметить возможность десериализации для данной библиотеки, что упрощает чтение ответов на запросы. Применение в связке "rxJava" и "Retrofit", делает обработку запросов простым и понятным.

Дизайн и графический интерфейс приложения является немало важной частью и нуждается в адаптивности к различным версиям операционной системы "Android", а также различным разрешениям экрана. Для этого воспользуемся библиотеками "Support Library" от разработчиков операционной системы "Android", данные средства дают возможность проектировать графический интерфейс таким образом, чтобы различные элементы поддерживались на устройствах с устаревшими версиями операционной системы Android. Одной из подзадач является визуализация информации в виде различного рода графиков и диаграмм. Использование библиотеки "MPAndroidchart" предоставляет готовое решение для разработчиков, однако недостатком может оказаться трудность при адаптации решения под свои нужды. Предоставление графической информации об анализах пациента (рентгенография, справки и т.д.) является важной функциональной составляющей приложения. Для отображения графических изображений существует множество готовых решений, одно из которых "Picasso" – библиотека с открытым исходным кодом, дающая возможность работы с графическими изображениями.

В результате было реализовано мобильное приложение, взаимодействующее с серверной частью и предоставляющее :

- информацию о пациенте;
- информацию об осмотрах пациента;
- информацию о назначениях для пациента;
- информацию об анализах пациента (в виде графических изображений);
- информацию о температурах пациента (построение графиков).

Данное приложение также предоставляет возможность пользователю навигации по отделениям и палатам.

В дальнейшем в данном мобильном приложении планируется реализовать следующий функционал:

- сохранение данных на устройстве;
- авторизация пользователей;
- возможность редактирования информации о пациенте;
- распознавание голоса, для поиска и ввода информации.

В заключении стоит отметить важность и актуальность данного проекта, внедрение его позволит решить проблему оперативного доступа и поиска информации, а также упростит ведение электронной документации медицинскими работниками.

Библиографический список

1. Android [Электронный ресурс] <https://developer.android.com/>: официальный сайт для разработчиков Android приложений. Дата обновления 2017. URL: <https://developer.android.com> (дата обращения: 24.03.2017)
2. Архитектура REST руководство по проектированию архитектуры REST [Электронный ресурс]. Дата обновления 2016. URL: <https://habrahabr.ru/post/38730/> (дата обращения: 24.03.2017)

УДК 681.3

Ю. И. Пашкова, студентка 4-го курса; А. С. Коваленко, студентка 4-го курса**Научный руководитель: М. А. Лебедева, старший преподаватель.***Севастопольский государственный университет***АВТОМАТИЗИРОВАННАЯ СИСТЕМА УЧЕТА И УПРАВЛЕНИЯ ВОДОМЕРНЫМИ ПОДКЛЮЧЕНИЯМИ ГУПС «ВОДОКАНАЛ»**

Аннотация: статья посвящена автоматизированной системе учета и управления водомерными подключениями для предприятия ГУПС «Водоканал». Система призвана заменить устаревшую систему, ранее имеющуюся на предприятии, в связи с увеличением объема данных и абонентов, а также увеличить скорость ее работы и при этом обеспечить надежную защиту персональных данных абонентов.

Ключевые слова: автоматизированная система, инвентаризация водометров, водомерное подключение.

Annotation: the article describes automated system of accounting and control water meter connections for the enterprise SUES «Vodokanal». The system is designed to replace the legacy system that previously held the company. Due to increase in the amount of data and clients, system should work fast and provide the protection of personal data.

Key words: Automated system, inventory of water meters, water connection.

Коммунальные предприятия, обеспечивающие население услугами первой необходимости (водоснабжение, электроэнергия, природный газ), хранят большие данные о подключённых абонентах, в связи с чем встает вопрос о введении на предприятии информационной системы (ИС), позволяющей систематизировать данные и поддерживать их актуальность, обеспечивать гибкость управления ими и совместную работу сотрудников на всем предприятии.

Государственное унитарное предприятие Севастополя (ГУПС) «Водоканал», ранее имеющее информационную систему, разработанную в конце 90-х годов, отмечает ее моральное устаревание, которое ведет за собой замедление работы системы и ее последующее «зависание». В связи с этим предприятие нуждается в разработке новой ИС, решающей данную проблему, и, при этом, сохраняющей совместимость с уже готовой базой данных, использующей СУБД MySQL. Упрощенная концептуально-логическая схема базы данных представлена на рисунке 1.

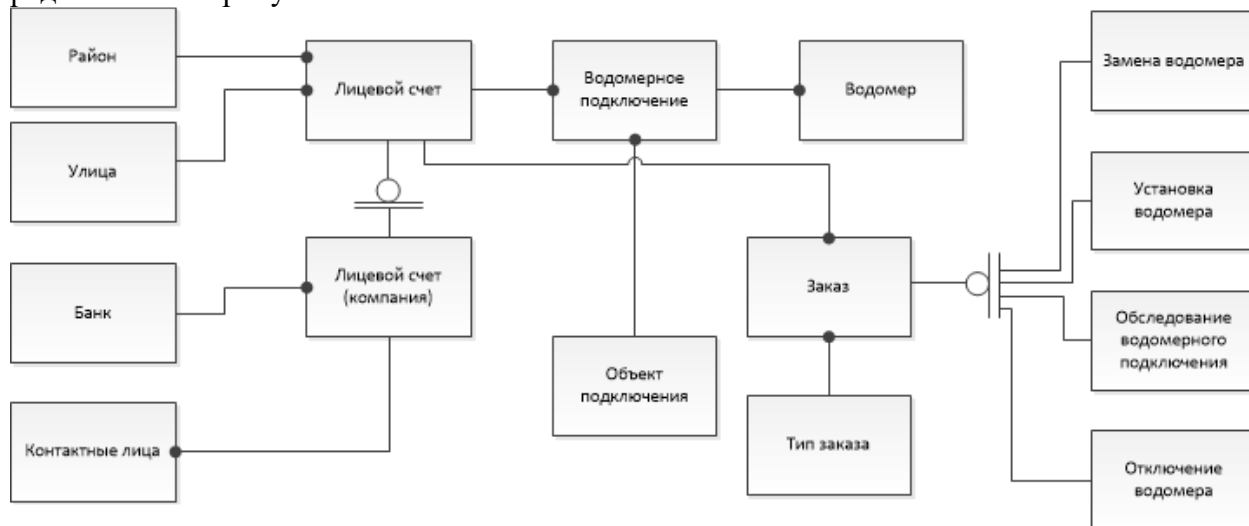


Рис.1. Упрощенная концептуально-логическая схема базы данных

В связи с тем, что база данных содержит в себе персональную информацию об абонентах, встает также вопрос о ее защите, используя современные средства: парольная защита на основе функций хеширования, шифрование дампов БД. Дамп базы данных содержит описание структуры базы и содержащихся в ней данных в виде команд SQL, используется для восстановления данных, в случае повреждения БД.

Задача построения ИС ГУПС «Водоканал» имеет свои особенности. Особенности заключаются в том, что ИС должна предоставлять информацию о водометрах – счетчиках воды, предназначенных для измерения и учёта объёма воды, проходящего по водопроводу в месте установки счётчика.

Основные задачи, решаемые ИС:

- Высокая скорость работы ИС, несмотря на большие объемы данных;
- Возможность запуска ИС на компьютерах, под управлением как Windows, так и Linux;
- Предоставление полной, достоверной и актуальной информации работникам предприятия в любой момент времени;
- Единое информационное пространство для управления и коммуникаций на предприятии;
- Обработка информации, ее защита от потерь и несанкционированного доступа;
- Минимизация потерь при реализации воды, за счет контроля расхода воды.

Так как ИС рассчитана на работу в компьютерных системах с различными ОС, то для обеспечения кроссплатформенности используется объектно-ориентированный высокоуровневый язык Java. Для взаимодействия с базами данных используется Java DataBase Connectivity (JDBC) — платформенно-независимый стандарт взаимодействия Java-приложений с различными СУБД. JDBC реализован в виде пакета `java.sql`, входящего в состав Java SE (Standard Edition). JDBC позволяет работать с СУБД без глубоких знаний специфики конкретной базы данных.

Разработанная ИС включает в себя: модуль инвентаризации, модуль защиты данных, модуль отчетности, базу данных. Структура ИС представлена на рисунке 2.

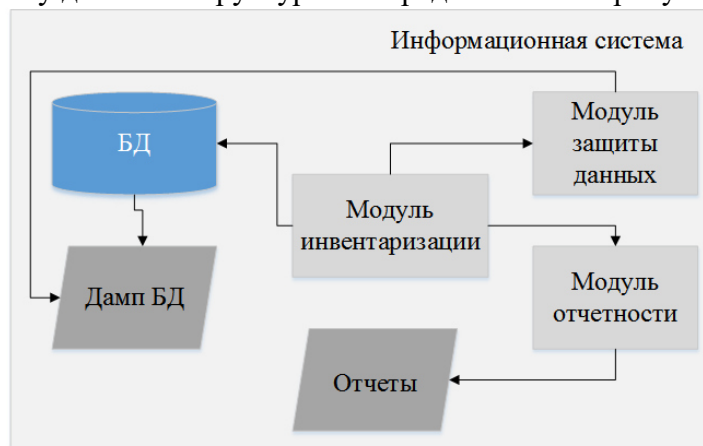


Рис. 2. Структура ИС

Модуль инвентаризации позволяет вести удаленную инвентаризацию приборов учета воды (водомеров). Фрагмент интерфейса модуля инвентаризации приведен на рисунке 3.

Модуль реализует следующие функции ИС:

- Внесение нового абонента в базу данных с последующим присваиванием ему лицевого счета;
- Изменение информации о лицевом счете абонента;
- Добавление нового водомера;
- Добавление нового водомерного подключения;
- Просмотр информации о
 - лицевых счетах физических и юридических лиц;
 - водомерах и водомерных подключениях, открытых на лицевом счете;
 - долге абонента или суммарном долге по найденным лицевым счетам;
 - заказах, связанных с обслуживанием водомеров и водомерных подключений;
 - контактных лицах лицевого счета (для юридических лиц);
- Оформление заказов на выполнение работ по замене, установке, снятию и обследованию водомеров;
- Изменение статуса оформленных заказов.

Лицевой счет

Регион: СЕВАСТОПОЛЬ Район: ГАГАРИНСКИЙ

Фамилия, имя, отчество: ПРЯНИКОВ ВИКТОР Маш.код: 524185 Лиц.счет: 126782 Сальдо: -900.78

Последний счет: ПЕТРОВИЧ

Дата договора: 28.07.2016 №: 13 Дата выписки: 11.11.2011

Адрес: Ул. ЮМАШЕВА Дом: 16 Корп.: 70

Владелец: ПРЯНИКОВ ВИКТОР ПЕТРОВИЧ

Категория потребителей: НАСЕЛЕНИЕ Почтовый индекс: 299065 Телефон: +79789037637

Статус лицевого счета: ЗАКРЫТ

Банк-плательщика: ОКПО: 6782499 МФО: 16437484

РНКБ: 96784632 Расчетный счет: 3567876 Код финанс.: 1252366 № свид.: 14547832

Машинный код	Долг (руб)	Л/счет	Фамилия	Имя
244521	0.0	439045	СЕМЕНОВ	СЕМЕН
979742	0.0	987976	ПЕРЕКЛЮЧА	НИКИТА
762780	-1098.0	128469	ШУЛЬГА	ОЛЬГА
678178	-0.9	904371	АЛЕКСЕЕВА	ЛЮДМИЛА
524185	-900.78	126782	ПРЯНИКОВ	ВИКТОР

Новый водомерный узел

*Код ввода *Коренной *Код ВП *Местонахождение *Состояние

*Принадлежность водомера *Объект *Вид подключения

*Принадлежность территории *Категория потребителей

*Подача Контактные телефоны

*Регион *Район

*Адрес *Дом Корпус Квартира

*№ ТУ *Дата ТУ *Глубина (н) Примечание

Очистить форму ОК Отмена

* - поля, обязательные к заполнению

а)

б)

Рис.3. Элементы интерфейса модуля инвентаризации

(а – просмотр информации о лицевых счетах абонентов; б – добавление нового водомерного узла)

Модуль отчетности включает в себя формирование отчетных документов в формате PDF, а также формирование шаблонов документов (например, образец извещения абонента о долге).

Модуль защиты обеспечивает парольную защиту и защиту дампа базы данных (файлы *.sql). При реализации парольной защиты вход в систему, запуск приложения, запрос на доступ к данным сопровождается запросом пароля и последующим сравнением введенного пароля с оригиналом (рисунок 4). В качестве метода хеширования пароля используется хэш-функция MD5 – односторонняя хэш-функция, по результатам преобразования которой, невозможно восстановить исходную информацию. Хэш оригинала хранится в базе данных, и, во время входа пользователем в систему, сверяется с полученным хэшем от пароля, который ввел пользователь. В случае успеха, пользователь проходит аутентификацию и получает доступ к системе.

После выполнения идентификации и аутентификации модуль защиты устанавливает полномочия (совокупность прав) работника для последующего контроля санкционированного использования объектов информационной системы.

Инвентаризация водомеров

Лицевой счет Справочники Банк Отчеты

Имя пользователя: admin

Пароль:

Войти Выход

Рис.4. Окно идентификации пользователя

Поскольку ИС используется в государственном предприятии, выбор шифрующего алгоритма должен соответствовать ГОСТ. В ИС защита дампа базы данных производится с помощью симметричного алгоритма блочного шифрования «Магма», который утверждён в качестве стандарта «ГОСТ 28147-89 Системы обработки информации. Защита криптографическая. Алгоритм криптографического преобразования». В основе шифра лежит сеть Фейстеля. Сеть состоит из ячеек, называемых ячейками Фейстеля. На вход каждой ячейки поступают данные и ключ. На выходе каждой ячейки получают изменённые данные и изменённый ключ.

«Магма» шифрует блоки данных, размером 64 бита, с использованием 256-битного ключа за 32 раунда. Общая структура алгоритма представлена на рисунке 5 (а). Каждый раунд

выполняет следующие операции: 1) сложение с раундовым ключом K_i по модулю 2^{32} ; 2) каждый 4-битный подвектор 32-битного входного вектора проходит через S-блок S_i (блок подстановки); 3) полученный 32-битный вектор циклически сдвигается влево на 11 бит. Структура раунда представлена на рисунке 5 (б).

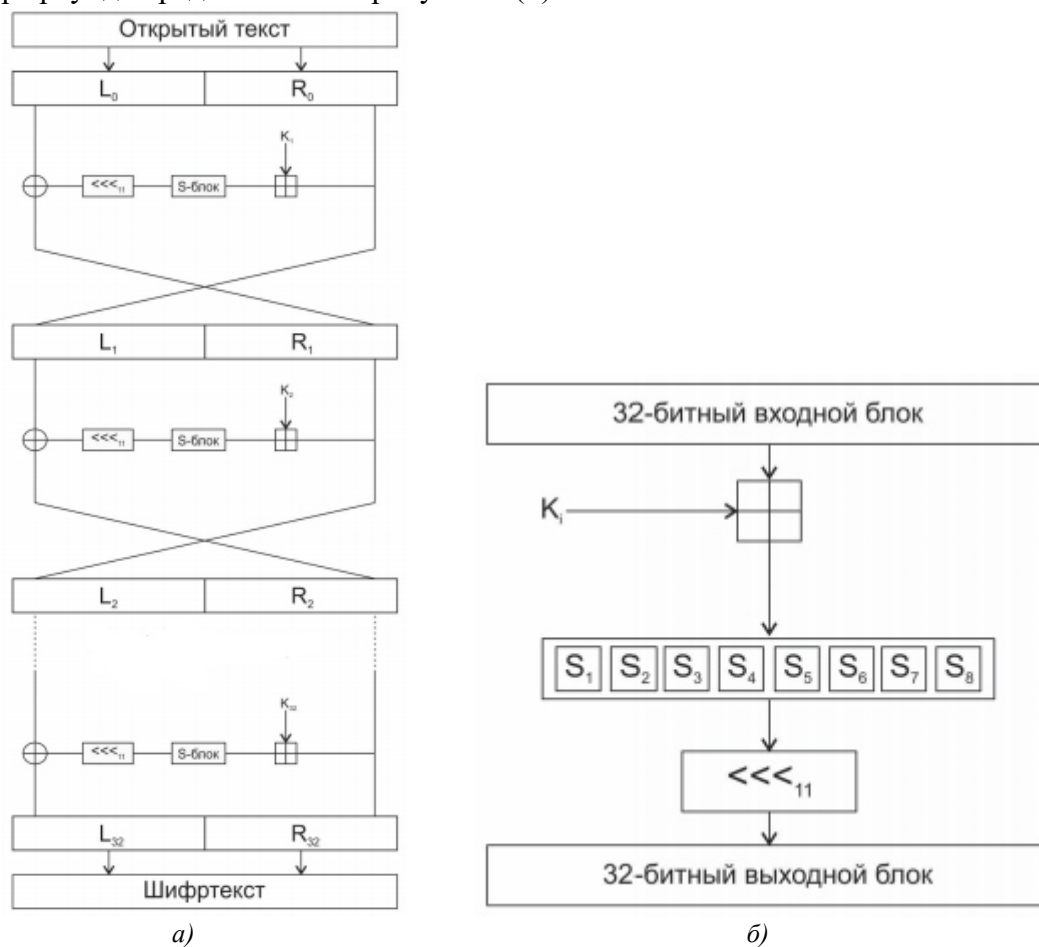


Рис. 5. Шифр «Магма»

(а – общая структура алгоритма; б – структура раунда)

Несмотря на то, что этот шифр разработан в 70-х годах прошлого века, он до сих пор является ГОСТом и для него пока не опубликовано криптоанализа, существенно снижающего стойкость.

В отличие от старой системы ГУПС «Водоканал», разработанная ИС имеет более высокую скорость работы, что исключает ее «зависание», а также позволяет работать с уже сформированной БД на платформе MySQL. Таким образом, у работников не возникнет никаких трудностей в освоении новой системы, поскольку ее интерфейс интуитивно прост и понятен, а администратор всего за несколько шагов подключит БД предприятия к разработанной ИС.

Библиографический список

1. Шнайер Б. Прикладная криптография/ Б. Шнайер.—СПб.:Питер, 2003. — 368 с.
2. Проектирование информационных систем. Учебное пособие для студентов / Сост. А. В. Бычков Кубан. гос. технол. ун-т. Каф. ВТ и АСУ. - Краснодар: Изд-во ГОУВПО «КубГТУ», 2008. -82 с.
3. Википедия. Свободная Энциклопедия. MD5 [Электронный ресурс].Режим доступа - <https://ru.wikipedia.org/wiki/MD5>.
4. Студопедия. Информационные системы. [Электронный ресурс]. Режим доступа - http://studopedia.ru/18_61568_informatsionnie-tehnologii-it.html

УДК 681.3

А.Ю. Поддубный, магистрант**Научный руководитель: Шумейко И.П., канд. физ-мат. наук, доц.***Севастопольский государственный университет***РАЗРАБОТКА ТЕХНОЛОГИИ ДЛЯ ФОРМИРОВАНИЯ НАУЧНОЙ ОТЧЕТНОСТИ ИЗ РАЗРОЗНЕННЫХ ИСТОЧНИКОВ СБОРА ДАННЫХ***Аннотация: В настоящей работе представлено описание системы формирования отчетной документации из разрозненных источников сбора данных при проведении археологических исследований.**Ключевые слова: археологические исследования, разрозненные источники данных, отчетная документация, мобильное устройство**Annotation: In this paper, the systems for the formation of reporting documents from disparate sources of data for archaeological research are presented.**Keywords: archaeological research, disparate data sources, reporting documentation*

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 информационных технологий, для реализации задач, признанных приоритетными [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. На данный момент на территории РФ не существует единой информационной системы археологических исследований, в связи с чем является актуальной тема создания таковой.

Цель статьи. Разработать комплексную модель информационного обеспечения историко-археологических комплексов РФ, методики формирования единой информационной базы и оптимизации потоков информации в единой информационной базе, учитывающей особенности информационных обеспечений существующих разнородных источников для проекта «Археолог-Херсонес» [3,4].

Основная часть. В рамках реализации поставленной цели исследования в работе разрешается противоречие, состоящее в том что в настоящее время информационное пространство представляет несвязанный набор информационных обеспечений систем автоматизированного сбора и обработки данных, что существенно снижает эффективность формирования научной отчетности, а также её качество и полноту. В основе задачи лежит извлечение информации из разрозненных текстовых источников данных, которые были получены в ходе проведения полевых археологических работ, исходя из тематики, заданной пользователем.

Для достижения поставленной цели необходимо выполнить следующие задачи:

- проанализировать информационную структуру археологических исследований[3];
- предложить структурную схему технологии формирования отчетной документации из разрозненных источников.

Актуальность создания единой информационной системы обусловлена целым рядом факторов, среди которых особо следует выделить информатизацию, возрастание роли информационного обеспечения при принятии управленческих решений, а также многофакторность и разнообразие проблем, требующих решения. Данная технология позволит не только объединить и синхронизировать технологические процессы сбора данных, получения отчетов о полученных результатах с их одновременным документированием в банках данных и хранилищах, а также использовать современные технологии в целях оперативного получения информации из корпоративных документированных хранилищ и ее аналитической обработки для предоставления лицам, принимающим решения. Ранее была проведена разработка мобильного приложения для автоматизации полевых археологических работ [4].

На рисунке 1 представлена схема комплексной информационной технологии.

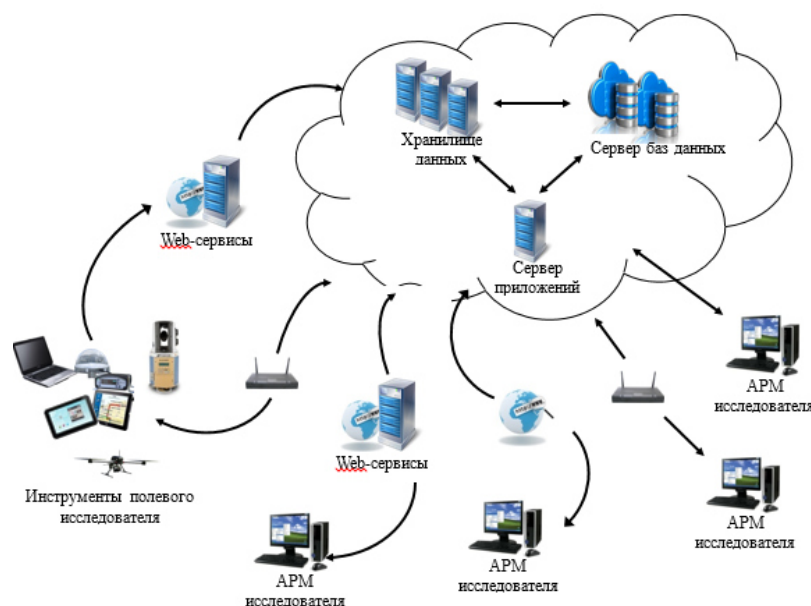


Рис. 1. Функциональная схема комплексной информационной технологии

Для описания модели была построена диаграмма в нотации DFD (рис.2), а также её декомпозиции (рис. 3,4).

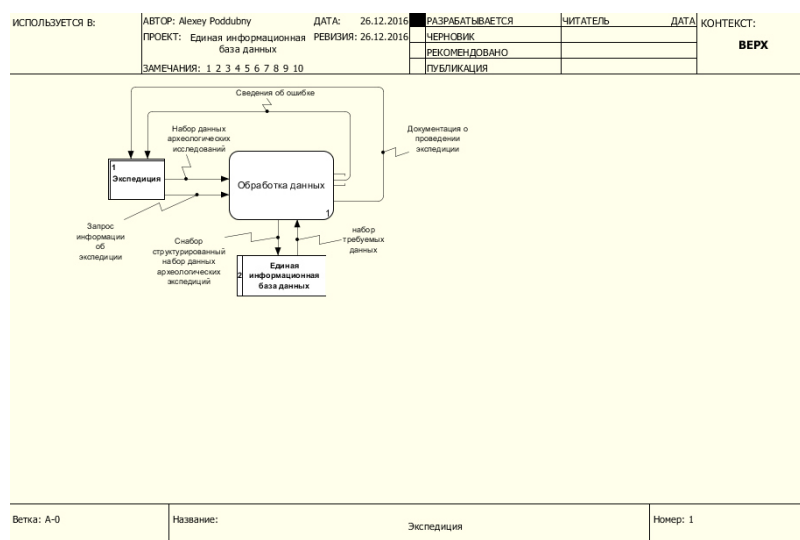


Рис. 2. Контекстная диаграмма в методологии DFD

Исходя из информации, представленной на рис. 2-4, основным процессом является процесс обработки данных, который обрабатывает данные археологических исследований, полученных в ходе экспедиции, которая представлена внешней сущностью. В качестве хранилища данных выступает единая информационная база данных археологических исследований.

Процесс обработки данных состоит из двух компонентов: импорт и экспорт. Процесс импорта являет собой совокупность процессов извлечения данных, проверки данных на содержание ошибок, и, в случае их отсутствия, производится группировка данных и

приведение их к соответствующему виду, после чего валидные наборы данных вносятся в единую информационную базу.

Процесс экспорта данных состоит из 3-х этапов: обработки запроса, извлечения данных из единой информационной базы и формирования отчета с необходимой пользователю информацией.

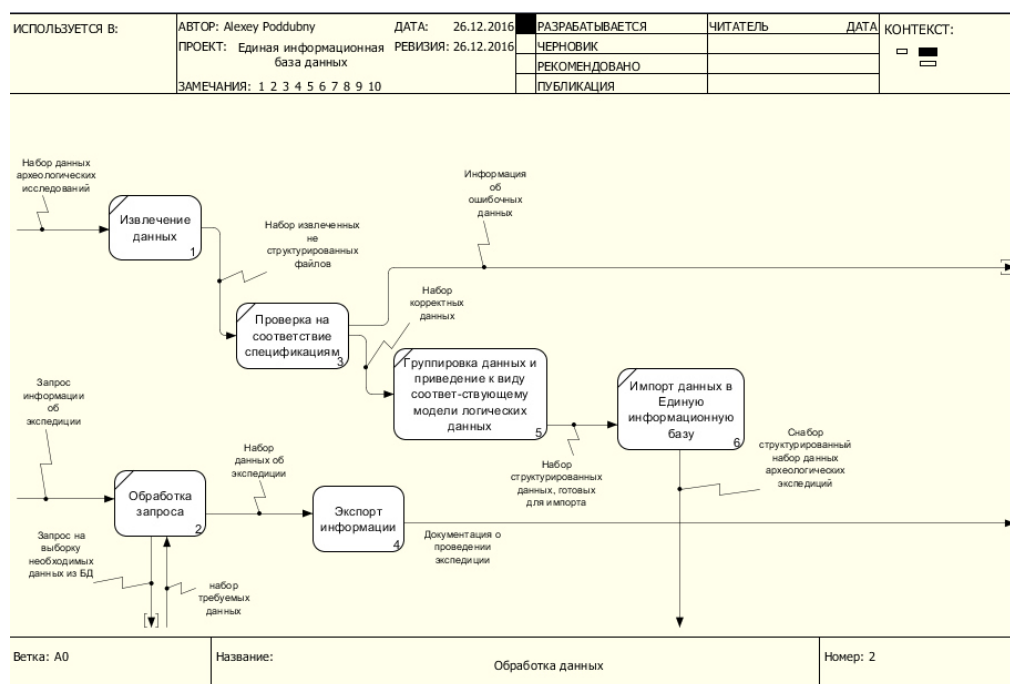


Рис. 3 Диаграмма декомпозиции контекстной диаграммы

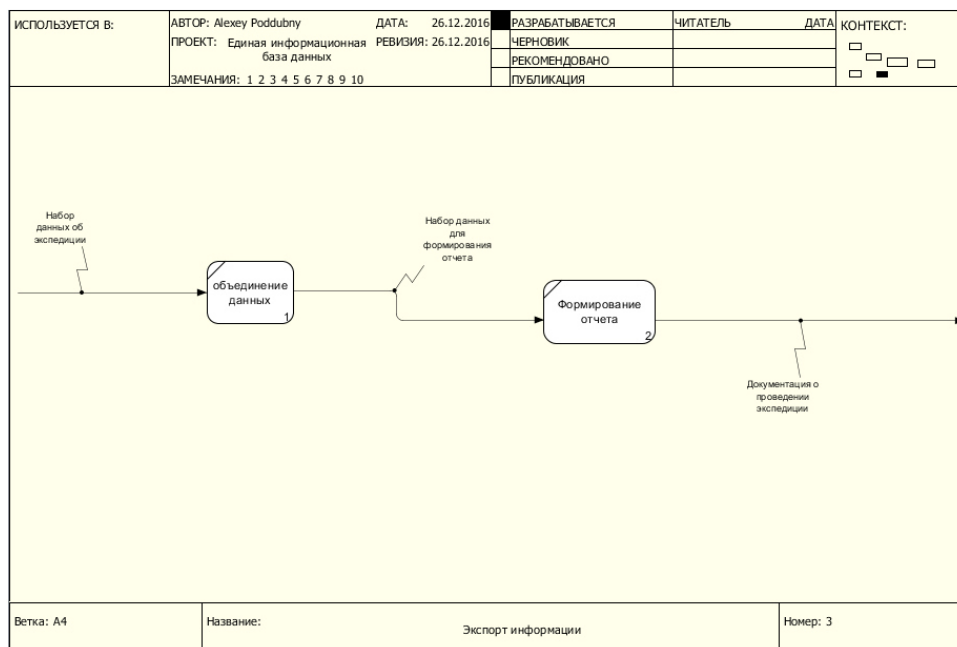


Рис. 4 Диаграмма декомпозиции блока «экспорт информации»

Таким образом, единая информационная система должна содержать в себе систему сбора данных, систему первичной обработки данных, локальное хранилище данных, а также систему вторичной обработки данных и глобальное хранилище данных. Вторичная система

обработки данных является тем самым связующим звеном между локальными системами сбора и обработки данных, которое позволит упростить и ускорить процесс археологических исследований.

Для описания бизнес-процессов, была применена технология IDEF0. На рис. 5-7 изображены IDEF0 диаграммы процессов археологических исследований. На декомпозиции первого уровня изображены основные процессы археологических исследований в рамках единой информационной системы. Основными этапами являются полевые археологические работы, обработка и изучение материала, формирование научной отчетной документации и публикация в единой информационной базе данных. Процесс публикации в свою очередь, состоит из этапа извлечения данных, разделения данных по функциональным характеристикам, объединения данных из различных источников и сохранения структурированного набора в единой информационной базе данных.

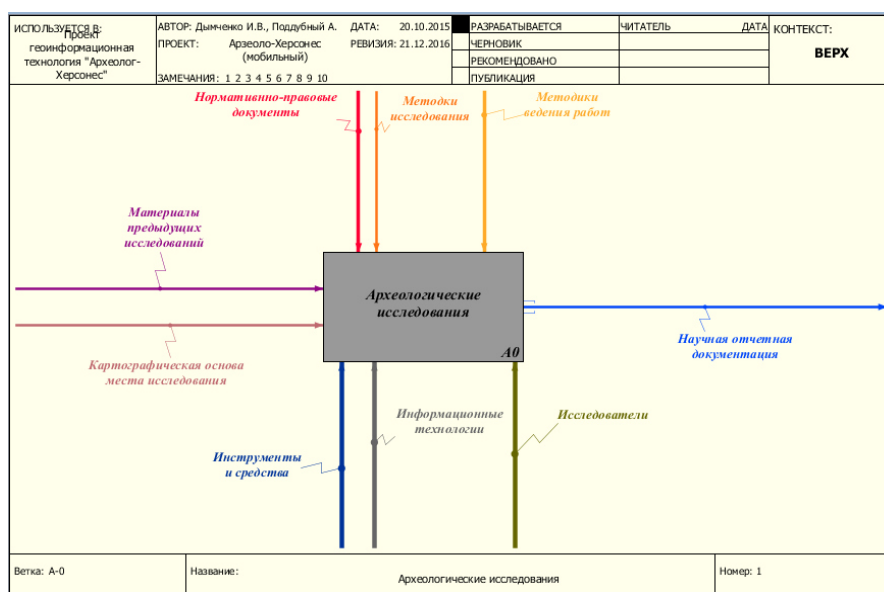


Рис. 5 Контекстная диаграмма в методологии IDEF0

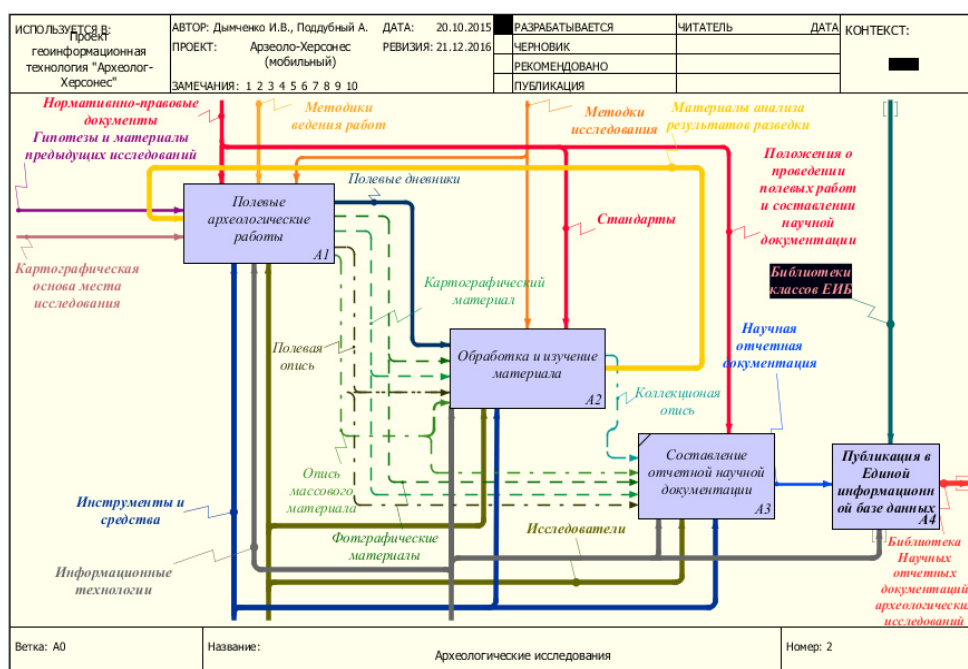


Рис. 6 – Диаграмма декомпозиции контекстной диаграммы

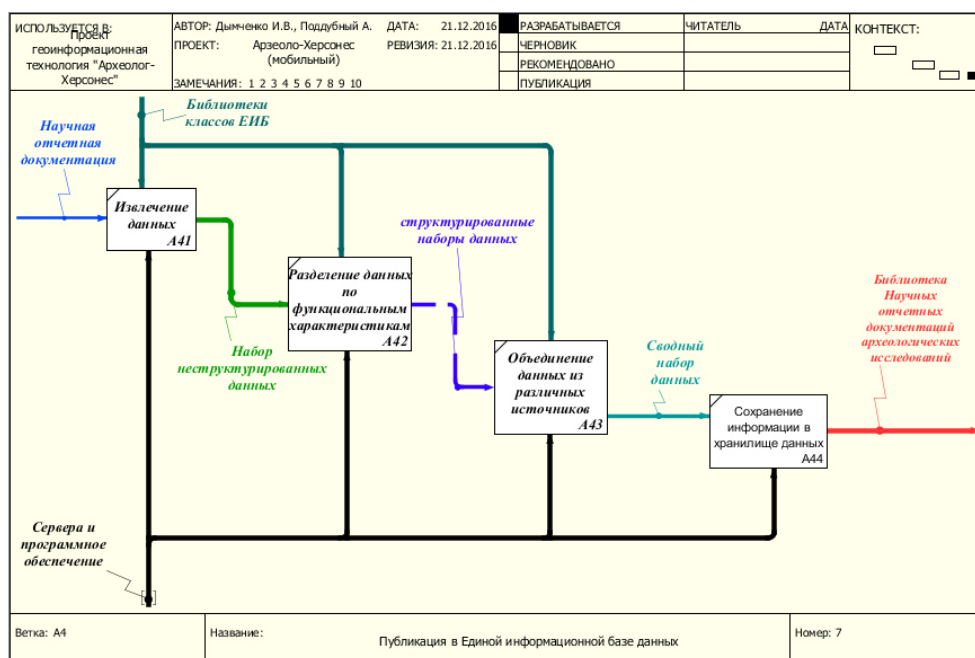


Рис. 7 – Диаграмма декомпозиции блока «Публикация в единой информационной базе археологических исследований»

Единая информационная база данных будет содержать в себе библиотеку научных отчетных документаций археологических исследований.

Выводы и дальнейшее направление исследований. В процессе работы над проектом были изучены процессы проведения археологических работ, обоснована необходимость привлечения современных информационных технологий для их автоматизации. Разработана модель информационной технологии для формирования отчетности из разрозненных источников сбора данных.

В дальнейшем планируется программно реализовать данный процесс формирования отчетности.

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/ © КонсультантПлюс, 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. Поддубный А.Ю. Разработка мобильного приложения для автоматизации полевых археологических работ. Мир компьютерных технологий: Сборник статей студенческой научно-технической конференции, г.Севастополь, 04 – 08 апреля 2016 г. – г. Севастополь: СевГУ, 2016. – 241 с., С.200-204.

УДК 004.7

С.С. Райков, А.О. Пармузин, магистранты 2 курса**Научный руководитель: В.С. Чернега, канд. техн. наук, доцент***Севастопольский государственный университет***ОРГАНИЗАЦИЯ БЕСПРОВОДНОГО КАНАЛА СВЯЗИ ДЛЯ УПРАВЛЕНИЯ МЕДИЦИНСКИМ ОБОРУДОВАНИЕМ**

Аннотация: описаны требования и особенности реализации микропроцессорного модуля, обеспечивающего съем и передачу по беспроводному каналу связи измерительной информации и командных сообщений с медицинской электронной аппаратуры.

Ключевые слова: медицинская информационная система, сопряжение с медицинским оборудованием, микроконтроллер.

Annotation: requirements and features of the microprocessor module of transmission through a wireless channel of measuring information and commands from a medical electronic equipment are described.

Key words: medical information system, conjugation to the medical equipment, microcontroller.

В настоящее время информационные технологии все шире находят применение в медицине. Одним из перспективных направлений является создание медицинских информационных систем для хирургических отделений [1,2]. Современные операционные медицинских заведений насыщены сложной электронной аппаратурой, которой управляют высший или средний медицинский персонал. Наличие вспомогательного персонала в операционной увеличивает затраты на операцию, а также ухудшает условия стерильности. В связи с этим актуальным является создание систем сбора информации с медицинского оборудования и отображения ее на центральном мониторе хирурга, а также передачи команд управления для изменения параметров функционирования аппаратуры.

Реализация такой системы осложняется значительным уровнем электромагнитных помех, создаваемых работающим электронным оборудованием, необходимостью обеспечения практически безошибочной передачи команд управления, выполнения гальванической развязки модулей сопряжения с электронным оборудованием.

Целью данной работы является создание универсального модуля съема информации и передачи измерительной информации с электронного медицинского оборудования по беспроводному каналу связи на центральный компьютер операционного зала.

В процессе достижения поставленной цели авторами были изучены интерфейсы и протоколы связи электронного медицинского оборудования, проведена оценка помеховой обстановки в операционной, осуществлен выбор помехоустойчивого кодирования и шифрования передаваемых сообщений, выполнен выбор программно-аппаратной реализации модуля съема и передачи данных.

Установлено, что основными внешними интерфейсами медицинского оборудования являются стандартные интерфейсы RS-232C, RS-485, Ethernet (соединитель типа RJ-45) [3-5]. Обмен информацией осуществляется по стандартным протоколам Modbus и Profibus.

Исходя из этих предпосылок, модуль съема и передачи информации должен быть оснащен соответствующими разъемами, позволяющими подключать модуль к электронной аппаратуре, расположенной в операционном зале. В связи с тем, что в интерфейсах RS-232C и RS-485 используются разнополярные сигналы, необходимо на стыке модуля съема информации предусмотреть преобразователь уровней униполярных сигналов в биполярные и наоборот. Для обеспечения гальванической развязки модуля с медицинским оборудованием следует применить оптоэлектронные преобразователи.

Основой модуля сопряжения является 32-разрядный программируемый микроконтроллер типа STM32. При выборе элементной базы для реализации модуля использовался критерий минимизации стоимости при равной производительности и функциональности. На основании этого критерия был выбран модуль STM32 Nucleo F401RE – отладчик, программатор и микроконтроллер со всей необходимой «обвязкой». Под «обвязкой» следует понимать вспомогательные электронные компоненты, такие как сглаживающие конденсаторы в цепях питания, кварцевые резонаторы (генераторы) для

реализации высокоточного тактирования, стабилизаторы напряжения и прочие элементы, необходимые для поддержания работы микроконтроллера. Для сопряжения униполярных цепей с биполярными используется микросхема MAX232. Преимуществом такого элемента является возможность обеспечить биполярные сигналы при наличии одного однополярного источника питания. Для гальванической развязки выбраны оптроны типа PC-845, содержащие по четыре оптронные пары в одном DIP-корпусе.

При выборе Wi-Fi модуля использовались такие же обобщенные критерии минимизации стоимости при равной производительности и функциональности, как и при выборе микроконтроллера. Сравнение Wi-Fi модулей различных производителей по критериям объема ОЗУ, ПЗУ, максимальной скорости передачи данных и степени поддержки интерфейсов приведено в таблице 1.

Таблица 1 – Наиболее популярные модели Wi-Fi модулей

Производитель	Название модуля	Поддерживаемые интерфейсы	Объем ОЗУ	Объем ПЗУ	Макс. скорость передачи
ST Microelectronics	SPWF01SA.11	UART, I2C, SPI, JTAG	64 кб	1.5 Мб	54 Мбит/с
Ai-Thinker	ESP8266	SPI, UART, I2C, I2S	80 кб	8 Мб	54 Мбит/с
MXCHIP	EMW3162	UART, SPI, I ² C, ADC, USB, CAN	128 кб	1 Мб	150 Мбит/с

Для решения поставленной задачи был выбран модуль ESP8266, так как размер его оперативной памяти позволяет использовать стороннее встроенное программное обеспечение с более богатым функционалом, по сравнению со стандартным. Данный модуль при этом обладает относительно невысокой стоимостью. Существуют и другие производители Wi-Fi модулей, но в основном они сконструированы на базе чипов компании Espressif или ST Microelectronics и потому не имеют существенных отличий в характеристиках.

При тестировании выбранного модуля оказалось, что в режиме «station» (станция) часто происходит разрыв TCP-соединения и после программного сброса модуля его не удается восстановить. В ходе поиска решения было определено, что это является следствием программной ошибки самого модуля. Также проблемой являлась обработка возвращаемых модулем данных, которые представляют собой результаты выполнения команд. Результаты выполнения команд представляют собой строки переменной длины на естественном языке, что затрудняет единообразную обработку состояния устройства. Поэтому наиболее предпочтительным являлось бы кодирование результатов команд. Так как для строки, с целью последующей обработки, нужна дополнительная память (буфер), в отличие от кода состояния, который может занимать память одного регистра. Для решения вышеперечисленных проблем было принято решение сменить встроенную микропрограмму.

Таким образом, был разработан прототип системы, представленный в виде функциональной схемы на рисунке 1.

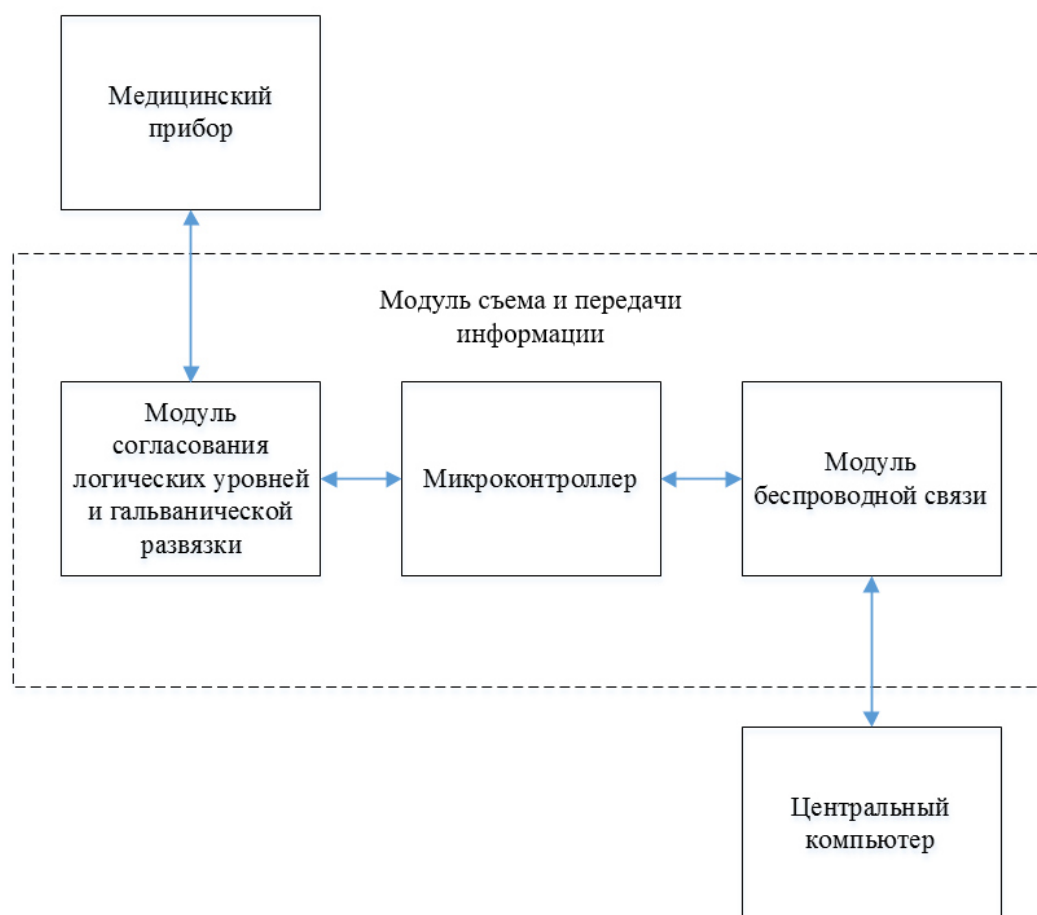


Рис. 1 – Функциональная схема разрабатываемой системы

Модуль согласования логических уровней и гальванической развязки предназначен для обеспечения совместимости интерфейса медицинского прибора и микроконтроллера, а также повышения помехозащищенности и точности передачи сигнала. Микроконтроллер предназначен для сбора и обработки передаваемых (получаемых) данных, а также управления работой модуля беспроводной связи. Стрелками на функциональной схеме обозначены потоки данных.

Так как беспроводной канал связи является общедоступным, в дальнейшем планируется исследовать способы защиты передаваемой информации от несанкционированного доступа.

Библиографический список

1. Чернега В.С. Проблемы создания и внедрения информационных технологий в хирургических операционных // Перспективные направления развития отечественных информационных технологий: материалы II межрегиональной научно-практической конференции. Севастополь, 13-17 сентября 2016 г. – Севастополь: СевГУ, 2016. – С. 97-98.
2. 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.
3. Олифер В. Г., Олифер Н.А., Компьютерные сети. Принципы, технологии, протоколы, Учебник для вузов 3-е изд. СПб.:Питер, 2006, 958 с.
4. Григорьев В.А, Лагутенко О.И., Распаев Ю.А., Сети и системы радиодоступа, М.: Издательский дом «Вильямс», 2003, 640 с.
5. Рошан П., Лизри Дж, Основы построения беспроводных локальных сетей стандарта 802.11, М.: Издальский дом «Вильямс», 2004, 304 с.

УДК 004.65

А.И. Салидинов, магистрант**Научный руководитель: Ю.Г. Шаталова, к.т.н., доцент***Севастопольский государственный университет***РАЗРАБОТКА СИСТЕМЫ РЕПЛИКАЦИИ ДЛЯ РАСПРЕДЕЛЕННОЙ БАЗЫ ДАННЫХ АВТОМАГАЗИНА**

Аннотация: в данной статье представлен анализ и классификация методов репликации, а также определены свойства распределенной базы данных системы, проектируемой в рамках выполнения магистерской работы. На основании проведенных исследований определены параметры репликации, которые будут использованы в проектируемой распределенной системе.

Ключевые слова: распределенные базы данных, механизмы репликации, функции репликации, удаленный узел, центральный узел, синхронная репликация.

Annotation: the efficiency of using distributed databases has long been recognized, but the relevant technologies are either too limited or complex. This article presents analysis and classification of replication methods, as well as the properties of the distributed database of the system designed in the framework of the master's work. Based on the research, the replication parameters that will be used in the projected distributed system are determined.

Key words: distributed database, replication mechanisms, replication functions, remote host, central node, synchronous replication.

В настоящее время сфера информационных технологий характеризуется быстрым развитием методов обработки данных. Структуры современных информационных систем представляют собой разветвленную схему, в состав которой входят несколько разделенных узлов, которые функционируют как единое целое.

Одним из элементов таких систем являются распределенные базы данных (РБД), основной особенностью функционирования которых является то, что со стороны пользователя не должно быть разницы, где и каким образом расположены необходимые ему данные [1]. Данная особенность таких систем достигается за счет использования механизма репликации данных, т.е. копирования данных, который способствует поддержанию данных в актуальном состоянии на всех узлах базы данных. Использование данного механизма является необходимым. Во-первых, для обеспечения резервирования данных на случай непредвиденных сбоев или отказов в системе. Во-вторых, применение механизма репликации способствует объединению информации при создании автоматизированной системы. В-третьих, использование репликации обеспечивает работоспособность тех узлов распределенной системы, которые не могут поддерживать постоянное соединение с центральным сервером базы данных. Это, в свою очередь, позволяет обеспечить автономную работу узла системы. В силу сказанного, разработка системы репликации для распределенной базы данных конкретного предприятия является актуальной задачей.

В статье предлагается система репликации, разработанная для РБД автомагазина. Для построения эффективной системы репликации распределенной базы данных необходимо провести исследование существующих методов репликации и проанализировать их особенности с точки зрения возможности реализации именно для данной РБД. Исследование требований, предъявляемых к разрабатываемой системе репликации, механизмов репликации распределенных баз данных и выбор оптимальных методов для их практической реализации является целью данной статьи.

Как и любой другой механизм, механизм репликации данных несет в себе следующие функции: а) масштабируемость: эффективная обработка как малых, так и больших наборов данных; б) отображение и трансформация: поддержка репликации в гетерогенных системах, использующих несколько платформ; в) репликация объектов: возможность репликации объектов, отличных от обычных данных; г) средства определения схемы репликации: предоставление возможности администратору базы данных задавать данные и объекты, подлежащие репликации.

В ходе исследования была определена следующая классификация методов репликации [2]:

1. По направлению репликации: а) однонаправленная: изменения происходят только в одной базе данных (БД); б) мультинаправленная: изменения происходят во всей БД.

2. По выполнению сеанса репликации: а) репликация реального времени: синхронизация происходит сразу после внесенных изменений; б) отложенная репликация: процесс репликации запускается в запланированное время или же при наступлении определенного события.

3. По способу передачи информации: а) прямая передача: соединение серверов БД, при помощи клиентского программного обеспечения, для прямого изменения и анализа реплицируемых данных; б) вероятностная передача: при такой передаче канал неустойчив и не гарантирует устойчивую связь без обрывов, поэтому данные передаются цельными пакетами.

4. По типу репликации: а) синхронная: транзакции не фиксируются до тех пор, пока на всех узлах с копиями не будут внесены изменения; б) асинхронная: изменения в копии данных вносятся независимо.

5. По способу анализа реплицируемой информации: а) репликация по текущему состоянию: сравнение записей одной таблицы с записями другой, и на основании этого принимается решение о синхронизации; б) дельта репликация: в БД предусмотрен журнал изменений, и алгоритм репликации переносит изменения по дельтам изменений согласно журнала.

6. По типу тиражирования данных: а) базовое тиражирование: копии обеспечивают доступ типа «только для чтения»; б) усовершенствованное тиражирование: считывание и обновление всех копий таблиц во всей системе.

Различные режимы репликации предъявляют неодинаковые требования к распределенной системе, поэтому вопрос выбора этих параметров в каждом конкретном случае является актуальным. Также в ходе проводимого исследования были определены основные свойства распределенной базы данных автомагазина. Такие как:

- низкое время обработки запроса;
- высокая степень актуальности данных;
- возможность работа удаленного узла без постоянного соединения с базой;
- уменьшение нагрузки на сервер базы данных;
- уменьшение сетевого трафика.

Были проанализированы также требования к скорости внесения изменений на различных узлах РБД и на центральном узле, и требования к затратам ресурсов, связанным с реализацией репликации. Основываясь на проведенных исследованиях, были выбраны следующие параметры механизма репликации. Для разрабатываемой РБД целесообразно использовать мультинеправленную синхронную репликацию, которая позволит добиться оптимальной по скорости и затратам ресурсов репликации данных во всех узлах распределенной сети. Так как при работе системы предусматриваются частые запросы на чтение данных с центрального узла, то на удаленных узлах разумно использовать репликацию реального времени, чтобы обеспечить незамедлительную синхронизацию, после внесения каких-либо изменений.

Выводы. В работе было проведено исследование механизма репликации, приведена классификация методов репликации. Проанализированы особенности имеющихся подходов к репликации РБД и требования к системе репликации базы данных автомагазина. На основании полученных сведений были определены параметры репликации, которые в свою очередь легли в основу практической реализации распределенной системы, разрабатываемой в рамках квалификационной работы магистра.

Библиографический список

1. К. Дж. Дейт Введение в системы баз данных, 8-е издание.: Пер. с англ. — М.: Издательский дом «Вильямс», 2005. — 1328 с.
2. Рябков Н.С. Аналитический анализ обзор методов репликации и синхронизации баз данных / Н.С. Рябков // Информационные технологии в менеджмент качество и инновационном менеджменте. – 2006. - № 6. - С. 56-63

УДК 519.21

Л.В. Флора, студентка 4-го курса**Научный руководитель: М.В. Заморёнов, канд. техн. наук, доц.***Севастопольский государственный университет***ПРОГРАММНО-АППАРАТНЫЙ КОМПЛЕКС СИМУЛЯТОРА СПОРТИВНОЙ СТРЕЛБЫ**

Аннотация: в данной статье рассмотрено создание программно-аппаратного комплекса симулятора спортивной стрельбы на базе микроконтроллера и ПО для реализации математической модели полета снаряда. Проанализированы доступные методы реализации платформы для сбора исходных данных, их передачи и обработки. Выявлена и обоснована целесообразность применения выбранных аппаратных и программных средств. Кроме того, рассмотрены перспективные направления развития проекта, как в сторону удобства использования, так и повышения качества симуляции.

Ключевые слова: спортивная стрельба, симулятор, микроконтроллер, датчики, математическое моделирование

Annotation: this article describes the development of software and hardware simulator of a sports shooting based on microcontroller and software implementing the mathematical model of the projectile fired. Author analyzed available methods of implementation of the platform for data acquisition, transmission and processing. Revealed and substantiated the feasibility of the selected hardware and software. Also prospective directions of development of the project in the direction of ease of use and improve the quality of the simulation were considered.

Key words: sports shooting, simulator, microcontroller, sensors, mathematical modelling

Отдел молодежной политики и спорта Севастопольского государственного университета в рамках реализации плана культурно-массовой работы много внимания уделяет организации досуга и спортивной подготовки студентов нашего университета. Для выявления и, как следствие, приоритетного развития, наиболее востребованных спортивных направлений в октябре 2016 года был проведен опрос среди студентов 1-5 курсов всех направлений подготовки. Кроме классических футбола (24%), волейбола (23%) и различных единоборств (20%), значительный интерес (20%) вызвали стрелковые виды спорта – пулевая стрельба и стрельба из лука. Судя по всему, нестабильная геополитическая обстановка так или иначе влияет на интересы нашей молодежи. Таким образом, целью данной работы стала разработка программно-аппаратного симулятора тренажера для проведения занятий по стрелковой подготовке.

Современный уровень развития техники позволяет создать удобное и достаточно автономное устройство, позволяющее выполнять упражнения с максимальной реалистичностью. Исходя из коммерческой доступности в данном проекте используются:

- Для получения исходных первичных данных - ультразвуковые датчики расстояния типа сонар и трехосевой гироскоп-акселерометр;
- Для приема данных и их первичной обработки - автономный микроконтроллер на базе ATmega328;
- Для моделирования полета снаряда, настройки параметров и отображения результатов – кроссплатформенное программное обеспечение, разработанное на языке Java в среде разработки приложений NetBeans.

Важным вопросом является разработка математической модели, дающей возможности имитационного моделирования для дальнейшего изучения и анализа траекторий полета снаряда. Для этого было проведено изучение физических основ баллистики снаряда пневматического оружия с учетом факторов.

Гироскоп представляет собой устройство, которое реагирует на изменение углов ориентации контролируемого тела. В классическом представлении это некий инерционный предмет, который быстро вращается на подвесах. Как результат, вращающийся предмет всегда будет сохранять свое направление, а по положению подвесов можно определить угол отклонения. Электронные же гироскопы построены по иной схеме и устроены сложнее. Акселерометр - это устройство, которое измеряет проекцию кажущегося ускорения,

то есть разницы между истинным ускорением объекта и гравитационным ускорением. Гироскоп реагирует на изменение в пространстве независимо от направления движения, с помощью акселерометра может измерять линейные ускорения предмета, а также искусственно рассчитываемое расположение предмета в пространстве.[2]

Характеристики MPU6050:

- напряжения питания 2,375 - 3,46 вольт
- потребляемый ток до 4 мА
- интерфейс передачи данных - I2C
- максимальная скорость I2C - 400 кГц
- вход для других датчиков I2C
- внутренний генератор на 8 МГц (вне модуля возможность подключить внешний кварцевый резонатор на 32,768 кГц или 19,2 МГц)

Функции MPU6050:

- трех осевой MEMS гироскоп с 16 битным АЦП
- трех осевой MEMS акселерометр с 16 битным АЦП
- Digital Motion Processor (DMP)
- FIFO
- прерывания
- температурный сенсор
- самопроверка гироскопа и акселерометра
- регистр идентификации устройства

Ультразвуковой дальномер HC-SR04

Принцип его работы основан на испускании ультразвука и его отражения от впереди находящихся предметов. Исходя из времени возвращения звука, по простой формуле, можно рассчитать расстояние до объекта.

Основные технические характеристики:

- Напряжение питания: 5В
- Ток покоя: <2мА
- Эффективный угол: не более 15°
- Диапазон измерения расстояния: 2–450см
- Разрешение: 0.3см [1]

На текущем этапе будет реализовано полуавтономное устройство целеуказания с упрощенным программным обеспечением, позволяющим выполнять стрельбу лишь по статичной мишени.

Модульность структуры позволяет вести дальнейшую разработку системы с целью повышения реалистичности математической модели, а также удобства пользования. Структура системы приведена на рис.1.

Перспективными целями развития системы являются:

- автономность устройства целеуказания: внутренний источник питания, передача данных по радиоканалу;
- усложнение математической модели полета снаряда [3] с учетом сложных внешних условий и специфики стрельбы из других видов оружия;
- улучшение графического интерфейса программного обеспечения;
- создание управляющего устройства для рабочего места оператора-тренера стрельбы.



Рис. 1 – Структура системы

Функциональность, гибкость и широкая доступность Arduino-совместимых плат и разнообразие периферийных устройств делают тренажер чрезвычайно эффективным средством обучения начальным навыкам стрельбы.

Очевидно, что виртуальная версия тренажера не может стать заменой реальной тренировки в полигонных условиях. Несмотря на это, она позволяет студенту получить первичные навыки, подготовить его теоретически и практически. Перспективной целью является доработка системы для более опытных стрелков, которая предоставит возможность стрельбы в сложных условиях (ветер, осадки, ночь).

Библиографический список

- 1) Ультразвуковой дальномер HC-SR04 // Интернет-магазин Zlectro [сайт] URL: <http://zelectro.cc/HC-SR04> (дата обращения 21.03.2017)
- 2) Акселерометр и гироскоп MPU6050 // Сайт Паяльник [сайт] URL: <http://schem.net/mc/mc324.php> (дата обращения 21.03.2017)
- 3) Сыпченко Р.П., Боев В.Д. Компьютерное моделирование. Элементы теории и практики: Учеб. пособие СПб.: ВАС, 2009

СЕКЦИЯ 10

Геоинформационные системы и технологии



УДК 681.3

Е.А.Гальшина, студент 4-го курса

Научный руководитель: И.В.Дымченко, ст. преподаватель.

Севастопольский государственный университет

ПРИМЕНЕНИЕ ГЕОИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ ПРИ РЕШЕНИИ ПРОБЛЕМ ОРГАНИЗАЦИИ СБОРА, ХРАНЕНИЯ И ПРЕДСТАВЛЕНИЯ ДАННЫХ АРХЕОЛОГИЧЕСКИХ ИССЛЕДОВАНИЙ

Аннотация: В данной статье рассматриваются особенности проведения археологических экспедиций, проблемы сбора археологических данных, централизованного хранения полученной информации и составления отчётов по итогам исследований. Предлагается решение рассмотренной проблемы в виде общедоступного хранилища данных, где загружаемые данные будут структурироваться для последующего упрощения составления отчёта по результатам экспедиции.

Ключевые слова: археология, экспедиция, отчёт, геоинформационная система, хранилище данных, веб-интерфейс.

Annotation: This article reviews features of archaeological expeditions, problems of archaeological data collecting, centralized storage of the information received and reporting on the results of studies. The decision of reviewed problem is proposed as public data warehouse, where downloaded data would be structured for simplification of reporting on the results of the expedition.

Key words: archaeology, expedition, report, geographic information system, data warehouse, web interface.

Под полевыми археологическими исследованиями понимают совокупность научных методов, направленных на раскопку, изучение и сохранение памятников, представляющих собой историческую ценность и находящихся под угрозой исчезновения. Это длительный, сложно организованный процесс, проходящий во много этапов. Право организовывать и проводить полевые археологические исследования, согласно законодательству и инструкциям, имеют только специально обученные эксперты, имеющие соответствующее образование, квалификацию, а также достаточный опыт работы в области археологии, поскольку раскопки и изучение памятников древности – это процесс, требующий предельной аккуратности, знания дела и наличия опыта.

Полевые исследования играют огромную роль в пополнении фонда археологических источников, представляющего собой основу для последующих теоретических реконструкций. Во время раскопок обнаруживаются и исследуются различные материальные остатки и связанные с ними культурные напластования. Исследования включают в себя такие этапы, как выявление (разведка), описание и непосредственное изучение памятников. Такие памятники, как курганы, городища, древние сооружения и некоторые другие, можно выявить по внешним признакам, другие же – по скоплению находок на сравнительно небольшой площади или обнажениям культурного слоя. В отдельных случаях при изучении раскопок используют специальные приборы, которые обнаруживают в земле следы деятельности человека по наличию показательных химических элементов или по плотности искусственных сооружений, что значительно отличается от окружающих напластований (георадары, электрометрия), аэрофотосъёмку и др.

Основной способ изучения памятников археологии – раскопки. К наиболее важным методам, которые применяются в их ходе, относится стратиграфический, который состоит в выявлении напластований, образованных в результате человеческой деятельности (культурный слой), наблюдении за их последовательностью и взаимосвязью. При проведении полевых исследований эксперты стремятся получить археологические комплексы, т.е. скопления, или группы, одновременных материалов на сравнительно недалёком расстоянии друг от друга. Находки одного стратиграфического слоя идентифицируются как условно закрытые комплексы. С полевой археологией непосредственно связаны защита и музеефикация памятников, а также противодействие незаконным, грабительским раскопкам и вандализму. Ценный материал, получаемый в результате проведения ряда археологических исследований, становится источником новых сведений в археологии, ряде дисциплин (нумизматика, эпиграфика, сфрагистика и др.) и некоторых отраслях исторического знания. Огромный информационный массив непрерывно поступающих новых данных требует

строгого соблюдения методики раскопок различных сооружений, определенных методических правил и приемов исследования культурных отложений и разнообразных артефактов. Предельно конкретная и объективная фиксация археологических материалов, перевод их в письменный и/или цифровой вид (описания, чертежи, фотографии, видеоматериалы и т.д.) создаёт один из наиболее широких разделов источниковедческой базы археологии – ее фактологический фонд.

На сегодняшний день в полевом археологическом исследовании достигнуты большие успехи. Детальная разработка способов исследования различных категорий памятников позволяет вести раскопки с предельно чёткой и объективной фиксацией и минимальной потерей информации, содержащейся в них. Значительную роль в этом играет стандартизация методов полевых исследований, что обеспечивает высокий уровень контроля качества работы и возможность применения современных информационных технологий.

В сбор данных входит два основных процесса: определение местоположения и обследование памятника и научная раскопка тщательно отобранных экземпляров. Очевидно, что определение места возможного расположения находки является первым этапом. Предварительную разведку можно производить совершенно любым методом. Для того, чтобы обеспечить обнаружение и исследование показательной выборки образцов на памятнике, еще до раскопок его поверхность тщательно исследуется, собирается и фиксируется максимально возможное число образцов на поверхностном уровне. В эти материалы также входят фотографии, обмеры и часто, особенно в УКР-проектах, пробы бурением или электронными устройствами.

По итогам первых исследований и раскопок даже сравнительно небольшого памятника специалистам приходится прорабатывать огромную кипу находок. Все найденные материалы складываются в полевой лаборатории для последующего анализа, очистки, сортировки. Помимо вещественных материалов, исследователям также надо проработать цифровые данные – фотографии, слайды, заметки, оцифровать данные, что представлены в виде записей в полевых дневниках, чертежей и графиков. Кроме того, еще имеются радиоуглеродные датировки и образцы пыльцы, другие особые находки, которые нужно передать на обследование специалистам. Первая стадия обработки данных, таким образом, происходит непосредственно на памятнике, где все вещественные находки очищают, сортируют, проводят защитную обработку, что позволяет довести их до археологической лаборатории для более тщательного изучения.

После доставки в лабораторию специалисты проводят более тщательный анализ данных, который выполняется с помощью специального оборудования и зачастую затягивается на достаточно длительный срок. В лабораторных условиях проводится классификация артефактов, идентификация материалов, изучение остатков пищи, радиоактивная датировка и т.д. Все эти анализы должны дать информацию, необходимую для интерпретации археологического материала. Информация, полученная в результате проведения анализов, классификации и прочих исследований, представляет собой огромный объём данных, который должен быть зафиксирован.

Археология – наука достаточно молодая (хотя само понятие появилось ещё в древней Греции) и активно развивающаяся. Поэтому неудивительно, что и в ней достаточно широко используются и развиваются информационные технологии, которые принято называть геоинформационными. Геоинформационные технологии применяются в археологии уже более 20 лет. Сегодня существуют несколько самостоятельных полностью сформировавшихся направлений их использования. В 2002 году в Институте археологии была создана группа археолого-географических информационных систем (АГИС), занимающаяся вопросами создания, разработки и внедрения геоинформационных методов в исследования археологов, а также применения методов дистанционного зондирования, геофизики и компьютерного моделирования. АГИС регулярно проводит собрания и круглые столы, где читаются доклады, посвященные исследованиям в различных областях, объединяющих археологию и геоинформатику. В частности, одна из секций, рассматриваемых на этих собраниях, называется «Геофизические методы в полевых археологических исследованиях». В 2015 году

в Москве проводилась вторая международная конференция «Археология и геоинформатика», где первым докладом стал доклад «Первый опыт создания геоинформационной системы национального масштаба «Археологические памятники России»».

Уже давно понятие «виртуальной археологии» стало достаточно распространенным. Повсеместно для реконструкции памятников археологической ценности используется многомерная графика, базы данных для хранения информации. При полевых археологических исследованиях широко применяются методы археогеофизики, позволяющие решить ряд задач, возникающих перед археологами, начиная от поиска места возможного положения памятника до пространственной реконструкции разрушенных и поврежденных объектов по геофизическим данным. Модели объектов и их реконструкций позволяют беспрепятственно в любой точке земного шара получить доступ к информации о них, а также воспроизвести и тиражировать их с помощью трёхмерной печати. Концепция виртуальной археологии была разработана, главным образом, для возможности проведения дискуссий среди учёных-историков о результатах моделирования для проверки научной гипотезы. В последнее время становится широко распространенной технология дополненной реальности, ставшая настоящим прорывом. Объект исследований, представленный в виде открытки со специальным зашифрованным графическим кодом, можно вращать, рассматривать с разных сторон, как будто он находится непосредственно перед экспертом. Популярность также набирают интерактивные выставки, где объекты, представляющие особую археологическую ценность, воспроизводятся в виде голограмм или объёмных фигур в специальных компьютерных средствах.

На сегодняшний день большую популярность приобрела система, позволяющая вводить достоверную информацию об археологических раскопках и памятниках. Она называется «географическая информационная система» (ГИС). ГИС – это компьютерная система для сбора, хранения, поиска, анализа и представления пространственных данных любого вида. Лучше всего её можно описать как БД с возможностью составлять карты. Она также способна генерировать новую информацию на базе хранящихся в ней данных. ГИС обладает огромным потенциалом для изучения распределения памятников и решения пространственных задач в археологии, особенно по артефактам, поселениям и культурам, разбросанным по регионам. С помощью ГИС можно исследовать абсолютно любые археологические объекты.

Исходя из перспективы археологических исследований, у ГИС есть огромное преимущество – это возможность обрабатывать большие массивы данных. Это особенно полезно, когда возникает проблема анализа сложных поселений. Аналитическую работу, на которую раньше уходили годы, можно проделать за считанные секунды. До появления ГИС большинство археологических обследований и раскопок поселений ограничивалось самим памятником. Теперь археологи могут выйти за рамки памятника и рассматривать, к примеру, природный потенциал тех областей, где памятники обнаружены не были, как путь к оценке общего распределения памятников в границах региона и возможного обнаружения новых, ранее не исследованных поселений.

В археологии ГИС применяют относительно недавно, но ее ценность очевидна. Помимо самой ГИС, активно разрабатываются системы, дополняющие её именно для области применения в археологии. К примеру, учёный состав Омского государственного университета им. Ф.М. Достоевского разработал геоинформационную систему археологических объектов ArGIS, предназначенную для интеллектуального анализа данных (Data Mining), получаемых ГИС.

Археологические исследования на территории Российской Федерации по сей день не имеют общего стандарта составления отчётности, то есть, и систематизма в целом. Общепринятые правила практически отсутствуют, различаются также и подходы к представлению материала, по этой причине и написание отчётов об археологических экспедициях становится «творческим» процессом.

Отчет об археологических исследованиях — это основная форма представления результатов археологических исследований. Он является основанием/средством для

последующих научных работ, базируемых на представляемых в нем данных. Отсутствие в отчете какой-либо важной информации, ложное или неясное ее изложение подрывает доверие к самому отчету и дальнейшим исследованиям.

Требования к оформлению научных отчетов о полевых археологических исследованиях (утверждены Главным архивным управлением в “Основных правилах работы ведомственных архивов” (1986 г.), Постановлением Президиума РАН “Об утверждении нормативных актов о совершенствовании архивного дела в Российской академии наук” № 125 от 21.06.94 г., решением Правительства Российской Федерации “О мерах по сохранению Архивного фонда Российской Федерации и улучшению использования архивных документов” от 25 ноября 1999 г.) и Положение о порядке проведения археологических полевых работ (археологических раскопок и разведок) и составления научной отчётной документации (Утверждено Решением Ученого совета Института археологии Российской академии наук от 30 марта 2007 г.) действуют и по сей день, однако следует отметить, что некоторые положения, которые прописаны в действующих ГОСТах, не очень удобны для применения на практике, а порой противоречат даже действующим правилам и нормам русского языка. Поэтому в практике многие исполнители НИР не применяют эти спорные положения. Ярким примером может служить действующий ГОСТ 7.32-2001 «Отчет о научно-исследовательской работе. Структура и правила оформления», неудобство применения многих положений которого привело к тому, что даже смежные ГОСТы, например, ГОСТ Р 53579-2009 «Отчет о геологическом изучении недр. Общие требования к содержанию и оформлению», допускают иные подходы.

Отчёты по археологическим исследованиям составляются руководителем экспедиции, с подробным описанием истории исследования памятника и чёткой обрисовкой всех проведенных раскопок, найденных артефактов, сооружений, чертежей местности, найденной стратиграфии, обязательно также требуется перенести основные дневниковые записи. Они должны полностью отображать всю проделанную работу, содержать выводы, результаты исследований и проведенных анализов, фотографии находок, аэрофотоснимки. Отчетность составляется в электронном виде, и на сегодняшний день именно её составление является большой проблемой. Огромные объёмы информации приходится вносить в компьютер вручную, а, учитывая отсутствие строгих форм составления отчёта, этот процесс значительно затягивается, поскольку всю полученную информацию надо, прежде всего, структурировать, связать всё в хронологической последовательности, а также перенести все записи из полевых дневников. Учитывая, что отчёты достигают объёма в 250 страниц, составление отчёта может затянуться на достаточно длительный срок.

Кроме того, проблемой является локальное хранение информации. Вероятность утери данных достаточно высока по ряду причин. Потеря информации по причине выхода из строя устройства хранения (жёсткие диски, флеш-носители, карты памяти на мобильных устройствах), – одна из самых распространённых ситуаций. Вполне вероятна потеря или порча записей и чертежей, сделанных на бумаге.

Также, если экспедиция проводится на достаточно обширной территории, поделена на части и выполняется различными исследовательскими группами, важным моментом является объединение данных с разных точек для проведения сравнительного анализа полученных результатов.

Если объединить все проблемы археологической экспедиции в единое целое, то можно сделать вывод – требуется создать некую информационную технологию, с помощью которой появится возможность избежать потери данных в ходе локального хранения непосредственно на месте проведения экспедиции. Рассмотрев некоторые из программных решений, можно сделать вывод, что на данный момент уже существуют продукты, которые могут решить ту или иную проблему археологической экспедиции. Однако единой системы, которая объединяла бы преимущества всех этих продуктов, не имела привязки к устройству, имела возможность интерактивно подгружать медиаданные в некое общее хранилище и помогала структурировать получаемые данные, ещё не было разработано и введено в эксплуатацию. Главное – осуществлять это таким образом, чтобы впоследствии, при составлении отчётной

документации, можно было выгрузить хранимую информацию в документ/специальную форму, и процесс формирования отчёта сокращался в разы.

На сегодняшний день существует автоматизированная информационная система «Археограф», разработанная С.А. Васильевым, кандидатом исторических наук, научным сотрудником ИИМК РАН, структура которой предусматривает хранение, поиск и анализ информации по памятникам в самой системе АРХЕОГРАФ, организацию ее непосредственного взаимодействия с геоинформационной системой для последующего анализа средствами ГИС с использованием цифровых карт любого масштаба. Этот продукт является едва ли не единственным программным средством для помощи археологам в их работе, но в нём есть существенные недостатки. В первую очередь, эта автоматизированная ИС является приложением для компьютера, мобильной версии нет. Это означает, что загружать данные с мобильных устройств непосредственно с памятника при использовании данного программного продукта нет возможности. Вторая проблема – поскольку АИС «Археограф» последний раз обновлялась в 2010 году, то в требованиях к программным средствам указано, что ОС компьютера, на котором она должна быть установлена – Windows 2000/XP. При попытке поставить программу на Windows 10 появляется ошибка времени выполнения 217. Следовательно, использование этой системы сейчас крайне неудобно.

На рынке существует множество программных продуктов для автоматизации составления отчетов. Подавляющая часть из них предназначена для использования в коммерческих предприятиях, где для муниципальных органов требуются чётко стандартизированные отчёты.

В этих средствах для автоматизации создания электронных отчётов данные загружаются из некоторой базы, и в результате их структуризации и обработки на выходе получается отчёт по строгой структурированной форме. Однако в большинстве своём эти программные продукты и автоматизированные информационные системы предназначены для обработки финансовой информации и создания бухгалтерских отчётов. К примеру, одной из таких программ является 1С:Предприятие, который занимает ведущие позиции на рынке. Она идеально подходит для работы с бизнес-документацией, однако её формат, равно как и формат других существующих на рынке программных продуктов по созданию отчётностей, не позволяет создавать отчёты по археологическим экспедициям по ГОСТам и требованиям, принятым в Положении и требованиям, рассмотренным ранее.

Для решения проблемы предлагается разработать автоматизированную информационную систему, в основе которой будет использоваться хранилище данных, концептуальная модель которого представлена на рисунке 1. Главная цель создания хранилищ данных – сбор информации из различных источников для последующего её представления в формате, который является удобным для обработки и анализа, а также организации последующей деятельности. Представляя руководителя экспедиции, которому предстоит разработка отчёта, в качестве аналитика, можно сделать вывод, что в данном случае использование технологии хранилища данных полностью оправдано, поскольку основной задачей их создания является удобство обработки и представления информации для аналитика.



Рис.1 – Концептуальная модель хранилища данных.

Так как одной из наиболее весомой из указанных выше проблем являлась опасность потери полученных и хранимых данных в результате какого-либо повреждения носителя информации, предлагается в качестве способа записывания данных использовать мобильные устройства (преимущественно планшеты) с возможностью мгновенной передачи файлов различных форматов на локальный сервер. Таким образом, данные централизованно хранятся на сервере, и шанс того, что информация будет повреждена или утеряна, близится к нулю.

Для загрузки и выгрузки данных предложено использовать Web-интерфейс. Поскольку работа будет вестись с локальным сервером, такой подход к реализации наиболее удобен и рационален. Интерфейс можно легко подстроить под пользователя, а взаимодействие с сервером происходит быстро и интерактивно. Это станет существенным преимуществом перед рассмотренной выше АИС «Археограф», поскольку нет зависимости от операционной системы устройства, с которого будет производиться работа с системой. Также достоинством подхода является отсутствие привязки устройств к какому-либо приложению, доступ к информации может быть получен в любой момент с любого устройства, единственное условие – наличие веб-браузера.

В заключении стоит сказать, что полученная в результате работы автоматизированная информационная система должна стать незаменимым помощником в проведении археологических экспедиций.

Библиографический список

1. Н.И. Винокуров. Полевые археологические исследования и археологические практики. /Винокуров Н. И.: Прометей; Москва; 2013
2. Д.А. Авдусин. Полевая книга СССР. Учеб. пособ. — 2-е изд., перераб. и доп. М.: Высш. школа, 1980. — 335 с.
3. Б. Фаган, К. ДеКорс. Археология. В начале. Техносфера; Москва; 2007. ISBN 978-5-94836-119-2, 0-13-032906-1
4. http://www.archeo.ru/struktura-1/otdel-ohrannoi-arheologii/nauchnye-proekty-otdela-ohrannoi-arheologii/copy12_of_proekty

УДК 681.3

Е.Б. Доронина, магистрант**Научный руководитель: Ю.В. Доронина, проф., С.В. Доценко, проф.***Севастопольский государственный университет***ОРГАНИЗАЦИОННОЕ УПРАВЛЕНИЕ ПРИ ПРОЕКТИРОВАНИИ ГЕОИНФОРМАЦИОННЫХ СИСТЕМ НА ОСНОВЕ SADT И RUP-ТЕХНОЛОГИЙ**

Аннотация: В статье рассматривается подход к управлению процессом создания (проектирования) комплексных проектов, связанных с реализацией сложных архитектур информационной и геоинформационной среды. Предложены особенности предлагаемого подхода для учета и оптимизации коммуникативных взаимодействий в проектных группах.

Ключевые слова: организационное управление, геоинформационные системы, проектирование систем, коммуникативные процессы, SADT-технологии.

Annotation: The approach to manage the creation (design) of complex projects related to the implementation of complex architectures of information and geoinformation environment is considered in the article. Peculiarities of the proposed approach for consideration and optimization of communicative interaction in project groups are presented.

Key words: organizational management, geoinformations systems, planning of the systems, communicative processes, SADT-technologies.

Введение. Организационное управление при процессе проектирования ГИС является сложноформализуемым процессом, находящимся в постоянной динамике, поэтому его моделирование, с целью определения эффективных механизмов, является важным этапом повышения оптимальности проектирования и создания ГИС.

Постановка проблемы. В современных условиях возрастающих требований к информационным (ИС) и техническим системам (ТС) играет важную роль развитие подходов к организации их проектирования и построения [1, 2]. Усложнение систем способствует усложнению проектных, конструкторских, экспериментальных и других работ по их созданию. Каждый вид этой деятельности сопровождается коммуникативными процессами, выраженными в обмене субъектов информацией, как на вербальном, так и на программном (интерфейсном) уровнях. Все сказанное относится и к особому классу ИС – геоинформационным системам (ГИС), специфика которых заключается в усложнении коммуникативных взаимодействий при их создании, связанному с привлечением различных специалистов (например, в области геоинформатики, геологии, геодезии и т.п.).

Результаты исследований. При проектировании ИС и ТС необходимо учитывать следующие факторы: 1 – динамичность требований, возникающих в процессе проектирования и 2 – высокую скорость устаревания проектируемой системы. На рис.1 показано взаимодействие специалистов при создании сложной системы, в том числе ГИС.

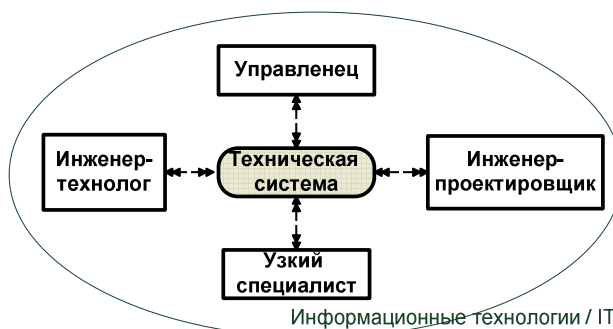


Рис. 1 – Взаимодействие специалистов при создании сложной системы

Таким образом, можно обозначить проблемы организационного управления, в том числе проблемы при обучении (подготовке) специалистов в области создания ГИС:

(1) Сложность ГИС заключается в гетерогенности компонент, коммуникативном многообразии и технологической неустойчивости в связи с быстрым устареванием технологий.

(2) Программы обучения специалистов в этой области ориентированы, в основном, на конкретные задачи создания ГИС, но осуществление комплексной прикладной подготовки специалистов в рамках системной инженерии крайне проблематично.

(3) Фактически проблемной частью процесса создания ГИС систем является отсутствие процессного проектирования в частности и организационного управления в целом.

На первом этапе применения принципов организационного управления предлагается реализация следующего процесса, последовательно задействующего задачи (1) - (3).

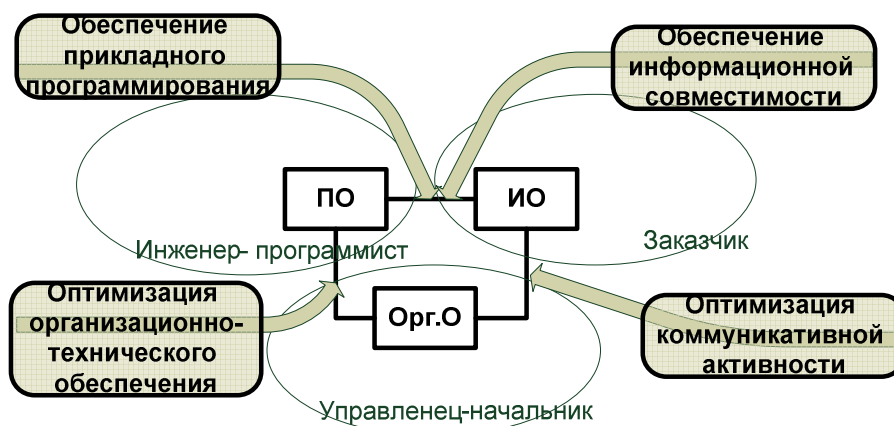


Рис. 2 – Задачи реализации составных элементов ИС:
ПО - программное обеспечение, ИО – информационное обеспечение,
Орг.О – организационное обеспечение

При реализации проекта ГИС предполагается, что наиболее эффективной является методология Rational Unified Process (RUP). Она опирается на проверенные практикой методы анализа, проектирования и разработки программного обеспечения (ПО), методы управления проектами. RUP обеспечивает прозрачность и управляемость процесса и позволяет создавать ПО в соответствии с требованиями заказчика, а также в соответствии с возможностями инструментальных средств поддержки разработки [3, 4]. На рис.3. приведена диаграмма сотрудничества из комплекса диаграмм методологии RUP, на которой отражен процесс взаимодействия (коммуникаций) при создании ГИС- проектов, описаны сущности, варианты использования и указана возможная локализация коммуникативного процесса при покупке ГИС заказчиком.

Анализ и синтез системы коммуникативных взаимодействий при проектировании сложных ГИС. На рис. 3-5 приведены схемы, связанные с описанием коммуникативных взаимодействий при проектировании сложных ИС (ГИС) в рамках организационного управления.

В основе описываемого подхода – методология функционального моделирования IDEF0 (*Integrated Definition Function Modelling*), принятая в 2000 г. Госстандартом РФ в качестве базы для моделирования реинжиниринга бизнес-процессов и процессов менеджмента качества [5]. В настоящее время международная организация ИСО (*International Organization for Standardization*), занимающаяся выпуском стандартов, изучает возможность использования методологии IDEF0 в качестве международного стандарта IPS (*Information Processing Standard*) по обработке информации.

Построенные диаграммы отражают декомпозированный процесс организационного управления созданием сложных ИС с уточнением в прикладной области ГИС – систем. Взаимодействие специалистов при создании сложной системы (рис.1) переходит в декомпозицию диаграммы «Процесс создания Сложной Системы» (рис.5), так же иерархическая подчиненность при создании ГИС отражена на рис. 5 в форме второй декомпозиции процесса «Руководитель», ветка А1.



Рис. 3 – Диаграмма Сотрудничества

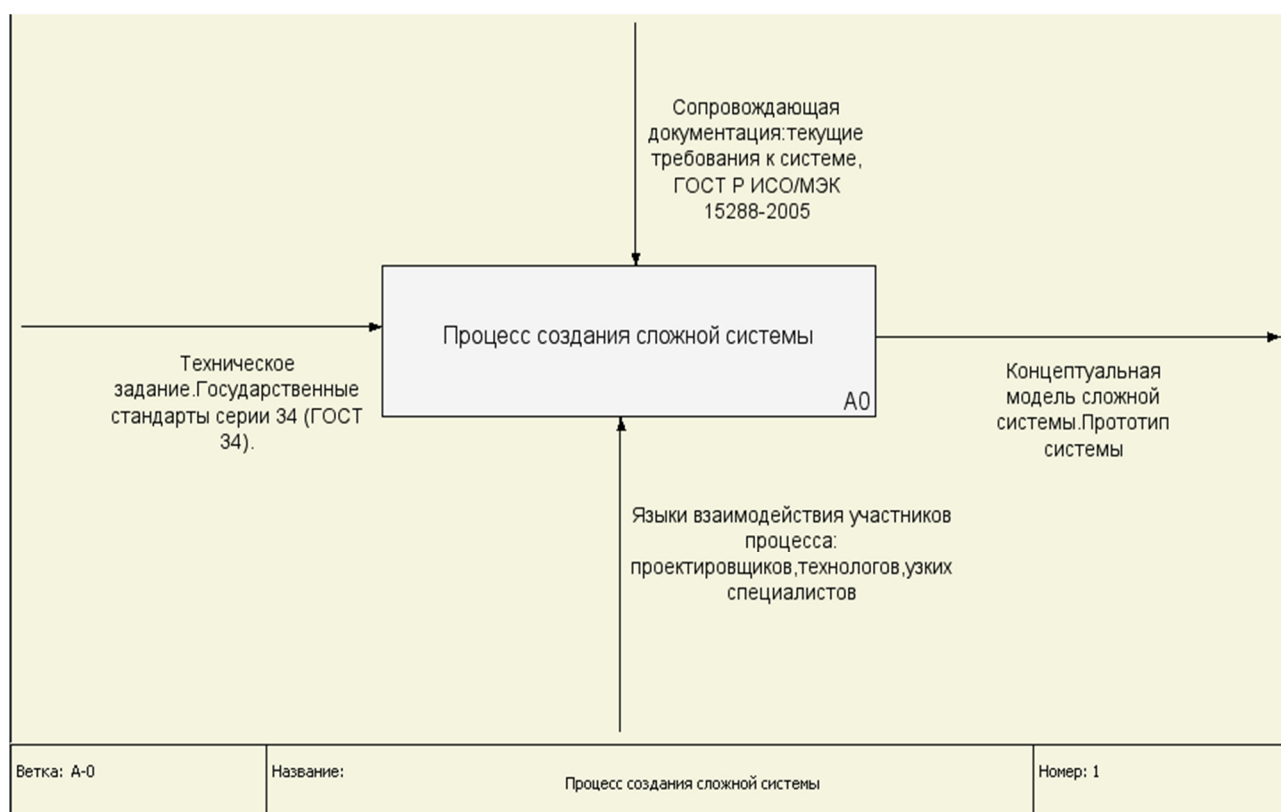


Рис. 4 – Контекстная диаграмма в нотации IDEF0

Предложенный подход позволил повысить эффективность создания ГИС, в том числе на основе следующих факторов:

1) графической формализации процессов при создании сложных иерархически структурированных систем;

2) получения инструментов для формального согласования и интеграции как общеизвестных методов, применяемых для создания систем выделенного класса, так и вновь разрабатываемых в процессе научного исследования методов и алгоритмов;

3) реорганизации и реинжиниринга процесса управления и согласования потоков данных при создании ГИС - проектов.

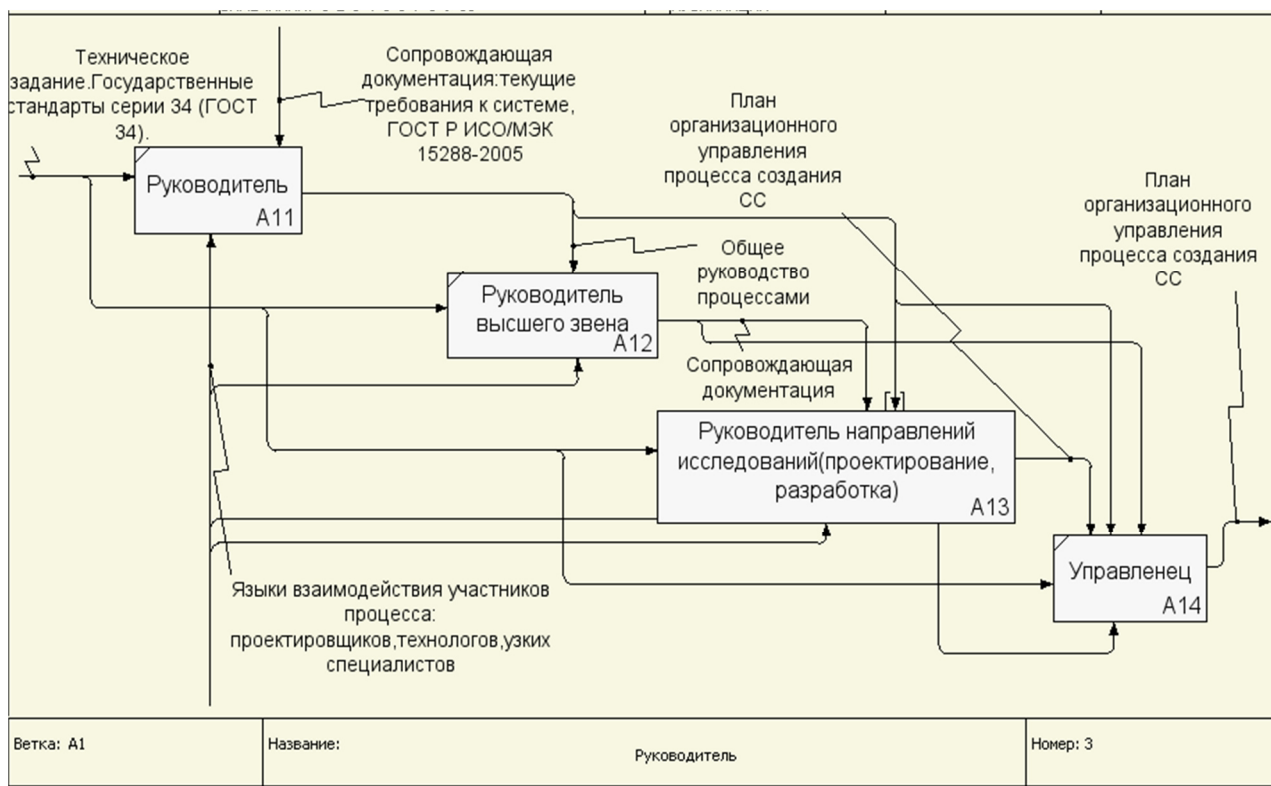


Рис. 5 – Вторая декомпозиция процесса «Руководитель», ветка A1

Выводы и рекомендации. В процессе исследования предложено комплексное применение SADT (в формате IDEF0) и RUP технологий для проектирования геоинформационных систем. Особенностью этих систем является многоуровневая структура организации и связи между управляющими и прикладными структурами.

Совокупное сочетание этих методологий позволило сформировать план организационного управления процессом создания ГИС - проектов, в том числе с учетом коммуникативных процессов.

Библиографический список

1. Проектирование информационных систем: учебное пособие / Г.В. Абрамов, И.Е. Медведкова, Л.А. Коробова. – Воронеж: ВГУИТ, 2012. – 172 с.
2. Зыков С.В. Основы проектирования корпоративных систем / С.В. Зыков; Нац.исслед.ун-т «Высшая школа экономики»: Изд. дом Высшей школы экономики. – Москва: 2012.– 431 с.
3. Галямина И. Г. Управление процессами: Учебник для вузов. Стандарт третьего поколения. – СПб.: Питер, 2013. – 304 с.
4. Вендров А.М. CASE-технологии. Современные методы и средства проектирования информационных систем.– М.: Финансы и статистика, 1998. – 176 с.
5. 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>



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 4.04.2017 г.

Уч.-изд. л. 22,45. Объем данных 10,9 Мб

ФГАОУ ВО «СевГУ», кафедра ИТиКС
299053, Севастополь, ул. Университетская, 33.

Тел.: (8692) 43-50-08

E-mail: itiks@sevsu.ru
