

Министерство науки и высшего образования РФ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ ЛЕСОТЕХНИЧЕСКИЙ
УНИВЕРСИТЕТ имени С. М. Кирова»

Кафедра информационных систем и технологий

ИНФОРМАЦИОННЫЕ СИСТЕМЫ И ТЕХНОЛОГИИ: ТЕОРИЯ И ПРАКТИКА

Сборник научных трудов

Выпуск 11

Санкт-Петербург
2019

Рассмотрен и рекомендован к изданию
Институтом леса и природопользования
Санкт-Петербургского государственного лесотехнического университета
22 апреля 2019 г.

Редакционная коллегия:

А. М. Заяц, кандидат технических наук, профессор (отв. редактор),
М. А. Шубина, кандидат технических наук, доцент (отв. секретарь),
С. П. Хабаров, кандидат технических наук, доцент

Составитель

кандидат технических наук, доцент (отв. секретарь) **М. А. Шубина**

Рецензенты:

доктор технических наук, профессор **И. В. Иванова**
(Национальный минерально-сырьевой университет «Горный»),
доктор технических наук, профессор ВТ НИУИТМО **А. А. Ожиганов**

Информационные системы и технологии: теория и практика:
сб. науч. тр. Вып. 11 / отв. ред. А. М. Заяц. – Санкт-Петербург:
СПбГЛТУ, 2019. – 162 с.

ISBN 978-5-9239-1112-1

Представлен кафедрой информационных систем и технологий.

Сборник подготовлен по материалам кафедры вуза, представленным на научно-технической конференции Института леса и природопользования СПбГЛТУ в феврале 2019 г., и практических работ, выполненных ее сотрудниками.

А.М. Заяц, кандидат технических наук, профессор
Кафедра информационных систем и технологий
СПб ГЛТУ им. С.М.Кирова
zamfta@yandex.ru

Т.И. Игнатьева, студентка ИЛиП
СПб ГЛТУ им. С.М.Кирова
ignatyvat@mail.ru

МАТЕМАТИЧЕСКАЯ МОДЕЛЬ ФУНКЦИОНИРОВАНИЯ БЕСПРОВОДНОЙ СЕНСОРНОЙ СЕТИ

Определим сенсорную сеть как распределенную систему сенсорных узлов, обменивающихся между собой и другими сетями информацией, характеризующей состояние объектов реального физического мира, с целью выработки ответной реакции на данную информацию. Задачей сенсорной сети является контроль и мониторинг измеряемых параметров физических сред и объектов [1-4].

Сенсоры – источники данных в сенсорной сети служат посредниками между физическими процессами, происходящими в окружающей среде и представлением этих процессов в цифровом виде показаний датчиков. В узлах автоматически выполняется нормализация, фильтрация, аналого-цифровое преобразование и первичная обработка сигналов с подключенных датчиков в микропроцессоре, а также диагностика возможных неисправностей датчиков.

Полученный цифровой код в виде пакета записывается в память микроконтроллера для дальнейшей обработки в процессоре и после временно хранится там же до отправки в радиоэфир.

Каждый сенсорный узел фиксирует эти показания и передает их шлюзам или базовой станции (БС) либо непосредственно, либо через другие сенсорные узлы. Шлюзы наряду с объединением БСС с другими сетями осуществляют агрегацию информации от группы датчиков с целью облегчения последующей передачи данных в центры управления или другие внешние объекты [2].

Для передачи информационных пакетов на базовую станцию сенсорный узел периодически посылает их в эфир. Эти пакеты принимаются, любым свободным для приема узлом-ретранслятором, расположенным в области радиовидимости передающего сенсорного узла, после чего в узле, принявшем пакет, происходит проверка контрольной суммы. В случае успешной проверки осуществляется дальнейшая обработка пакета. Здесь же может производиться добавление в пакет ранее принятых данных других узлов, расположенных в зоне радиовидимости и в дальнейшем передача этого пакета на базовую станцию.

При разработке математической модели в узлах сети выделим следующие подсистемы:

- коммуникационная подсистема - обеспечивает беспроводные соединения с другими узлами в сенсорной сети и содержит радио приемопередатчик;
- вычислительная подсистема - обеспечивает функциональность узла и обычно состоит из микроконтроллера, в состав которого обычно входят процессор, оперативная SRAM, энергонезависимая EEPROM и флэш-память, аналого-цифровой преобразователь (АЦП), таймер, порты ввода/вывода;
- сенсорная подсистема - обеспечивает соединение сенсорных беспроводных узлов с окружающей средой, параметры которой они измеряют, в ее состав могут входить аналоговые и цифровые сенсоры, актуаторы;
- подсистема электропитания - обеспечивает энергетическое снабжение всех элементов беспроводного сенсорного узла и включает устройства генерации и аккумуляции энергии;
- при моделировании процесса функционирования БСС будем рассматривать его на следующих независимых этапах [5, 6]:
 - контроль и фиксация параметров окружающей среды, прием и преобразование измерений, полученных в аналоговой форме, в цифровую для предварительной обработки в микропроцессоре узла и формирование пакета данных;
 - передача данных собственных датчиков в радиоэфир на базовую станцию;
 - прием измеренных параметров окружающей среды от других узлов сети с последующей передачей их на базовую станцию (шлюз).

Заметим, что прием измеренных параметров окружающей среды и передача обработанных данных в виде информационных пакетов производится одними и теми же устройствами приемо-передатчиками узла, а получение информации от датчика и предварительная ее обработка (преобразование аналоговых параметров в цифровые и формирование пакета для основной обработки на БС) производится микропроцессорами узлов.

Так как прием и передача производится одними и теми же устройствами – приемо-передатчиками сенсорных узлов, то в произвольный момент времени эти устройства могут быть заняты либо приемом, либо передачей.

Для формального описания процесса функционирования БСС ее можно представить многофазной, многоканальной системой массового обслуживания (СМО), процесс обслуживания заявок в такой системе позволяет адекватно описать все процедуры мониторинга состояния любого объекта, реализуемые с помощью БСС.

При построении модели введем некоторые ограничения и предпосылки, которые позволят учесть особенности структуры узлов сенсорной сети, параметры СМО и циркулирующие потоки заявок на различных этапах их обработки:

- множество $N = \{n_i\}_{i=1}^N$ приемо-передатчиков узлов БСС, где N – мощность множества определяемое числом сенсорных узлов;
- множество $K = \{k_i\}_{i=1}^I$ абонентов БСС;
- множество $\Lambda = \{\lambda_i\}_{i=1}^I$ интенсивностей потоков заявок от абонентов.

Абонентами, генерирующими потоки заявок на вход СМО, будем считать:

– датчики, фиксирующие параметры окружающей среды с последующей их обработкой в микропроцессоре и дальнейшей передачей информационных пакетов на базовую станцию;

– сенсорные узлы, передающие информационные пакеты на ближайший узел для ретрансляции.

Событие, состоящее в том, что от k -го абонента поступили измеренные данные на вход АЦП узла, интерпретируется как поступление заявки на вход СМО. Заявкой также считается пакет данных принятый приемником, от соседнего узла для ретрансляции его передатчиком.

Под обслуживанием заявок понимается процесс параллельно-последовательной их обработки на этапах описанных выше.

Для рассматриваемой системы массового обслуживания определим следующие ее параметры:

- количество $G = \{g_j\}_{j=1}^G$ этапов (фаз) обслуживания в СМО;
- количество $D = \{d_j\}_{j=1}^D$ обслуживающих приборов, составляющих СМО;
- множество $S = \{s_j\}_{j=1}^S$ параллельно обслуживающих приборов на g – фазе;
- множество $T = \{\tau_{gj}\}_{g=1, j=1}^G$ средних времен обслуживания заявок на g – фазе;
- множество $H = \{h_j\}_{j=1}^H$ дисциплин обслуживания заявок;
- множество $A = \{\lambda_i\}_{i=1}^I$ интенсивностей входящих потока заявок от различных абонентов.

Упорядоченную совокупность множеств $Z = (G, D, S, T, H, A)$ будем называть вектором параметров СМО, который полностью ее описывает.

Рассмотренные исходные предпосылки и введенные ограничения позволяют приступить к непосредственному построению модели процесса функционирования БСС для чего необходимо:

– провести анализ возможных состояний БСС, при принятых предпосылках и ограничениях;

– рассчитать количество возможных состояний беспроводной сенсорной сети $E = \{e_\xi\}_{\xi=1}^E$;

– определить возможные переходы из состояния в состояние (задать граф возможных переходов);

– определить вероятности пребывания процесса в e_ξ состоянии;

– оценить качество функционирования БСС.

Состояния процесса обслуживания заявок в СМО будем различать по количеству обслуживающих приборов (приемников, микроконтроллеров, передатчиков узлов), одновременно занятых обслуживанием и по номеру этапа, на котором находится каждая обслуживаемая заявка (под этапами будем понимать нахождение заявки на соответствующем обслуживающем приборе).

Тогда каждому состоянию e_ξ процесса из множества $E = \{e_\xi\}_{\xi=1}^E$ можно поставить в соответствие вектор занятости $\omega_\xi = (\omega_{1\xi}, \omega_{2\xi}, \omega_{3\xi}, \dots, \omega_{G\xi})$, каждый компонент $\omega_{g\xi}$ которого является случайной целочисленной величи-

ной, определяющей количество приборов, одновременно занятых обслуживанием заявок, а индекс g соответствует этапу, на котором она обслуживается. Компоненты вектора занятости ω_ξ однозначно определяют количество занятых приборов на g этапе (приемо-передатчики и микропроцессоры БСС), а также суммарное количество заявок Ξ_ξ обслуживаемых в e_ξ состоянии

$$\Xi_\xi = \sum_{g=1}^G \omega_{g\xi}, \quad \forall \xi = 1, E. \quad (1)$$

Введем множество Ω , элементами которого являются векторы ω_ξ , характеризующие каждое состояние процесса, тогда множество $\Omega = \{\omega_\xi\}_{\xi=1}^E$ векторов занятости однозначно задают все состояния рассматриваемого процесса.

Процесс обслуживания заявок в СМО можно представить как блуждание точки по узлам E - мерной решетки, пространственные координаты состояний которой задаются компонентами соответствующих векторов ω_ξ

Конфигурация пространственной решетки определяется тем, что для любого состояния ω_ξ должно выполняться следующее условие: количество приемников и передатчиков, одновременно занятых обработкой данных, не должно превышать общего числа узлов БСС

$$\sum_{\substack{g=1 \\ g \neq g^*}}^G \omega_{g\xi} \leq N, \quad \forall \xi = 1, E \quad (2)$$

где g^* - номер этапа, на котором ведут обработку процессоры микроконтроллеров сенсорных узлов.

Это условие вытекает из того, что приемо-передатчик узла может вести как прием, так передачу данных, но в произвольный момент времени t он может находиться либо в состоянии приема, либо передачи, одновременно эти два процесса в узлах в момент времени t существовать не могут. Принять на обслуживание - прием или передачу может любой свободный приемо-передатчик.

Рассмотренному процессу в СМО можно поставить в соответствие граф с началом в точке θ , координаты $\omega_1 = \theta$, $\omega_2 = \theta$, $\omega_3 = \theta$.

Геометрические размеры и конфигурация графа определяются значением количества N узлов БСС и введенными ограничениями.

На рисунке изображен граф переходов с соответствующей нумерацией состояний и направлениями возможных переходов процесса обслуживания заявок в СМО.

Вершины графа изображены в виде окружностей, внутри которых показан вверху номер e_ξ , а внизу численные значения компонентов вектора занятости ω_ξ , характеризующего e_ξ состояние. Здесь для примера, приведен граф для БСС, включающий два узла. При принятых допущениях и ограничениях конфигурация графа моделируемого процесса и направления возможных переходов останутся прежними при любом количестве узлов сети.

Особенности обработки измеренных данных от датчиков в БСС позволяет описать процесс обслуживания заявок в СМО полумарковским процессом [7] (рис. 1).

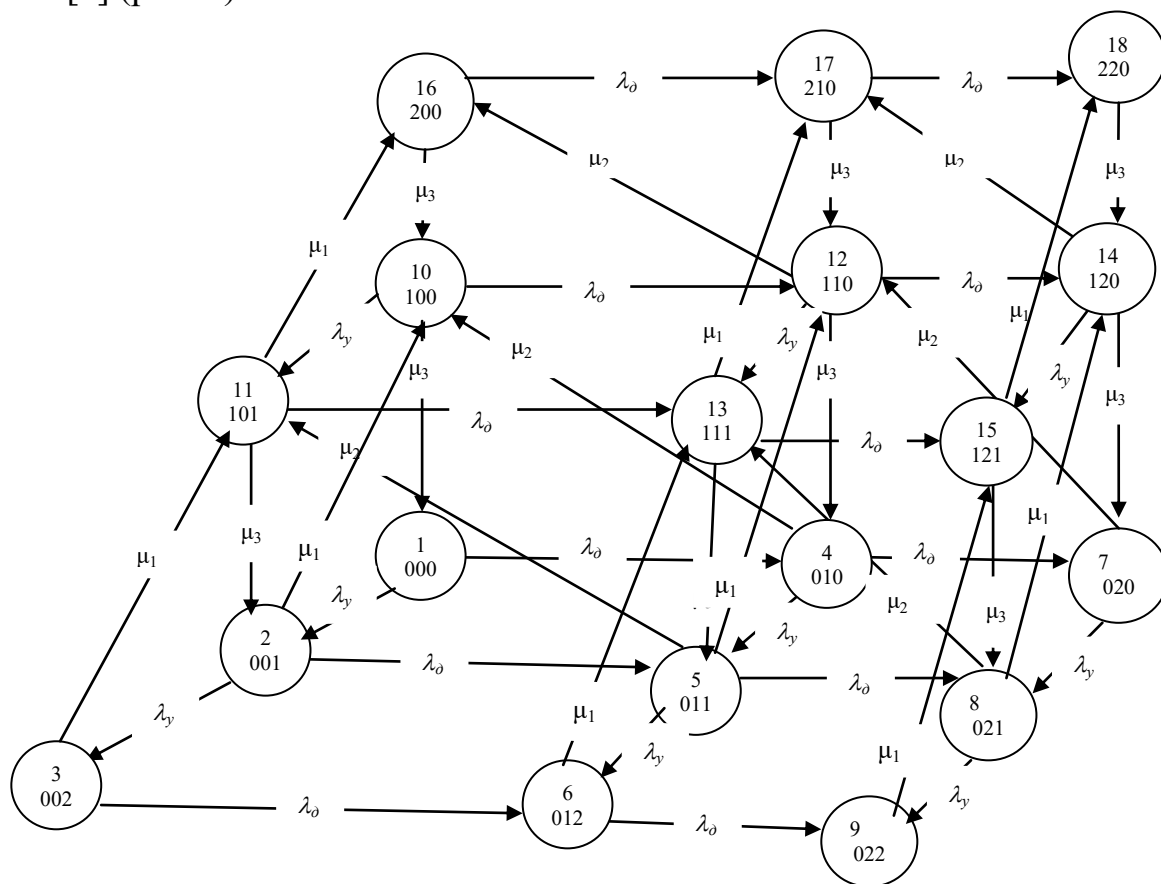


Рис.1. Пример графа моделируемого процесса

Такой процесс эволюционирует в соответствии с очередностью поступления заявок и параллельно-последовательным их обслуживанием соответствующими приборами на разных этапах. Если рассматривать процесс только в моменты смены состояний (окончание обслуживания на g – ом этапе), то функционирование БСС описывается конечной цепью Маркова с матрицей переходных вероятностей P_{ij} и матрицей функций распределения $F_{ij}\eta_{ij}(t)$, ($t \geq 0, i, j \in [1, E]$) длительностей пребывания процесса в состоянии i при условии, что следующий переход будет в состоянии j .

Для Марковской цепи справедливо то, что, если в i – ом состоянии процесс находится случайное время t_i , имеющее показательное распределение с параметром H_i , равным сумме интенсивностей переходов η_{ij} во все соседние состояния, то функция распределения безусловного времени пребывания процесса в i – ом состоянии определяется следующим образом:

$$F_i(t) = 1 - \exp\{-H_i t\}, \quad t \geq 0, \quad (3)$$

а среднее время пребывания в i – ом состоянии $m_i = 1/H_i$ (4).

При известных интенсивностях переходов η_{ij} вероятности перехода за один шаг определяются из выражения

$$P_{ij} = \eta_{ij} m_i \quad (5)$$

Получим в общем виде соотношения, связывающие элементы и их значения в матрице переходных вероятностей P_{ij} с координатами вектора ω_ξ процесса в пространстве его возможных состояний $E = \{e_\xi\}_{\xi=1}^E$. Здесь и далее будем использовать следующие обозначения:

- Λ_Σ - интенсивность суммарного входящего потока заявок $\Lambda_\Sigma = \{\lambda_y, \lambda_d\}$;
- μ_1 - параметр обслуживания заявок приемников узла, равный $1/\tau_1$, $\tau_1 \in T$;
- μ_2 - параметр обслуживания заявок в микропроцессоре узла, равный $1/\tau_2$, $\tau_2 \in T$;
- μ_3 - параметр обслуживания заявок передатчиком, равный $1/\tau_3$, $\tau_3 \in T$.

Рассмотрим произвольное e_ξ состояние, задаваемое вектором $\omega_\xi = (\omega_{1\xi}, \omega_{2\xi}, \omega_{3\xi})$ и характеризуемое тем, что обработкой занято r свободный микропроцессоров $\omega_{2\xi} = r$, $0 \leq r \leq N$ и $p = \beta + \alpha$ приемо-передатчиков, причем β обслуживания приема $\omega_{1\xi} = \beta$, а α обслуживанием перед передачи $\omega_{3\xi} = \alpha$ и $0 \leq p \leq N$. Так как процесс в моменты перехода из состояния в состояние обладает марковским свойством, то ординарность событий, переводящих систему из состояния e_ξ в соседние состояния, позволяет установить тот факт, что пространственные координаты любого e_φ , $\varphi \in E$ соседнего состояния связаны с координатами ξ -го состояния условием $\omega_{1\varphi}, \omega_{2\varphi}, \omega_{3\varphi} = (\omega_{1\xi} + C), (\omega_{2\xi} + C), (\omega_{3\xi} + C)$, где $C = 0, \pm 1$.

Так, если процессор микроконтроллера узла провел обработку измененных данных и сформировал пакет для передачи, то система переходит в соседнее состояние e_φ , задаваемое вектором $\omega_\varphi = (\omega_{1\xi}, \omega_{2\xi} - 1, \omega_{3\xi} + 1)$. Если же передатчик передал пакет, то переход произойдет в состояние

$$\omega_\sigma = (\omega_{1\xi}, \omega_{2\varphi}, \omega_{3\varphi} - 1) \text{ и т.д.}$$

Принимая во внимание вышеизложенное можно записать, в общем виде выражения для определения вероятностей перехода из одного состояния в любое соседнее, при условии, что времена обслуживания на каждом этапе имеют показательное распределение:

$$P_\xi(\omega_1, \omega_2, \omega_3), \varphi(\omega_1, \omega_2, \omega_3 + 1) = \lambda_d \tau_\xi(\omega_1, \omega_2, \omega_3),$$

$$P_\xi(\omega_1, \omega_2, \omega_3), \varphi(\omega_1, \omega_2 + 1, \omega_3) = \lambda_y \tau_\xi(\omega_1, \omega_2, \omega_3),$$

$$P_\xi(\omega_1, \omega_2, \omega_3), \varphi(\omega_1 - 1, \omega_2, \omega_3 + 1) = \omega_{1\xi} \mu_1, \tau_\xi(\omega_1, \omega_2, \omega_3),$$

$$P_\xi(\omega_1, \omega_2, \omega_3), \varphi(\omega_1, \omega_2, \omega_3 - 1) = \omega_{3\xi} \mu_3, \tau_\xi(\omega_1, \omega_2, \omega_3),$$

Среднее время пребывания процесса τ_ξ в любом произвольном состоянии $\xi(\omega_1, \omega_2, \omega_3)$ вычисляется по формуле

$$\tau_\xi(\omega_1, \omega_2, \omega_3) = \frac{1}{\lambda_d Y + \lambda_y U + \sum_{g=1}^G \omega_g \xi \mu_g}, \quad (7)$$

где U – ограничитель, $U = \begin{cases} 0, & \omega_1\xi + \omega_3\xi = N \\ 1, & (\omega_1\xi + \omega_3\xi) < N \end{cases}$,

Y – ограничитель, $Y = \begin{cases} 0, & \omega_2\xi = N \\ 1, & \omega_2\xi < N \end{cases}$.

Получением выражений для элементов матрицы P вероятностей переходов фактически заканчивается процедура построения математической модели функционирования БСС.

Решение задачи исследования многоканальных, многофазных систем массового обслуживания, особенно с использованием алгоритма перебора - числа обслуживающих приборов на каждом этапе обработки, приводит к необходимости вывода уравнений описывающих СМО в общем виде. Для этого необходимо получить совокупность соотношений, которые позволят построить алгоритм для решения уравнений (6) при заданном числе этапов (фаз) G и варьируемом числе N узлов БСС.

Поставленную задачу следует решать в следующей последовательности:

1. Получить выражение для определения количества E состояний исследуемого процесса.
2. Разработать порядок нумерации в модели.
3. Вывести выражения, связывающие пространственные координаты состояний $\Omega = \{\omega_\xi\}_{\xi=1}^E$ с их номерами.
4. Разработать методику оценки показателей эффективности функционирования БСС.
5. Разработать алгоритм для определения стационарных вероятностей пребывания процесса в e_ξ состоянии и получения численных значений показателей эффективности функционирования БСС.

Используя представленную методику и построив математическую модель, можно определить и оценить показатели эффективности функционирования БСС, например, пропускную способность сети в целом и отдельных ее компонент, суммарное время занятости отдельных подсистем и другие ее параметры.

Библиографический список

1. Заяц А.М., Логачев А.А. Информационная система мониторинга лесов и лесных пожаров с использованием беспроводных сенсорных сетей. //Известия Санкт-Петербургской лесотехнической академии. 2016. № 216. - с. 241-254.
2. Заяц А.М., Думов М.И. Обзор беспроводных сенсорных сетей и технологий информационных систем оценки лесной пожароопасности и мониторинга лесов. //Сборник научных трудов «Информационные системы и технологии: теория и практика» - СПб.: СПбГЛТУ, 2016. №8. – с.9-21.
3. Заяц А.М., Хабаров С.П. Организация доступа к беспроводным AD НОС сетям информационных систем мониторинга лесных территорий из

среды Windows 10. // Известия Санкт-Петербургской лесотехнической академии. – СПб, 2018, Вып. 223. - с. 285-299.

4. Заяц А.М., Хабаров С.П. Настройка беспроводных соединений в системах мониторинга лесных территорий. // Леса России: политика, промышленность, наука, образование / материалы третьей международной научно-технической конференции. Том 1 – СПб.: СПбГЛТУ, 2018. – с.80-83.

5. Заяц А.М., Логачев А.А. Математические модели для поддержки принятия решений по предупреждению лесных пожаров при ограниченном объеме исходных данных. // Известия высших учебных заведений. Приборостроение. 2016. Т. 59. № 5. С. 342-347.

6. Хабаров С.П. Моделирование Ethernet сетей в среде OMNeT++ INET framework // Научно-технический вестник информационных технологий, механики и оптики. 2018. Т. 18. № 3. с. 462–472. doi: 10.17586/2226-1494-2018-18-3-462-472

7. Б. В. Гнеденко, И. Н, Коваленко. Введение в теорию массового обслуживания. – М.: Наука, 1987.

Н.П.Васильев, кандидат технических наук, доцент
Кафедра информационных систем и технологий
СПбГЛТУ им. С.М.Кирова
nikpv@mail.ru

Н.В.Лушкин, кандидат технических наук, доцент
Кафедра информационных систем и технологий
С ПбГЛТУ им. С.М.Кирова
Lushkin52@mail.ru

ИНТЕГРАЦИЯ ГИБРИДНЫХ ПРИЛОЖЕНИЙ CORDOVA С WEB-СЕРВЕРНОЙ ОБРАБОТКОЙ ГРАФИКИ

Введение

В настоящее время технология разработки гибридных мобильных приложений на основе cordova (<https://cordova.apache.org/>) становится всё более популярной. Объясняется это наличием огромной армии web-разработчиков, которые здесь могут реализовать свои профессиональные навыки. С другой стороны, кроссплатформенность этой технологии (она подходит для разработки под любую мобильную платформу: Android, iOS, Windows Phone и т.д.), относительная простота и более краткие сроки обучения (по сравнению с нативными технологиями) также играют свою положительную роль в её популярности [1,2].

Вместе с тем, основным недостатком гибридных приложений традиционно считается их низкая производительность, поскольку они работают опосредованно через компонент webView мобильной платформы. То есть там, где

требуется максимальное быстродействие, эта технология малопригодна. В том числе — это задачи с быстрой графикой: у `webview` есть некоторые проблемы при обработке графики типичной для игр или приложений с динамичным графическим интерфейсом. Хотя надо понимать, что компоненты `webview` постоянно совершенствуются и растёт мощность самих мобильных гаджетов. С другой стороны, если это не касается непосредственно интерфейса, никто не запрещает разрабатывать плагины, которые являются нативной частью разработки, то есть взаимодействуют непосредственно с API платформы, пишутся на Swift, Си или Java, и могут прикрывать узкие места приложения, требующие максимальной производительности. Напомним, что плагины в гибридных приложениях `cordova` — это, как раз, те добавки, которые используются для выхода за пределы `webview`, прежде всего, для обращения к внешней периферии гаджета — камере (фото, видео), геолокации, сети, акселерометру и тому подобное. Конечно, если похожих плагинов в экосистеме не найти, то всё равно придётся обращаться к их нативной разработке.

Альтернативой плагинам может оказаться распределённая обработка, например, с помощью `web-серверов`, мощность которых многократно превосходит гаджеты. Это может оказаться полезным, например, для обработки графических файлов. То есть интерфейс и взаимодействие с периферией гаджета разрабатывается в рамках возможностей `webview` и стандартных плагинов, а затратную часть с обработкой графики доверяем `web-серверу`. Для реализации подобного подхода, которому, как раз, и посвящается настоящая статья, придётся решить ряд подзадач:

- Выбрать фреймворк для разработки интерфейса приложения под `webview`. Выбор в настоящее время достаточно велик — это и `Ionic` (<https://ionicframework.com/>) и `jQuery Mobile` (<https://jquerymobile.com>) и `Sencha ExtJS Modern` и `Sencha Touch` (<https://sencha.com>) и др. Основная задача, которая при этом решается — это разработка интерфейса мобильного приложения, пригодного для `touch-экранов` небольшого размера, управляемого жестами.

- Обеспечить доступ к фотокамере и фотоальбому гаджета с возможностью получения доступа непосредственно к графическим файлам — сделанным фотографиям.

- Реализовать работу с файловой системой гаджета. Обычно это — "песочница" приложения, в файлах которой можно, например, сохранять настройки или последний рабочий файл с графикой.

- Обеспечить копирование файла из кэша камеры в "песочницу" приложения.

- Реализовать передачу файла на сервер и обратно — загрузку результатов обработки. Отметим, что возможностей `Ajax`, в данном случае может не хватить, поэтому необходимы иные решения;

- Реализовать алгоритм обработки изображения, например, на `php` с помощью графической библиотеки `GD` (впрочем, это можно сделать и на Си).

Для реализации поставленных задач функциональных возможностей только одного webview недостаточно – потребуются плагины cordova. Архитектура такого приложения вместе с необходимыми плагинами представлена на рис.1. Как видно из рисунка потребуется доступ к камере устройства, к хранилищу с данными (storage), а для передачи графики на сервер - доступ к сетевым возможностям устройства (WiFi, 2G, 3G или 4G-сеть оператора связи). При этом достаточно воспользоваться стандартными плагинами:

- cordova-plugin-camera для получения доступа к файлу с отснятым изображением непосредственно с камеры или из альбома мобильного устройства;
- plugin-file для записи и чтения содержимого файлов в или из хранилища устройства (storage);
- plugin-file-transfer для копирования файлов из одного места в другое, а также для передачи и загрузки изображения на сервер;
- plugin-network-information – сервис, который позволяет определить наличие связи того или иного вида: WiFi, 2G, 3G, 4G.

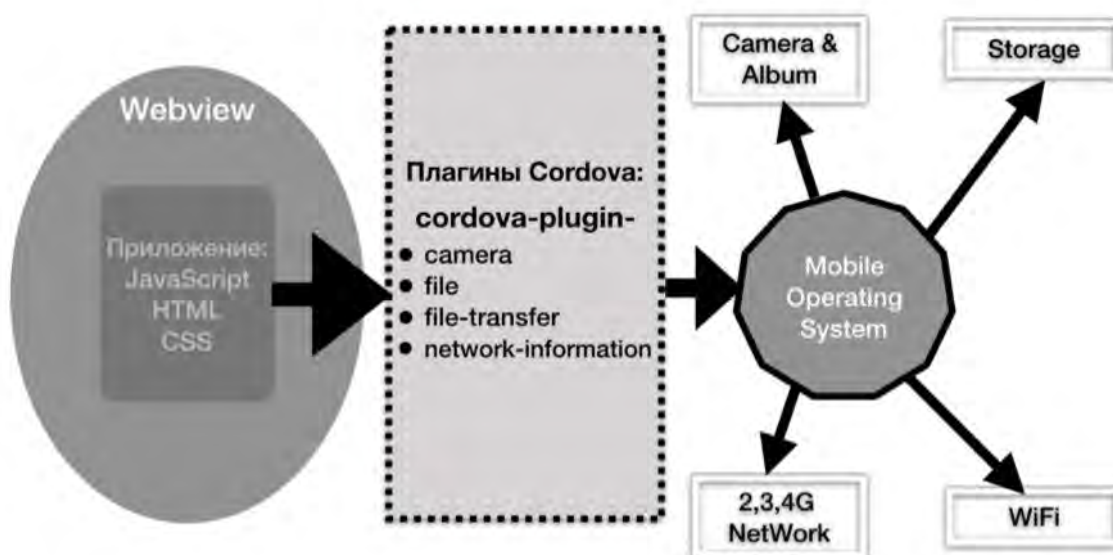


Рис. 1. Архитектура гибридного приложения, обеспечивающего распределённую обработку графики

Взаимодействие приложения с web-сервером схематично представлено на рис. 2. Для передачи фото используется плагин file-transfer, который, используя возможности http-протокола, загружает фото, например, в файл 1.jpg. С помощью php сделать это очень просто:

```
<?php
header('Access-Control-Allow-Origin: *');
move_uploaded_file($_FILES["file"]["tmp_name"], "./1.jpg");
?>
```

Для разработки под iOS придётся прибегнуть к платной программе сертифицированных разработчиков и освоить непростой процесс подписи и распространения приложений, который здесь не обсуждается.

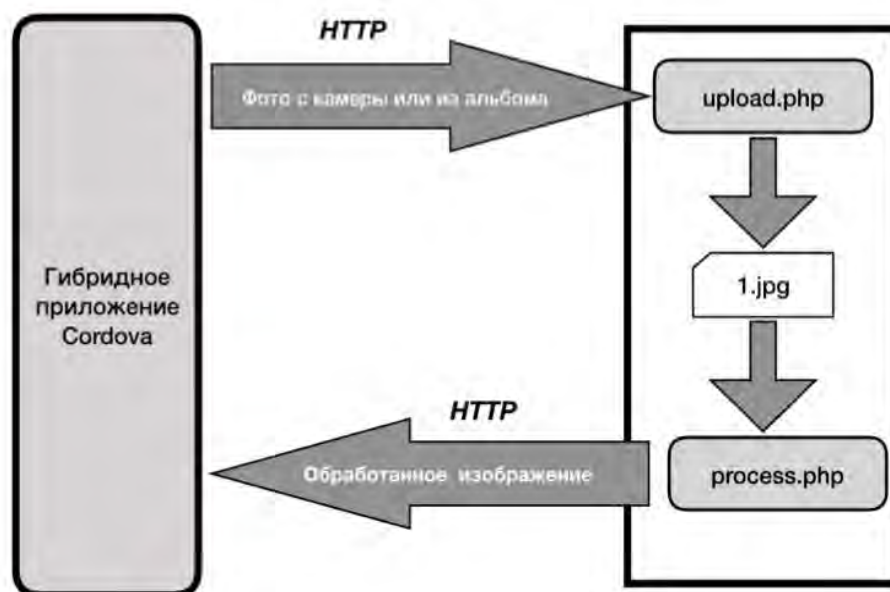


Рис. 2. Взаимодействие приложения с web-сервером

Выбор фреймворка

Отметим, что количество всевозможных JavaScript-фреймворков, известных и малоизвестных, популярных и менее популярных достаточно велико, чтобы в них можно было легко потеряться разработчику, а их развитие настолько динамично, что ситуация может измениться прежде, чем читатель прочтёт эти строки. **Некоторые из фреймворков ориентированы исключительно на разработку под браузер и малопригодны для мобильных устройств, отличающихся малыми размерами touch-экранов и использованием жестов.** Эти отличия характерны не только для браузеров, но и для компонентов webview, используемых гибридными приложениями.

Неплохим выбором является библиотека Ext JS фирмы Sencha (<https://www.sencha.com/products/extjs/>). Коммерческий вариант этого продукта стоит довольно дорого, однако, есть некоммерческий бесплатный вариант (<https://www.sencha.com/legal/GPL/>), использование которого регламентируется соответствующей лицензией GPL (General Public License). Фреймворк можно найти на github по ссылке <https://github.com/khusamov/sencha-extjs/releases/tag/6.2.0>.

Sencha Ext JS – это одна из самых мощных в настоящее время библиотек, ориентированная исключительно на язык JavaScript, хорошо документированная и имеющая общую основу как для разработки desktop-приложений под web, так и приложений для экранов мобильных устройств [3]. Если первая версия называется Ext JS Classic, то вторая – Ext JS Modern. Другим достоинством фреймворка на основе этой библиотеки является возможность интеграции с платформой cordova для разработки гибридных приложений. Поскольку при описании ключевых моментов обсуждаемой в статье технологии будем использовать именно эту библиотеку, а точнее – Ext JS Modern, то далее представим минимальную и выверенную последовательность действий настройки работы с фреймворком.

Интеграция Sencha Ext JS и Cordova

Для работы с библиотекой Ext JS достаточно скачать бесплатную программу `sencha-cmd` (<https://www.sencha.com/products/extjs/cmd-download/>) – это полнофункциональный инструмент командной строки (command line interface). Следует скачать версию со встроенной поддержкой Java, если Java не была установлена. Отметим, что компания предлагает также более удобные средства визуальной разработки, отладки и тестирования приложений Sencha Ext JS, но они – платные. В последнее время компания также активно пропагандирует специально разработанные модули под известную платформу `node.js` (<https://nodejs.org/en/>), которые полностью заменяют функционал `sencha-cmd`.

Настройка рабочей среды очень простая. Далее предполагается, что путь к команде `sencha` внесён в переменную окружения `path` - на `mac` и `unix` для этого выполняем команду:

```
export PATH=~/.bin/Sencha/Cmd/6.0.0.92:$PATH
```

в windows:

```
set PATH=%PATH%;C:\Sencha\Cmd\6.0.0.92
```

где, конечно, `6.0.0.92` меняем на актуальную версию.

- Инициуируем рабочее пространство – каталог с разрабатываемыми приложениями (`/path/to/workspace`):

```
sencha generate workspace /path/to/workspace
```

- Переходим в этот каталог:

```
cd /path/to/workspace
```

- Создаём приложение – стартовый шаблон:

```
sencha -sdk /path/to/framework generate app modern  
AppName ./AppPath
```

Путь к разархивированному фреймворку (`/path/to/framework`) указываем только один раз – фреймворк будет скопирован в рабочее пространство. Для последующих применений команда выглядит проще:

```
sencha generate app -ext -modern AppName ./AppPath
```

Здесь `AppName` – имя приложения, `AppPath` – каталог, где приложение будет развёрнуто.

- Переходим в каталог приложения и выполняем сборку:

```
cd TstApp  
sencha app build
```

- В результате будет создано полностью рабочее приложение, которое можно посмотреть в браузере (<http://localhost:1841/TstApp/>), если выполнить команду:

```
sencha app watch
```

Эта команда запустит web-сервер, слушающий порт 1841.

Интеграция с cordova также выполняется достаточно просто. Cordova – это модуль платформы node.js. Поэтому для его установки скачиваем предлагаемую на официальном сайте <https://nodejs.org> версию node (из двух вариантов рекомендуется выбирать не последнюю, а рекомендуемую для большинства пользователей – Recommended For Most Users). Платформа node.js работает в любой операционной системе и её установка, обычно, не вызывает проблем. Вместе с node будет установлен менеджер управления модулями npm. Устанавливаем модуль cordova глобально - выполняем в терминале с правами администратора следующую команду:

```
npm install -g cordova
```

Для интеграции проекта с cordova теперь достаточно выполнить ряд простых команд (всё делается из каталога приложения TstApp):

- Иницилируем cordova:

```
sencha cordova init learn.senchaCordova.TstAppId TstApp
```

Здесь `learn.senchaCordova.TstAppId` – это ID приложения (AppID в терминах iOS), которое строится по таким же принципам, как и доменные имена серверов (DNS), только в обратном порядке и с учётом регистра. Следует тщательно продумывать этот идентификатор, особенно для iOS-приложений, которые реально планируются развивать вплоть до загрузки в App Store. В результате будет создан подкаталог cordova, в котором будут находиться всё необходимое для успешной работы этой платформы в рамках проекта приложения Sencha Ext JS.

• В файле `app.json`, который находится в каталоге проекта приложения (TstApp) и хранит в популярном формате json его конфигурацию, следует указать мобильную платформу или платформы (достаточно раскомментировать соответствующую строку):

```
"builds": {
  "web": {"default": true},
  "native": {
    "packager": "cordova",
    "cordova": {
      "config": {
        // Uncomment the line below and add the platforms you
        // wish to build for
        // "platforms": "ios android",
        "platforms": "ios",

        "id": "learn.senchaCordova.TstAppId ",
        "name": "TstApp"
      }
    }
  }
}
```

- Наконец, выполнить сборку мобильного гибридного приложения cordova можно так:

```
sencha app build native
```

В результате выполнения этой команды в каталоге cordova/platforms/ios будет собран проект, готовый к загрузке в среду нативной разработки Xcode (для загрузки достаточно дважды кликнуть по файлу проекта с расширением xcodeproj). Заметим, что при указании платформы Android (или любой другой, поддерживаемой cordova), был бы создан каталог с проектом под Android Studio.

- Если приложение использует базовые плагины cordova, то их можно установить из каталога cordova стандартным способом, например, плагины перечисленные выше устанавливаются следующим командами:

```
cordova plugin add cordova-plugin-camera  
cordova plugin add cordova-plugin-file  
cordova plugin add cordova-plugin-file-transfer  
cordova plugin add cordova-plugin-network-information
```

После установки плагинов, приложение следует пересобрать. Вообще - вся работа с cordova полностью сосредоточена в одноимённом подкаталоге проекта. В этом подкаталоге можно выполнять любые другие действия, предусмотренные платформой cordova.

- Разработка приложения ведётся исключительно в рамках подкаталога app проекта на языках Java Script, HTML и CSS согласно современной концепции – MVC (Model View Controller). Модель MVC позволяет разделить весь программный код приложения на три части - код, определяющий данные приложения, пользовательский интерфейс и управляющую логику.

Камера (фотоальбом) устройства в проекте приложения Sencha Ext JS - Cordova

Плагин camera предоставляет полное имя графического файла, в котором содержится только что сделанный снимок или уже готовый снимок из альбома устройства (в зависимости от настроек). Этот файл находится в кэше устройства, то есть в памяти, где его сохранность не гарантируется, поэтому в первую очередь желательно скопировать его в безопасное место - память, выделенную для приложения. Во вторую очередь, конечно, следует каким-то образом представить изображение. В Sencha Ext JS наиболее подходящим для этих целей является компонент Ext.Image, который располагает свойством src и соответствующим методом setSrc(), позволяющим установить путь к файлу с изображением.

Таким образом, логика работы с камерой выглядит следующим образом. Обращаемся к плагину (navigator.camera):

```
navigator.camera.getPicture(success, fail, options);
```

Если работа с камерой прошла успешно (отсняли изображение, или выбрали его в альбоме), то управление передаётся функции обратного вызова

- success, в которой копируем это изображение из кэша в память приложения, например, в файл с именем 1.jpg:

```
function success(image_uri) {
    copyPhotoToFile('1.jpg', image_uri);
}
```

Функция copyPhotoToFile (рассматривается далее) копирует файл image_uri, предоставленный плагином, в файл 1.jpg, который уже находится в памяти (песочнице) приложения.

Если работа с камерой прошла неуспешно: например, отказались от какого-либо выбора фото – отснятого или из альбома, или произошла ошибка, то управление передаётся функции fail:

```
function fail(message) {
    Ext.Msg.alert('Error', "Failed: " + message);
}
```

Здесь показываем пользователю формулировку ошибки, предоставленную самим плагином в переменной message. Соответствующее окно с ошибкой реализуется с помощью метода alert библиотеки Ext JS.

В переменной options (она фигурирует выше в функции getPicture) задаются настроечные параметры, вот некоторые из них:

```
var options = {
    quality: 70, // качество фото
    ,destinationType: // где разместить результат
        navigator.camera.DestinationType.FILE_URI // в файле
    // камера или альбом
    ,sourceType:
        navigator.camera.PictureSourceType.CAMERA
        //( или PHOTOLIBRARY)
    // формат изображения
    ,encodingType: navigator.camera.EncodingType.JPEG
    ,correctOrientation: true
};
```

Файловая система устройства в проекте приложения Sencha Ext JS – Cordova

Базовый плагин cordova-plugin-file использует универсальную для всех поддерживаемых мобильных платформ концепцию манипулирования файловой системой. Приложения устройства владеют приватными директориями файлового хранилища, называемыми песочницами, в которых можно выделить поддиректории различные по назначению, по доступности (чтение, запись), по месту расположения (внутренняя или внешняя память) или по режиму сохранности файлов. Для каждой из таких частей плагин, который в приложении представляется объектом cordova.file, предусматривает соответствующую объект-ссылку, которая легко запоминается и является общей для различных мобильных платформ.

Корневая директория песочницы приложения представлена объектом cordova.file.applicationStorageDirectory. Директория, в которой приложение

установлено, представлена ссылкой `cordova.file.applicationDirectory`. В iOS и windows эта директория только для чтения (хотя её поддиректории `/Documents` в iOS и `/localState` в windows доступны и по записи).

Директория `cordova.file.dataDirectory` предназначена для произвольных данных приложения, доступна по чтению и по записи. В iOS эта директория не синхронизируется с iCloud. Если требуется синхронизация с облаками, то следует использовать `cordova.file.syncedDataDirectory` (iOS, Android, BlackBerry 10, windows). На Android можно обращаться к внешней памяти - `cordova.file.externalDataDirectory` (если есть). Наконец, своеобразный кэш устройства – **`cordova.file.externalCacheDirectory`**. Здесь целесообразно хранить файлы, которые приложение может легко воссоздать, поскольку операционная система может при необходимости (например, не хватает памяти) удалять файлы в этой директории. В то же время, нельзя полагаться на ОС и для экономии памяти лучше удалять их в приложении. Это далеко не полный список подобных точек входа, предусмотренных cordova.

Еще более универсальный подход предполагает использование `cdvfile`-протокола. Поясним его на примере. Пусть в директории данных приложения, то есть в `cordova.file.dataDirectory`, расположен файл с именем `1.jpg`. Полное имя этого файл может быть таким:

```
/var/mobile/Containers/Data/Application/46263B0B-4185-44C8-83E0-DC3B2C1F0432/Library/NoCloud/1.jpg
```

Здесь замысловатый номер – это некоторый идентификатор песочницы приложения. Понятно, что расположен файл в директории, которая не синхронизируется с облачным хранилищем (NoCloud). Пример взят с реального устройства iPhone. На других устройствах этот путь может быть устроен совершенно по-другому. В тоже время для большинства платформ путь к нашему файлу с помощью `cdvfile` может быть задан универсально и более компактно:

```
cdvfile://localhost/library-nosync/1.jpg
```

Более того, этот способ может быть использован в плагинах cordova, а также DOM-элементах документа, например, для `src` в `img`.

Теперь можно обсудить копирование отснятого изображения из кэша камеры в файл `1.jpg`. Напомним, что для этого у нас была предусмотрена функция `copyPhotoToFile()`, которой передавались два параметра имя файла – `filename` (например, `1.jpg`) и имя файла с фото в кэше камеры - `uri`. Во-первых, просим систему предоставить доступ к директории `cordova.file.dataDirectory` с помощью функции `window.resolveLocalFileSystemURL()`. Функция принимает три параметра – это запрашиваемая директория, функция-обработчик успешного исхода и функция, которая вызывается в случае ошибки:

```
function copyPhotoToFile(filename, uri) {
    window.resolveLocalFileSystemURL(cordova.file.dataDirectory,
        function(directoryEntry) { // успех
            // здесь с помощью directoryEntry
            // получим доступ к файлу с именем filename
        }
    );
}
```

```

    },
    function(e) { // ошибка - не получили доступ к директории
        console.log('resolveLocalFileSystemURL Error');
    }
);
}

```

В случае успеха получаем доступ к директории посредством объекта `directoryEntry` и аналогично получаем доступ к файлу с заданным именем (если его нет – то он будет создан):

```

directoryEntry.getFile(filename, { create: true },
    function(fileEntry) { // успех
        // здесь копируем фото из кэша камеры в файл filename - 1.jpg
    },
    function(e) { // ошибка
        console.log('getFile Error');
    }
);

```

Копирование файла `uri` с фото из кэша камеры в файл, представленный объектом `fileEntry`, реализуем с помощью плагина `cordova-file-transfer`, который работает не только в сети, но и в рамках локальной памяти устройства:

```

var fileTransfer = new FileTransfer();
var fileURL = fileEntry.toURL();
fileTransfer.download(uri, fileURL,
    function(entry) { // успех
        // здесь требуется установить src свойство компонента img
        // (класса Ext.Img) на файл с изображением
        // img.setSrc( fileURL + '?' + Date.now() );
        // также можно очистить кэш камеры:
        // navigator.camera.cleanup(onSuccess, onFail);
    },
    function(error) { // ошибка
        //console.log('fileTransfer Error');
    }, null, {}
);

```

Поясним представленный программный код:

- Предполагается, что в переменной `img` содержится ссылка на компонент `Ext.Img` библиотеки `Sencha Ext JS` (напомним, что именно этот компонент является наиболее подходящим для представления фото).
- В переменной `fileURL` содержится полный путь к скопированному изображению, который получен с помощью метода `toURL()` объекта `fileEntry`, посредством которого осуществляется доступ к файлу. Например, эта переменная может содержать следующий URL:

```

file:///var/mobile/Containers/Data/Application/46263B0B-4185-44C8-83E0-DC3B2C1F0432/Library/NoCloud/1.jpg

```

Вместо `toURL()` можно было бы воспользоваться методом `toInternalURL()`, который использует протокол `cdvfile`, в результате, получили бы:

```
cdvfile://localhost/library-nosync/1.jpg
```

- Для того, чтобы webview действительно перегружало изображение в компоненте `img` используется искусственный приём – добавляется в строку запроса текущее время в миллисекундах – `Date.now()`.

- После копирования изображения в песочницу следует зачистить кэш камеры:

```
navigator.camera.cleanup(onSuccess, onFail);  
function onSuccess() {  
    //console.log("Camera cleanup success");  
}  
function onFail(message) {  
    //console.log('Failed because: ' + message);  
}
```

Описанный процесс от создания фото до получения его копии в песочнице приложения схематично представлен на рис. 3.

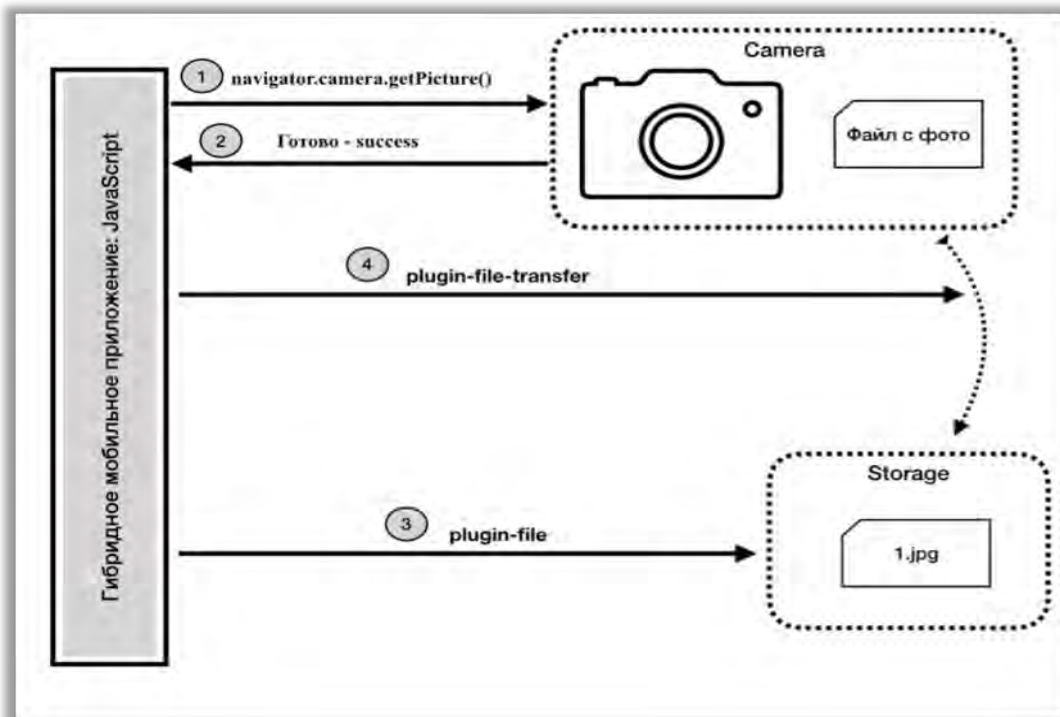


Рис. 3. Процессы взаимодействия приложения с плагинами от создания фото до его копирования в песочницу приложения. Номерами отмечена последовательность выполняемых операций

Загрузка изображения на сервер

Загрузка фото на web-сервер может быть реализована с помощью плагина `cordova-plugin-file-transfer`. Этот плагин достаточно универсален – мы его использовали для копирования фото из кэша камеры в песочницу приложения, а теперь используем для передачи на сервер посредством `http`-протокола. Серверная часть – стандартный код на `php` – уже приводился выше. Здесь обсудим соответствующую клиентскую часть мобильного приложения.

Прежде чем осуществлять загрузку, проверяем состояние сети. Сделать это можно с помощью плагина cordova-plugin-network-information следующим образом:

```
function checkConnection() {
  if(
    navigator.connection.type == Connection.NONE ||
    navigator.connection.type == Connection.UNKNOWN
  ) {
    Ext.Msg.alert('No connection',
                  'Connection is required!');
    return false;
  }
  return true;
}
```

В приведённом выше фрагменте кода плагин представлен объектами navigator.connection и Connection. Проверка сетевого доступа довольно проста и не требует комментариев.

Загрузку реализуем следующим образом:

```
function upload() {
  if( !checkConnection() ) return; // проверка сети
  var fileURL = "cdvfile://localhost/library-nosync/1.jpg",
  var options = new FileUploadOptions();
  options.fileKey = "file";
  options.fileName = fileURL.substr(fileURL.lastIndexOf('/') +
1);
  options.mimeType = "image/jpeg";
  var ft = new FileTransfer();
  ft.upload(fileURL, 'http://server/upload.php',
  function() {
    Ext.Msg.alert("Загружено!",
                  "Фотка ушла на сервер! Смотрим здесь:
http://server/1.jpg");
  },
  function(e) {
    Ext.Msg.alert("An error has occurred",
                  "Code = " + e.code +
                  "<br/>upload error source " + error.source +
                  "<br/>upload error target " + error.target);
  }, options);
}
```

В данном фрагменте кода плагин представлен объектом FileTransfer. Загрузка осуществляется с помощью метода upload(). Этот метод – принимает следующие параметры:

- fileURL – файла, подлежащий загрузке, для его указания вполне сойдётся cdvfile-протокол - cdvfile://localhost/library-nosync/1.jpg.
- URL принимающей файл серверной части - http://server/upload.php.
- Функция, которая вызывается в случае успеха - здесь можно просто проинформировать пользователя об успешной загрузке фото.

- Функция, которая вызывается в случае, если загрузка не прошла – выводим подробную информацию о причинах неудачи.
- Опции загрузки – имя файла на сервере, тип MIME и прочее.

Серверная обработка изображения

Обработку изображения на сервере можно организовать по-разному. Это может быть скрипт на php, который, например, использует известную графическую библиотеку GD. Это может быть программа на Си (если требуется максимальное быстродействие) и тогда мы обращаемся к этой программе посредством CGI (Common Gateway Interface). Для программ на Си существует многочисленные графические библиотеки в зависимости от платформы сервера и среды разработки.

Продемонстрируем обращение к серверной обработке, реализованной в виде скрипта на php (например, process.php), из гибридного приложения sencha-cordova. Предполагается, что для представления результатов обработки используется всё тот же компонент Ext.Image, который уже упоминался ранее. Пусть ссылка на этот объект содержится в переменной resholder, тогда достаточно настроить свойство src этого компонента на URL нашего сервера:

```
function showResult() {
    var res_uri = 'http://server/process.php' + '?' + Date.now();
    resholder.setSrc(res_uri);
}
```

Для тестирования вполне сойдется следующий простой скрипт, который подхватывает загруженный файл 1.jpg, делает на нём надпись "о б р а б о т а н о" и отправляет результат клиенту – мобильному приложению. С помощью GD эти действия выполняются следующим образом:

```
<?php
    $im = imagecreatefromjpeg('1.jpg');
    $red = imagecolorallocate ($im, 255, 0, 0);
    imagettftext ($im, 150, -45, 300, 300, $red, "arialbd.ttf",
        "О Б Р А Б О Т А Н
0");
    header ("Content-type: image/jpeg");
    imagejpeg ($im);
    imagedestroy ($im);
?>
```

Более полезными примерами являются алгоритмы определения объёма круглого леса по графическому изображению торцов брёвен (рис.4-7) [4,5]. Алгоритм реализован на Си и может быть доступен мобильному приложению через интерфейс CGI.



Рис. 4. Тестовая группа связанных областей из нескольких брёвен (представленных слева) и их серверная обработка (представлена справа)



Рис.5. Фотография графического объекта (представлена слева) и её серверная обработка (представлена справа)

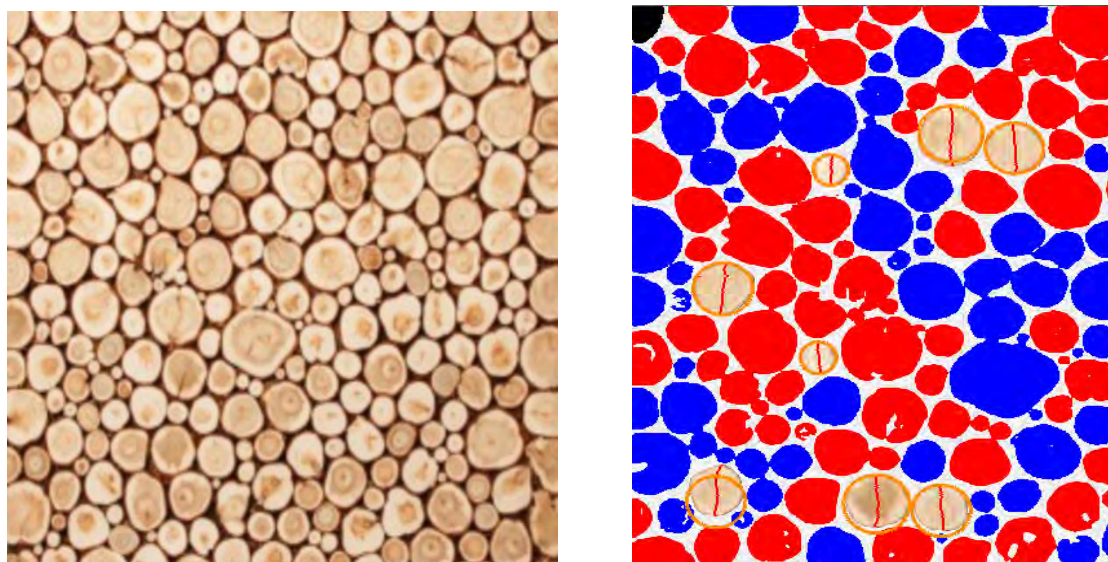


Рис.6. Фотография графического объекта (представлена слева) и её серверная обработка (представлена справа)

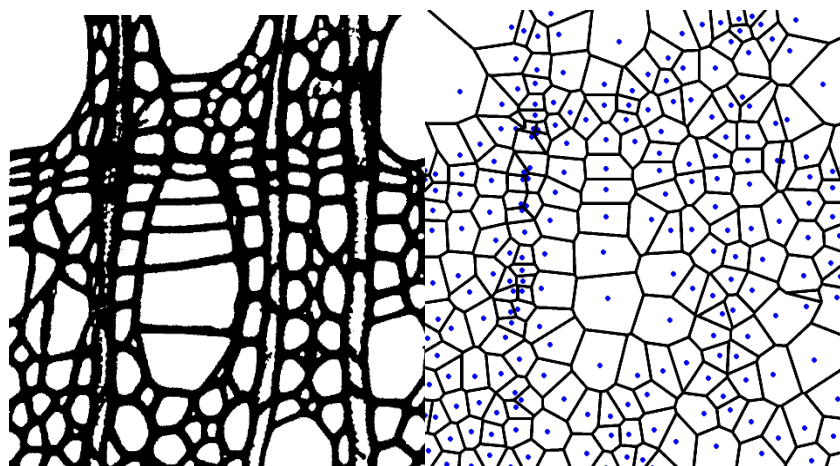


Рис.7. Пример моделирования древесных структур многоугольниками Вороного

Заключение

Обработка графики, как правило, оказывается достаточно затратной в плане основных вычислительных ресурсов таких, как процессорное время и оперативная память. Тем более, если требуются оперативные решения в процессе интерактивного взаимодействия пользователя с приложением.

В то же время мобильные устройства, которые, конечно, в настоящее время становятся всё более мощными (эта тенденция, по-видимому, сохранится) не обладают всё-таки достаточной мощностью. Особенно это касается гибридных приложений, которые воспроизводят свой интерфейс опосредованно через встроенный `webView`, который фактически является интерпретатором языков JavaScript, HTML и CSS и не может являться платформой для реализации затратных графических алгоритмов.

Решением может быть реализация графических алгоритмов:

- в виде плагинов, которые пишутся на нативных языках и непосредственно взаимодействуют с операционной системой мобильного устройства. Таких решений достаточно много, например, для распознавания штрих-кодов и тому подобное. В то же время оригинальные алгоритмы требуют реализации;
- в виде кода `NativeScript`, который работает непосредственно с API системы, но тогда приложение уже не будет гибридным;
- наконец, в виде программ на сервере. Это могут быть скрипты на PHP, которые подключают внешние графические библиотеки типа GD, или программы на C, C++, которые могут взаимодействовать с мобильным приложением, например, через CGI.

Именно последнему решению посвящена настоящая статья. Обсуждаются технологические и программные основы гибридного приложения построенного на основе Cordova и фреймворка Sencha Ext JS, реализующего обработку графики на стороне web-сервера.

Библиографический список

1. Васильев Н.П. Гибридные технологии разработки приложений для мобильных платформ // Информационные системы и технологии: теория и практика: сборник научных трудов – СПб.: СПбГЛТУ, 2017. №9, с.12-21.
2. Васильев Н.П. Универсальные технологии разработки мобильных приложений // Информационные системы и технологии: теория и практика: сборник научных трудов – СПб.: СПбГЛТУ, 2017. №10, ч.1, с.23-30.
3. Anand Daylan. Ext JS 6 By Example, Copyright © 2015 Packt Publishing, 2015. – 201с.
4. Лушкин Н.В., Васильев Н.П. Моделирование древесных структур многоугольниками Вороного (ячейками Дирихле)//Информационные системы и технологии: теория и практика: сборник научных трудов – СПб.: СПбГЛТУ, 2017.№9, с.43-53.
5. Лушкин Н.В., Копейкин М. Способ определения качества лесных объектов по графическим изображениям// Информационные системы и технологии: теория и практика: сборник научных трудов – СПб.: СПбГЛТУ, 2015. №7, с.41-46.

К. С. Веретенникова студентка
СПбГЛТУ им.С.М.Кирова
v.ksusha2@gmail.com

Ю.А. Жук кандидат педагогических наук, доцент
Кафедра трансформационных систем и технологий
СПбГЛТУ им.С.М.Кирова
zhuk_yua@mail.ru

ИСПОЛЬЗОВАНИЕ НЕЙРОННЫХ СЕТЕЙ ДЛЯ РАСПОЗНАВАНИЯ РАСТЕНИЙ

Введение

Растения являются одними из важнейших элементов в большей части экосистем и пищевых цепочек. Вопросами распознавания и идентификации занимались давно, применяя различные способы и методы классификации. В связи с массовым внедрением информационных технологий во все отрасли человеческих знаний, вопросы автоматизации в области идентификации биологических объектов тоже получил свое развитие. Попытки использования алгоритмов машинного обучения для распознавания объектов, в том числе и растений, предпринимали многие исследователи и ученые. Следует отметить, что распознавание растений, их классификация является сложной проблемой для компьютерного зрения, из-за множества неправильных форм и высокой изменчивости, даже в пределах одного класса. Идентификация растений может проводиться по их различным частям, органам, таким как плоды, кора, цветы, листья. Наибольшее число методов, которые использовались для решения данной проблемы, относится к распознаванию видов, классов, отдельных экземпляров растений по листьям. Рассмотрим наиболее популярные подходы в данной области, которые позволили получить эффективные модели и алгоритмы идентификации биологических объектов.

Свёрточные нейронные сети (CNN) являются наиболее распространенным решением для задач распознавания изображений и их классификации на широком диапазоне данных, таких как MNIST [7]. К особенностям такого типа нейронных сетей относится поэтапная обработка изображения. В качестве примера можно привести следующий: из изображения 200x200 сеть считывает квадрат размера 20x20, затем сдвигается и считывает новый квадрат. В последствие выходные данные передаются через сверточные слои, имеющие с глубиной свойство сжиматься. [8]

В 2015 году Ч. Чжанг и П. Чжоу [12] опубликовали свою работу, посвященную распознаванию растений по листьям, в основу которого был положен метод CNN. Для обучения нейронной сети был использован набор данных Flavia [14,11], содержащий изображения 1907 листьев 32 видов в высоком разрешении. Сверточная нейронная сеть представляет собой семь слоев, которые используют для увеличения количества данных для распознавания преобразования форм (таких как вращение, контрастность и т.п.) на этапе

предобработки. Ученые обнаружили, что такой подход позволяет объединить удаление лишних характеристик, данных, т.е. предобработку и процесс классификации в единый алгоритм, при этом сложность модели обучения не увеличивается, в то время как ее точность не ухудшается. Авторы работы указывают, что результаты экспериментальных исследований их алгоритма позволили классифицировать и идентифицировать растения 34 видов с точностью 94.6%.

В 2016 году Х. Чжоу и Ч. Ян опубликовали еще одну работу [13], посвященную классификации. Обучение нейронной сети проводили на множестве данных Leafsnap [15,5], включающем в себя 184 вида деревьев, произрастающих на территории США. У изображений в указанном множестве данных нет одинакового фона, поэтому ученые делали предобработку, состоящую из трех шагов: перевод из цветного спектра в градации серого; отделение переднего плана от фона; очистка фона. Для того, чтобы уменьшить влияние отклонений и искажений, которые возникают при съемке листьев вручную, авторы предприняли дополнительные шаги для предобработки, которая заключалась в выравнивании всех изображений листьев в вертикальное положение. Для уменьшения вычислительных затрат, накладывалось ограничение по размеру и составляло 50*50. Модель сверточной сети, которая была представлена в данной работе, базируется на LeNet [6], состоящей из 5 основных частей: 2 слоев свертки, 2 слоев подвыборки и одного многослойного перцептрона (MLP). В итоге ученым удалось добиться точности классификации для первоначального изображения 87,57%, а для изображения, которое прошло дополнительную предобработку (выравнивание), соответственно 91,36%.

Дальнейшие исследования в этом направлении провели К. Вик и Ф. Пуппе [10] в 2017 году. В своей работе они не ограничивались одним множеством данных, а использовали как Flavia, так и Foliage [4]. Авторы применяют предварительную обработку входных данных, предназначенных для обучения сети, для их стандартизации, когда каждое изображение преобразуется в размер 350x350 пикселей. При этом, размер входного слоя составляет 300x300x3 пикселя, где 3 – количество цветов. После обучения 9-слойной CNN, включающей в себя четыре слоя свертки, алгоритм дает на выходе точность классификации 99%.

Вероятностные нейронные сети (PNN) - это один из видов байесовых сетей, в которых вероятность принадлежности классам оценивается посредством аппроксимации ядер. PNN-сети широко используются в задачах классификации и распознавания образов. Недостатком подобной сети является ее размер и высокие требования к объему памяти, из-за чего сеть может медленно работать. [2,9]

Одной из наиболее интересных является работа А. Кадира, Л.Е. Нагрухо и др.[3]. В работе предложена система, которая была разработана с использованием PNN, в качестве классификатора. В классификационной модели заложен и описан ряд признаков, особенностей, характеристик по которым определяется лист растения. В дальнейшем используется статистиче-

ский метод анализа главных компонент (РСА) [8], суть которого заключается в уменьшении размерности данных. Для обучения системы было использовано множество данных Foliage, включающее в себя 60 видов растений, а также множество данных Flavias с 32 видами. В данной работе РСА играет ключевую роль, т.к. эксперименты показывают, что использование метода главных компонент способно увеличить точность классификации с 93% до 95%. Аналогичная точность была достигнута даже тогда, когда количество объектов уменьшилось с 64 до 25 объектов.

Заключение

Автоматизация распознавания растений является актуальным направлением исследований. Развитие и совершенствование методов и алгоритмов искусственного интеллекта, позволяет разрабатывать более эффективные модели и системы для идентификации объектов, в том числе биологических.

Следует отметить, что полная идентификация растений невозможна лишь созданием специализированных множеств, баз данных и разработкой систем идентификации, поскольку только лишь на территории России насчитывается более 11 тысяч видов сосудистых растений [1]. Цель этой работы показать, что разработка эффективных методов распознавания растений позволяет показать современные альтернативы классическому подходу к классификации биологических объектов.

Главными направлениями дальнейших разработок в сфере распознавания растений, на наш взгляд, является повышение точности и скорости классификации и идентификации растений, прежде всего в полевых условиях.

Библиографический список

1. Физическая география России и СССР. Почвы, растительность и животный мир [Электронный ресурс] // [URL:http://ecosystema.ru/08nature/world/geoussr/2-4-3.htm](http://ecosystema.ru/08nature/world/geoussr/2-4-3.htm).
2. StatSoft, Электронный учебник по статистике [Электронный ресурс] // [URL:http://statsoft.ru/home/textbook/glossary/gloss_v.html](http://statsoft.ru/home/textbook/glossary/gloss_v.html)
3. Kadir A., Nugroho L. E., Susanto A., Santosa P. I. Performance Improvement of Leaf Identification System Using Principal Component Analysis. // International Journal of Advanced Science and Technology. 2012. Vol. 44. P. 113-124.
4. Kadir A., Nugroho L. E., Susanto A., Santosa P. I.. Experiments of Zernike Moments for Leaf Identification // Journal of Theoretical and Applied Information Technology (JATIT); Vol 41, No. 1, 2012, pp. 82-93.
5. Kumar N., Belhumeur P. N. et al.. Leafsnap: A Computer Vision System for Automatic Plant Species Identification" // Proceedings of the 12th European Conference on Computer Vision (ECCV), October 2012.
6. LeCun Y, Bottou L, Bengio Y, Haffner P. Gradient-based learning applied to document recognition // Proceedings of the IEEE, vol. 86, no. 11, pp. 2278-2324, Nov. 1998.
7. LeCun, Y., Cortes, C.: The MNIST database of handwritten digits // Journal of Intelligent Learning Systems and Applications, Vol.9 No.2, May 12, 2017.

8. Shlens J., A Tutorial on Principal Component Analysis [Электронный ресурс] URL: <https://arxiv.org/pdf/1404.1100.pdf>
9. Specht, D. F. (1967-06-01). Generation of Polynomial Discriminant Functions for Pattern Recognition // IEEE Transactions on Electronic Computers. EC-16 (3): 308–319.
10. Wick, C., Puppe, F.. Leaf Identification Using a Deep Convolutional Neural Network. (2017) [Электронный ресурс] URL: <http://arxiv.org/abs/1712.00967>
11. Wu S.G., Baoet F.S. al. A Leaf Recognition Algorithm for Plant classification Using Probabilistic Neural Network//IEEE 7th International Symposium on Signal Processing and Information Technology, Dec. 2007, Cario, Egypt.
12. Zhang C., Zhou P., Li C., et al. A Convolutional Neural Network for Leaves Recognition Using Data Augmentation // IEEE International Conference on Computer and Information Technology, 2015
13. Zhou H., Yan C. and Huang H. Tree Species Identification Based on Convolutional Neural Networks // 2016 8th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC), Hangzhou, 2016, pp. 103-106.
14. База изображений растений Flavia [Электронный ресурс] // URL: <http://flavia.sourceforge.net/>
15. База изображений растений Leafsnap [Электронный ресурс] // URL: <http://leafsnap.com/dataset/>

С. П. Васильев, магистр
Н.Г. Полетаева, кандидат технических наук, доцент
Кафедра информационных систем и технологий
СПбГЛТУ им. С.М.Кирова
Spur49@mail.ru

ПРИМЕНЕНИЕ МЕТОДОВ МАШИННОГО ОБУЧЕНИЯ В ЗАДАЧАХ ОПТИМИЗАЦИИ

Искусственный интеллект (ИИ) - популярный, но определенный термин, который обычно указывает на алгоритмические решения сложных проблем людьми. Как показано на рис. 1, машинное обучение является основным строительным блоком для ИИ. Глубокое обучение является строгим подмножеством машинного обучения, относящимся к определенному классу многослойных моделей, которые используют слои более простых статистических компонентов для изучения представлений данных. Нейронная сеть является более общим термином для такого типа многоуровневой архитектуры статистического обучения.

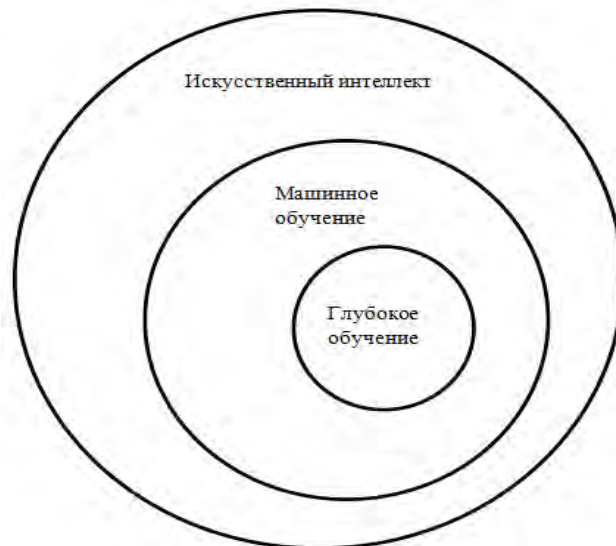


Рис. 1. Искусственный интеллект, связанный с машинным обучением и глубоким обучением

Многие прикладные экстремальные задачи могут быть сформулированы в форме той или иной задачи оптимизации на графах. Возможность использования нейронных сетей с обратной связью для решения задачи коммивояжера впервые была установлена Дж. Хопфилдом и Д. Танком [3]. Сеть Хопфилда состоит из единственного слоя нейронов $\{N_e \ i\}$, где $i = 1, \dots, n$, число n является одновременно числом входов и выходов сети. Аксон каждого нейрона связан с дендритами остальных нейронов через *синапсы*, образуя обратную связь. Нейронная сеть Хопфилда изображена на рис. 2 с помощью специальной двухслойной структуры, в которой нулевой слой не выполняет вычислительных функций, а лишь собирает выходы сети и направляет их обратно на входы. Каждый j -й нейрон первого слоя вычисляет взвешенную сумму своих входов (с помощью весов w_{ij} , где $i, j = 1, \dots, n$, а также веса смещения w_{0j}), генерируя сигнал NET_j , который затем с помощью нелинейной функции активации F преобразуется в сигнал OUT_j , где $j = 1, \dots, n$.

В качестве функции активации F можно использовать пороговую функцию. Если взвешенная сумма выходов с других нейронов больше порога T_j - выход j -го нейрона будет равен единице, в противном случае он равен нулю или остается без изменения, если сигнал NET_j равен порогу T_j . Таким образом, должны выполняться следующие условия:

$$OUT_j = 1, \text{ если } NET_j > T_j \quad (1)$$

$$OUT_j = 0, \text{ если } NET_j < T_j, \quad (2)$$

$$OUT_j \text{ не изменяется, если } NET_j = T_j, \quad (3)$$

$$\text{где } NET_j = \sum_{i \neq j}^n w_{ij} OUT_i + w_{0j}. \quad (4)$$

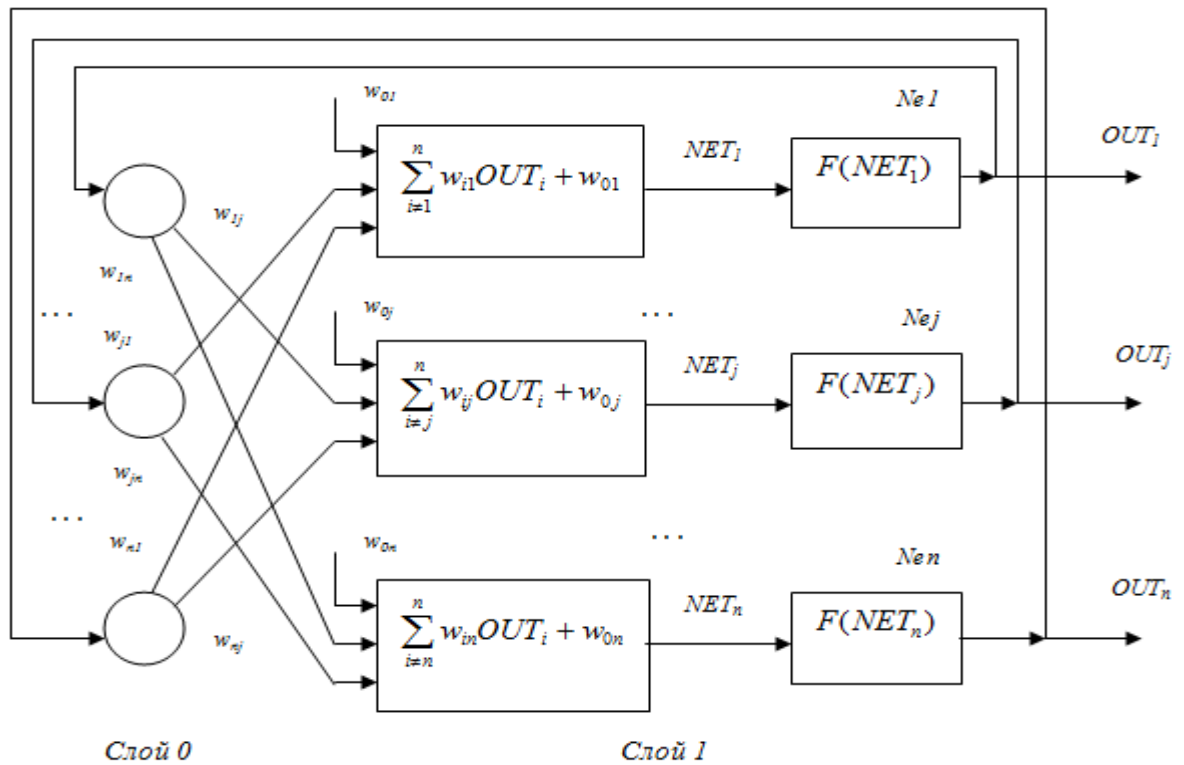


Рис. 2. Детализированная нейронная сеть Хопфилда

Рекуррентные нейронные сети, для которых возможно получить стабилизирующиеся к определенному значению выходы, называются устойчивыми, а если выходы сети не стабильны, то – неустойчивыми.

Сеть Хопфилда гарантированно будет устойчивой при выполнении следующих условий:

- 1) матрица весов симметрична: $w_{ij} = w_{ji}$,
- 2) $w_{ii} = 0$, для всех i .

Для доказательства устойчивости сети Хопфилда рассмотрим рекуррентную нейронную сеть как динамическую систему, имеющую некоторое энергетическое состояние. Если описать энергию системы функцией E и показать, что E при любом изменении состояния всегда убывает или не изменяется, то система (нейронная сеть) будет по определению устойчивой. Такая функция, называемая функцией Ляпунова, для сетей с обратными связями может быть представлена следующим образом [2]:

$$E = -1/2 \sum_i \sum_j w_{ij} OUT_i OUT_j - \sum_j w_{0j} OUT_j + \sum_j T_j OUT_j . \quad (5)$$

Здесь w_{ij} – вес от выхода нейрона i к входу нейрона j , OUT_j – выход нейрона j , w_{0j} , – вес смещения нейрона j , T_j – порог нейрона j .

Изменение энергии δE , вызванное изменением состояния нейрона δOUT_j , можно представить как:

$$\delta E = \left[\sum_{i \neq j} (w_{ij} OUT_i) + w_{0j} - T_j \right] \delta OUT_j = -[NET_j - T_j] \delta OUT_j . \quad (6)$$

Возможны три варианта:

1) В случае если величина NET_j больше порога T_j , из условия (1) следует, что $\delta OUT_j \geq 0$, в итоге $\delta E \leq 0$, энергия E убывает или стабилизируется.

2) Допустим, что $NET_j < T_j$, тогда изменение выхода j -го нейрона согласно (2): $\delta OUT_j \leq 0$ и энергия E должна уменьшиться или остаться без изменения.

3) Если взвешенная сумма входов с других нейронов равна порогу $NET_j = T_j$, то в соответствии с (3): $\delta OUT_j = 0$ и энергия E остаётся без изменения.

Таким образом, любое изменение состояния нейрона либо уменьшит энергию, либо оставит её без изменения. Благодаря такому свойству, энергия, в конце концов, должна достигнуть минимума и прекратить изменения, гарантируя устойчивость сети.

В общем случае, большая часть рекуррентных нейронных сетей являются неустойчивыми. Неустойчивые сети не пригодны для практического применения, и поэтому не рассматриваются.

Рассмотрим постановку задачи коммивояжера с применением нейронной сети Хопфилда, позволяющую значительно увеличить скорость её решения. Пусть задан неориентированный граф $G(N, U)$ с множеством вершин N и множеством расстояний между вершинами U (рис. 3), расстояния между парами вершин u_{ij} образуют матрицу U размером $N \times N$. Решением задачи будем считать упорядоченное множество вершин, которое необходимо посетить, а функцией решения является суммарная длина маршрута.

$G(N, U)$

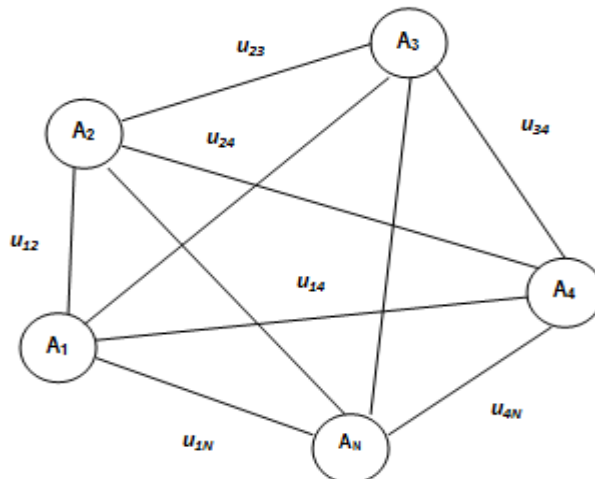


Рис 3. Граф задачи коммивояжера

Каждую вершину графа G представим строкой из N нейронов, выход только одного из которых равен единице, а выходы остальных равны нулю. В таком случае порядковый номер единичного нейрона укажет на номер вершины при обходе. Так как каждая вершина посещается только один раз, и в каждый момент посещается только одна вершина, то в каждой строке и каждом столбце окажется по одной единице (табл. 1).

Маршрут коммивояжера

Пункт	Порядок посещения				
	1	2	3	...	N
A ₁	1	0	0		0
A ₂	0	0	0		1
A ₃	0	1	0		0
...					
A _N	0	0	1	...	0

Рассмотрим сеть Хопфилда из $N \times N$ нейронов, в которой каждый нейрон имеет два индекса, соответствующих номеру вершины и порядковому номеру посещения этого пункта в маршруте. Например, нотация $OUT_{Xj} = 1$ означает, что пункт X был j -ым по порядку пунктом маршрута. В соответствии с табл. 1 в состоянии решения должны быть активны только N нейронов.

Введем функцию энергии сети, обеспечивающую выполнение требования на однократность посещения и охват всех городов:

$$E = \frac{C_1}{2} \sum_X \sum_i \sum_{i \neq j} OUT_{Xi} OUT_{Xj} + \frac{C_2}{2} \sum_i \sum_X \sum_{Y \neq X} OUT_{Xi} OUT_{Yi} + \frac{C_3}{2} \left[\left(\sum_X \sum_i OUT_{Xi} \right) - N \right]^2, \quad (7)$$

где C_1 , C_2 и C_3 – коэффициенты. Нетрудно видеть, что в выражении (7):

1. Первое слагаемое обеспечивает выполнение требования на посещение каждого города не более одного раза – равно нулю в том и только в том случае, если каждая строка содержит не более одной единицы.

2. Второе слагаемое обеспечивает выполнение требования на посещение только одного города в каждый момент времени – равно нулю в том и только в том случае, если каждый столбец (порядковый номер посещения) содержит не более одной единицы.

3. Третье слагаемое обеспечивает выполнение требования на посещение всех городов – равно нулю в том и только в том случае, если матрица содержит ровно N единиц.

Второе требование (предпочтение коротким маршрутам) удовлетворяется с помощью добавления следующего члена к функции энергии:

$$E_4 = \frac{C_4}{2} \sum_X \sum_{Y \neq X} \sum_i u_{XY} OUT_{Xi} (OUT_{Y,i+1} + OUT_{Y,i-1}), \quad (8)$$

Затем, необходимо установить соответствие между членами функции энергии в общем виде (5) и энергетической функцией задачи коммивояжера (7,8), то есть задать значения весов:

$$w_{Xi,Yi} = -C_1 \delta_{XY} (1 - \delta_{ij}) - C_2 \delta_{ij} (1 - \delta_{XY}) - C_3 - C_4 \cdot u_{XY} (\delta_{j,i+1} + \delta_{j,i-1}), \quad (9)$$

где δ_{ij} - символ Кронекера, принимающий значение 1, если $i=j$ и значение 0 в противном случае.

В качестве пороговой функции активации F нейрона удобно использовать функцию вида:

$$OUT = \frac{1}{2} \left[1 + \tanh \left(\frac{NET}{u_0} \right) \right], \quad (10)$$

где u_0 – пороговое значение смещения сети, а \tanh – гиперболическая тангенциальная функция. Данная функция обладает хорошей реализацией и наиболее приближена к пороговой функции.

После того, как исходные веса сети определены, можно, стартовав со случайного начального состояния (задав возмущение путём первоначальной генерации синаптических весов), проследить её эволюцию к устойчивой конфигурации, которая и даст решение задачи. Для этого необходимо предоставить сети возможность уменьшать свою функцию энергии путём движения в пространстве состояний в нужном направлении.

Критерием останова для сети Хопфилда является итерация, на которой состояния нейронов не изменяются, а функция энергии достигнет минимума. В этом случае необходимо остановить вычисления, а полученные значения выходов нейронов взять в качестве искомого маршрута коммивояжера.

Обучение с подкреплением. При обучении с подкреплением программный агент делает наблюдение и предпринимает действия внутри среды, получая в ответ награды. Его цель – научиться действовать так, чтобы довести до максимума ожидаемые долгосрочные награды [1].

Алгоритм, применяемый программным агентом для определения своих действий, называется политикой. Политикой может быть нейронная сеть, которая на входе принимает наблюдения и выдает действие, подлежащее выполнению. Подобно политике, жестко закодированной, эта нейронная сеть будет в качестве входа получать наблюдение и выдавать действие, подлежащее выполнению. Более точно, она оценит вероятность для каждого действия, чтобы выбрать действие случайным образом согласно оценочным вероятностям (рис. 4).

Одна из проблем обучения с подкреплением связана с тем, что для обучения агента необходимо иметь работающую среду. OpenAI Gym – комплект инструментов, который представляет широкое разнообразие имитированных сред (настольные игры, двумерные и трехмерные физические симуляции и т.д.), давая возможность обучать агентов, сравнивать их или разрабатывать новые алгоритмы.

Нейронную сеть, приведенную на рис. 4, можно обучать традиционным способом, сводя к минимуму перекрестную энтропию между оценочной и целевой вероятностью (обучение с учителем).

При обучении с подкреплением единственным руководством, которое получает агент, являются награды. Общая стратегия решения задачи заключается в оценке действия на основе суммы всех наград, которые поступили после него, обычно с применением дисконтной ставки γ на каждом шаге.

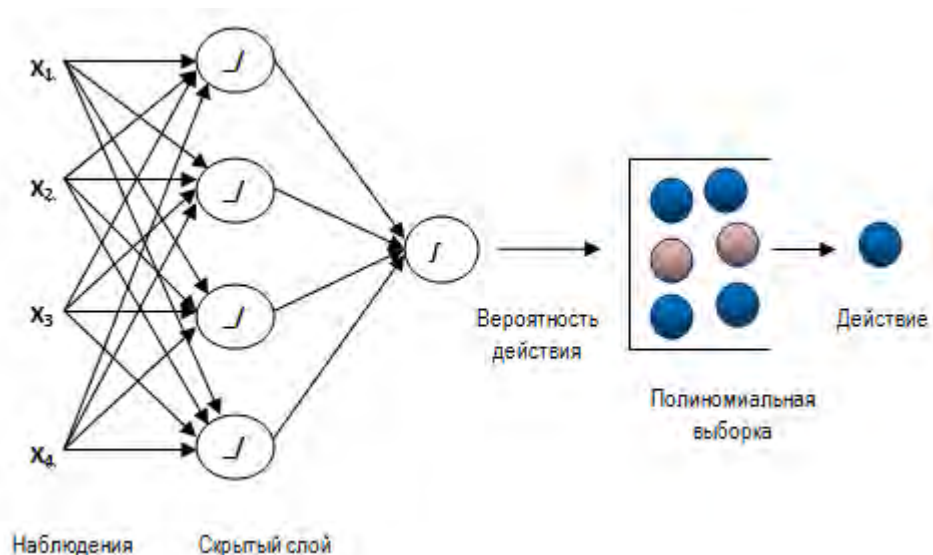


Рис 4. Политика в форме нейронной сети

Еще один подход предусматривает применение приемов оптимизации путем оценки градиентов наград относительно параметров политики и подстройки этих параметров, следуя градиенту в направлении более высоких наград. Такой подход называется градиентами политики (Policy Gradient – PG). Популярный класс алгоритмов PG, называемый алгоритмами REINFORCE (Reward Increment = nonnegative Factor × Offset Reinforcement × Characteristic Eligibility – прирост наград = неотрицательный множитель × подкрепление вознаграждения × характерный возможный выбор), был введен Р. Уильямсом.

Распространенный вариант алгоритма REINFORCE можно описать следующим образом:

- Дать возможность политике в форме нейронной сети сыграть в игру несколько раз и на каждом шаге вычислять градиенты без применения, которые сделают выбранное действие более вероятным.
- После прогона нескольких эпизодов подсчитать оценки каждого действия.
- Если оценка действия положительная – это означает, что действие было хорошим, и можно применить вычисленные ранее градиенты, чтобы сделать выбор данного действия в будущем еще более вероятным. Если же оценка действия отрицательная – это значит, что действие было плохим, и необходимо применить противоположные градиенты. Решение заключается в умножении каждого вектора-градиента на соответствующую оценку действия.
- Вычислить все результирующие векторы-градиенты и использовать их для выполнения шага градиентного спуска.

Марковские системы принятия решений. Стохастические процессы без памяти называются цепями Маркова. Марковский процесс имеет фиксированное число состояний и на каждом шаге случайным образом переходит из одного состояния в другое. Вероятность перехода из состояния s в состоя-

ние s' является фиксированной величиной и не зависит от прошлых состояний, а определяется только парой (s, s') .

Марковские процессы принятия решений (Markov Decision Process – MDP) впервые были описаны Р. Беллманом, они напоминают цепи Маркова, но с одним условием: на каждом шаге агент может выбирать одно из нескольких возможных действий, а вероятности переходов зависят от выбранного действия. Кроме того, определенные переходы между состояниями возвращают награду (положительную или отрицательную), и цель агента в том, чтобы отыскать политику, которая будет обеспечивать максимум награды с течением времени.

Способ оценки оптимальной ценности состояний для любого состояния s , предложенный Беллманом, представляет собой сумму всех будущих наград с учетом дисконтной ставки. Если агент действует оптимально, то для всех s применимо уравнение оптимальности Беллмана:

$$V^*(s) = \max_a \sum_{s'} T(s, a, s') \{R(s, a, s') + \gamma \cdot V^*(s')\}, \quad (11)$$

где $V^*(s)$ – оптимальная ценность состояния s ;

$T(s, a, s')$ – вероятность перехода из состояния s в состояние s' при условии, что агент выбрал действие a ;

$R(s, a, s')$ – награда, которую агент получает, когда переходит из состояния s в состояние s' при условии, что он выбрал действие a ;

γ – дисконтная ставка.

Так же можно точно оценить оптимальную ценность каждого возможного состояния: сначала все оценки ценностей состояний инициализируются нулями, после чего они многократно обновляются с использованием алгоритма итерации по ценностям:

$$V_{k+1}(s) \leftarrow \max_a \sum_{s'} T(s, a, s') \{R(s, a, s') + \gamma \cdot V^*(s')\}, \quad (12)$$

где $V_k(s)$ – оценочная ценность состояния s на k -той итерации алгоритма, что в свою очередь является примером динамического программирования, которое разбивает задачу на легко поддающиеся обработке подзадачи, допускающие итеративное решение.

Алгоритм Беллмана для оценки оптимальных ценностей пар «состояние-действие» (Q -ценностей) можно сформулировать следующим образом: оптимальная Q -ценность пары «состояние-действие» (s, a) , обозначаемая $Q^*(s, a)$, представляет собой сумму будущих наград с учетом дисконтной ставки, которые агент может ожидать в среднем после того, как он достигнет состояния s и выберет действие a при условии его оптимального поведения после этого действия.

Все оценки ценностей изначально инициализируются нулями, а затем обновляются с применением алгоритма итерации по ценностям для всех пар (s, a) :

$$Q_{k+1}(s, a) \leftarrow \sum_{s'} T(s, a, s') \{R(s, a, s') + \gamma \cdot \max_{a'} Q_k(s', a')\}. \quad (13)$$

После нахождения всех оптимальных Q -ценностей определение оптимальной политики $\pi^*(s)$ заключается в следующем:

$$\pi^*(s) = \underset{a}{\operatorname{arg\,max}} Q^*(s, a), \quad (14)$$

когда агент пребывает в состоянии s , он должен выбрать действие с наивысшей Q -ценностью для этого состояния.

Данный подход применим к ситуации дорожного движения для оптимизации маршрута передвижения мобильного агента, поскольку дорожная сеть представляет собой стохастическую среду. Оптимальность передвижения в такой среде во многом зависит не только от расстояния, которое необходимо преодолеть для достижения конечной цели, но также от многочисленных внешних факторов, в том числе от объема дорожного трафика, дорожных происшествий и т.д.

Модель угроз. Будем рассматривать противника, который намерен отслеживать мобильных агентов. Предположим, что злоумышленник имеет ограниченное число прослушивающих станций для расположения в сети. Поскольку дорожные перекрестки являются стратегическими точками сети, через которые проходят все мобильные агенты, считаем, что локальный противник располагает свои прослушивающие станции только в этих местах. Прослушивающие станции имеют достаточно большую зону покрытия, чтобы обнаружить мобильных агентов на входе и выходе перекрестка. Противник собирает идентификаторы на каждом перекрестке, где находится прослушивающая станция. На основе собранной информации он пытается отслеживать местоположение мобильного агента.

Для того чтобы преодолеть отслеживание противником, мобильные агенты могут использовать несколько уникальных идентификаторов, которые изменяются с течением времени. В целях предотвращения пространственного и временного соотношения местоположения агентов, смена идентификаторов происходит в специальных областях, называемых зонами смешивания. Зоны смешивания идентификаторов можно разделить на два типа: активные и пассивные. К первому относятся такие зоны смешивания, которые кроме смены псевдонима, запрашивают действия со стороны пользователя – мобильного агента, такие как выключение их приемопередатчиков или использование мобильного прокси. Ко второму типу относятся зоны смешивания, в которых мобильные агенты, пользуясь отсутствием покрытия противника, автоматически изменяют свой уникальный идентификатор без каких-либо дополнительных действий.

Выигрыш злоумышленника при отслеживании местоположения мобильного агента будет тем выше, чем больше будет размещено прослушивающих станций. Однако, размещение прослушивающих станций влечет за собой затраты c_s на каждую станцию.

Мобильные агенты в свою очередь хотят сохранить высокий уровень конфиденциальности своего местоположения. Но их механизмы обеспечения конфиденциальности, размещение зон смешивания, также требуют опреде-

лённых затрат. В связи с этим существует две основные стоимости размещения зон смешивания:

- первая – стоимость смены псевдонимов c_p^i , которая включает в себя расходы на получение нового псевдонима (в том случае, если агенты используют беспроводную сенсорную маршрутизацию, она будет также включать стоимость обновления таблиц маршрутизации);
- вторая – затраты на пересечение активной зоны смешивания c_q^i , которая также учитывает временную потерю доступа к службам, при использовании агентом периода молчания.

Первая стоимость присутствует в обоих случаях – активных и пассивных зонах смешивания, в то время как вторая существенна только для активных зон смешивания идентификаторов.

Допустим, что суммарная стоимость смены псевдонимов и нахождения в состоянии молчания при пересечении активной зоны смешивания пропорциональна нормированной интенсивности движения [4]:

$$c_p^i = \alpha \lambda_i, \quad (15)$$

$$c_q^i = \beta \lambda_i, \quad (16)$$

где λ_i – нормированная интенсивность движения на i -том перекрёстке [5], α и β – стоимости смены псевдонимов и пребывания в состоянии молчания для мобильного агента.

Кроме того, определим стоимость размещения активных зон смешивания идентификаторов следующим образом:

$$c_m^i = c_p^i + c_q^i. \quad (17)$$

Таким образом, противник должен расположить свои прослушивающие станции в стратегической манере, принимая во внимание защитную стратегию мобильных агентов. Игровое взаимодействие заключается в том, как мобильные агенты и злоумышленник максимизируют их собственные выигрыши.

В моделируемой конфликтной ситуации, основанной на теоретико-игровом взаимодействии, при расчете ценности состояний вводятся некоторые определённые правила в качестве исходных данных:

1. Ценность начального состояния считается максимальной и равна единице.
2. Вершина, являющаяся целью, также имеет максимальную ценность, и равна единице.
3. Ценность вершины, в которой располагается зона смешивания идентификаторов, также считаем максимальной и равной единице.
4. Ценность вершин предполагаемого размещения прослушивающих станций злоумышленника равна -1.

Ценность задаётся максимальной в перечисленных вершинах (1-3), поскольку в условиях стохастической среды в этих вершинах мобильные агенты абсолютно уверены, что не производится прослушивание. Вершины, в которых злоумышленник размещает прослушивающие станции, исходя из

принципа оптимальности, имеют наибольшее количество примыкающих дорог, и дорожный трафик там наиболее интенсивный. Поэтому при начальном подсчете ценности состояний, такие вершины обладают максимальной отрицательной ценностью.

Предположим, что a – действие, выполняемое в состоянии s (выбор пути), определяется следующим образом (рис. 5):

$$a = \begin{cases} 0.8 = const & \text{для выбранного направления движения} \\ 0.1 = const & \text{для всех остальных возможных вариантов} \end{cases}$$

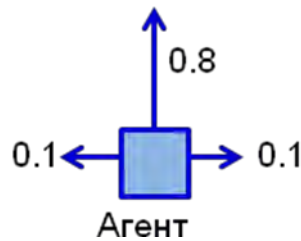


Рис. 5. Действие a , выполняемое в состоянии s

Применим рассмотренный алгоритм (11-13) к процессу MDP, представленному на рис. 6. Если предположить, что $R(s)$ – это кратковременное вознаграждение за пребывание в состоянии s , тогда ценность некоторого состояния равна сумме непосредственного вознаграждения за пребывание в этом состоянии и ожидаемой ценности следующего состояния, при условии, что агент выбирает оптимальное действие:

$$Q_{k+1}(s) = R(s) + \gamma \cdot \max_a \sum_{s'} T(s, a, s') \cdot Q_k(s'). \quad (18)$$

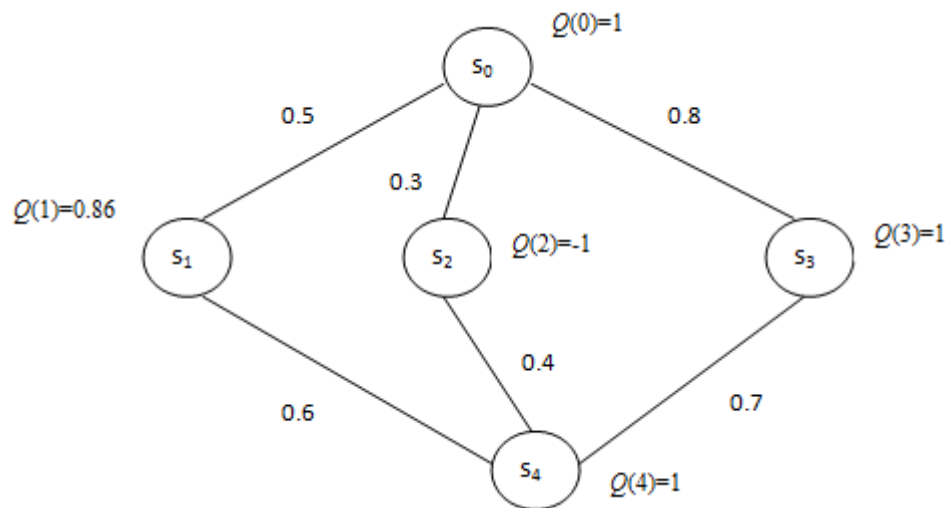


Рис. 6. Представление маршрута в связанном графе с учетом ценностей состояний

Дисконтная ставка γ (коэффициент обесценивания) описывает предпочтение агентом текущих вознаграждений перед будущими. Если дисконтная ставка γ равна нулю, то отдаленные вознаграждения являются малозначимыми, в рассматриваемом случае коэффициент γ равен единице, поскольку предполагается, что ценность текущего состояния определяется исходя из информации о ценности последующих состояний.

Применим метод Беллмана для связанного графа, представленного на рис. 7.

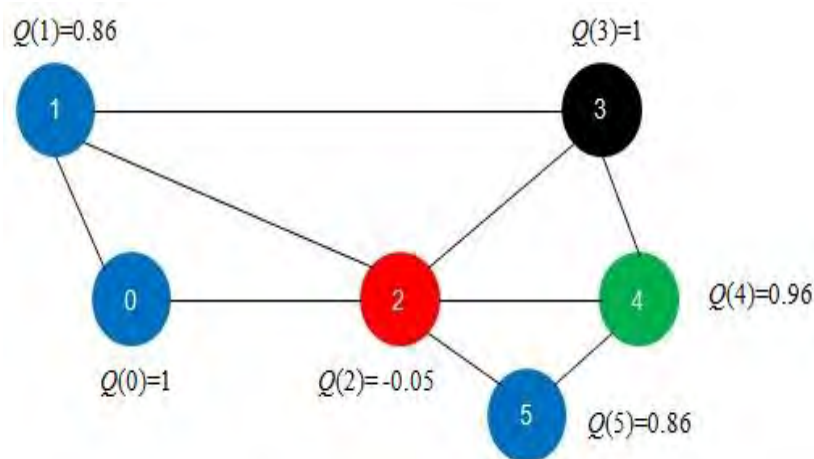


Рис. 7. Ценности состояний в связанном графе, при $\gamma = 1$ и $R(s) = -0.04$

Среднее время выполнения алгоритмов оптимизации приведено в табл. 2.

Т а б л и ц а 2

Сравнение разработанных алгоритмов по скорости

	Стенд: AMD A10-4600M 2,3 ГГц, 8 ГБ, Windows 7			
	Метод Беллмана	Алгоритм Дейкстры	Алгоритм Флойда-Уоршелла	Алгоритм Форда-Беллмана
Среднее время поиска, 1000 запусков	0,824 с	0,649 с	0,811 с	0,733 с

На представленном графе (рис. 7) классические алгоритмы поиска пошли по пути 0-2-3, так как расстояние между этими узлами минимальное. Это ведет к раскрытию агента, поскольку он проходит через перекресток, на котором с большой долей вероятности находится прослушивающая станция $Q(2) = -0.05$. Предлагаемый алгоритм Беллмана предложил маршрут 0-2-4-3, так как показатель ценности данного маршрута максимален, а агент, пройдя через прослушивающую станцию, обязательно посещает зону смешивания, чтобы поменять идентификатор и безопасно достичь своей цели в вершине 3. При этом реализация метода Беллмана не сильно уступает реализации алгоритма Дейкстры по скорости, находясь на том же уровне, что и два других алгоритма.

Библиографический список

1. Жерон О. Прикладное машинное обучение с помощью Scikit-Learn и TensorFlow: концепции, инструменты и техники для создания интеллектуальных систем: пер. с англ. – СПб.: ООО «Альфа-книга», 2018. – 688 с.

2. Хайкин С. Нейронные сети: полный курс, 2 издание: пер. с англ. – М.: Издательский дом «Вильямс», 2006. – 1104 с.

3. Hopfield J.J. Neural networks and physical systems with emergent collective computational abilities// Proc. Natl. Acad. Sci. USA, 1982, Vol. 79, pp. 2554-2558.

4. On the effectiveness of changing pseudonyms to provide location privacy in VANETs/ Levente Buttyan, Tamas Holczer, Istvan Vajda // 4th European Workshop on Security and Privacy in Ad-hoc and Sensor Networks, ESAS 2007, pp. 129-141.

5. Tracking Games in Mobile Networks / M. Humbert, M. H. Manshaei, J. Freudiger, J.-P. Hubaux // Decision and Game Theory for Security. First International Conference, GameSec, Berlin, Germany, 2010, vol. 6442, pp. 38-56.

М.Р. Вагизов кандидат технических наук, доцент
СПбГЛТУ им.С.М.Кирова
С. А. Тепляков студент

РАЗРАБОТКА WEB-КАРТОГРАФИЧЕСКОГО СЕРВИСА GIS-AERO

Развитие технологий программирования для геоинформационных систем открывают перспективные возможности разработки отраслевых картографических web-сервисов для разных видов деятельности. С развитием web-картографии, увеличением числа открытых программных библиотек, модулей и плагинов качественно повышается географическая визуализация и отображение отдельных элементов пространственной информации для человека. Вторым аспектом развития ГИС-технологий стоит выделить, улучшение интерфейса web-картографических программ, особое внимание на стадии проектирования должно уделяться способу взаимодействия программной среды с пользователем. Отмечено, что качественная реализация графического интерфейса (GUI) влияет не только на использование функционала программы, но и на общее количество пользователей, особенно, если web-картографический сервис будет являться общедоступным ресурсом. [1]. При проектировании web-приложений, в составе функций которого есть процедуры работы с картой и анализом информации, в основном, опираются на следующие подходы к разработке данных видов приложений:

1. Определение типа приложения.
2. Определение функциональных возможностей приложения.
3. Выбор среды и инструментов разработки приложения.
4. Структурно-логическое проектирование приложения.
5. Модульное программирование.
6. Тестирование приложения.

7. Запуск и отладка приложения, оптимизация процессов приложения.

На кафедре лесной таксации лесоустройства и геоинформационных систем, совместно с кафедрой информационных систем и технологий СПбГЛТУ, ведутся прикладные исследования в области проектирования и разработок новых геоинформационных систем и web-приложений для лесной и смежных отраслей. На базе открытых картографических материалов и свободно распространяемых плагинов с открытой лицензией начат проект по разработке картографического web-приложения GIS-Aero.

Web-картографический сервис не только предоставляет доступ к карте, но также является полезным инструментом отображения необходимых задач пользователю. Учитывая интерактивное взаимодействие пользователя с системой, принципы разработок, которого подробно описаны в предыдущих статьях [1], в концепции разработки сервиса GIS-Aero заложены определённые функции системы. Среди функций картографического web-сервиса можно выделить два направления отображения функционала работы.

1. Стандартизированные функции для работы с картой (поиск, навигация перемещение, создание полигонов).

2. Специальные функции (планирование маршрутов, информационные списки, отображение отдельных динамических слоёв).

В отличие от геоинформационных программ, где слой представляет собой фиксированный объект, в web-сервисах возможно изменение слоя в режиме on-line самим пользователем, не только внешнего оформления, но и динамического изменения поведения слоя, что придаёт большую интерактивность взаимодействию с картой. Для удобства отображения и быстроты взаимодействия с пользователем, загрузки пространственных объектов необходимо использовать функцию кэширования данных. В сервисе GIS-Aero реализована функция отображения пространственных объектов.

Спецификации картографического web-сервиса GIS-Aero. Основная цель сервиса предоставление информации планирования туристических маршрутов, просмотр действующих аэропортов, просмотр туристических мест и зон отдыха, с тематическим отображением объектов и сводной информации о выбранных объектах. Актуальность разработки данного сервиса связана с необходимостью повышения информированности пользователей о туристических объектах, как в пределах страны, так и за рубежом. Вторым модулем информационного обеспечения сервиса является отображение конкретно заданных параметров пользователем поиска определения мест отдыха с заданными параметрами (горнолыжные базы, дайвинг, рафтинг и др.) Перечисленные списки и их географическое отображение позволят упростить поиск пользователям требуемых мест на карте. Данные функции, сформированы в web-сервисе в виде выпадающего списка, при переходе на определённый элемент на карте автоматически отображаются все объекты по заданному критерию (рис.1).

Проведенный анализ в русскоязычном сегменте интернета показывает довольно небольшое количество качественно реализованных специальных web-сервисов, для планирования комплексных туристических путешествий,

поиска мест на карте, к примеру, особо охраняемых природных территорий (ООПТ).

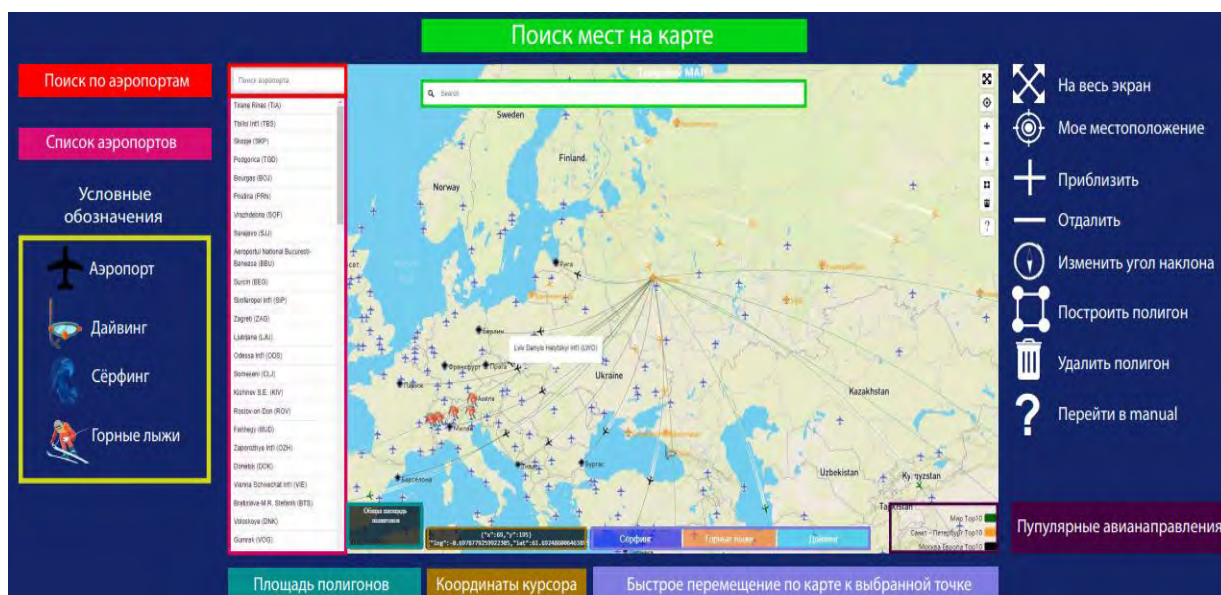


Рис. 1. Реализованный интерфейс сервиса GIS-Aero

Отдельным элементом реализована функция, поле **аэропорты** (рис.2). Слой аэропорты показывает все аэропорты, которые отображаются на экране, при перемещении карты список будет обновлен, если в списке отсутствуют поля, значит, в заданном масштабе карты отсутствуют открытые гражданские аэропорты, в таком случае стоит отдалить или переместить карту, до появления близлежащих аэропортов. В поле **аэропорты** реализована функция поиск по названию аэропорта, что позволяет быстро сориентироваться на отображаемой карте.

Техническая структура web-сервиса. Согласно спецификациям Open Geospatial Consortium (OGS) необходима обязательная реализация отображения следующих элементов на web-карте:

- 1) Изображения карты средствами (WMS)
- 2) Пространственные векторные объекты (WFS)
- 3) Растровые наборы данных (WCS)
- 4) Листы web-карты (WMTS)

Руководство данными спецификациями необходимо для предоставления карт в открытом формате, это позволит признать сервис на международном уровне в сети. Для расширения функциональных возможностей в web-приложении картографического сервиса может быть использована технология WebSocket для обмена сообщениями между браузером и веб-сервером в режиме реального времени (рис. 3) [2].

Основные инструменты реализации web-картографического сервиса. Базовая часть кода реализована на языке программирования JavaScript. Язык структурирования - HTML 5. За картографическую основу взяты открытые карты Open Street Maps. Частично, элементы отображения объектов и геопо-

зиционирование разработаны при помощи визуальных блоков средствами конструирования **mapbox**.

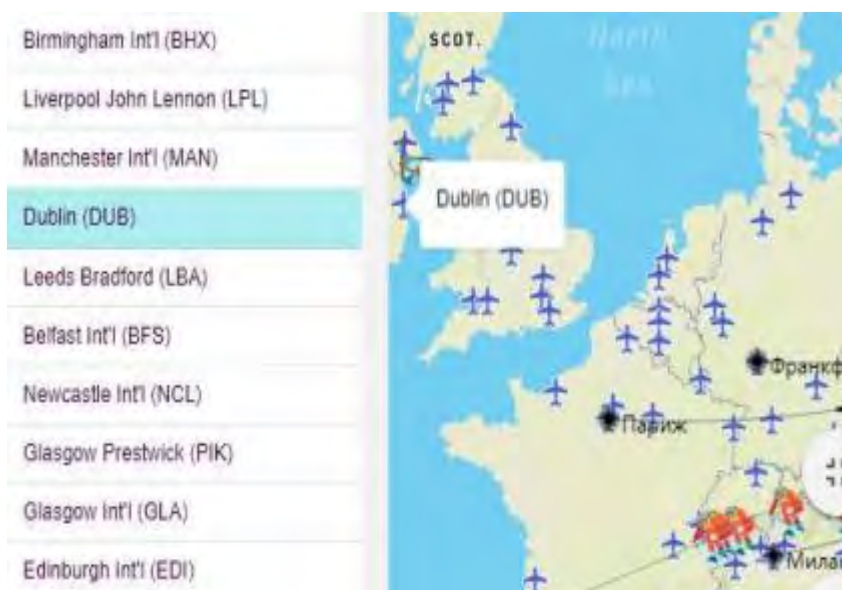


Рис.2. Аэропорты в сервисе GIS-Aero

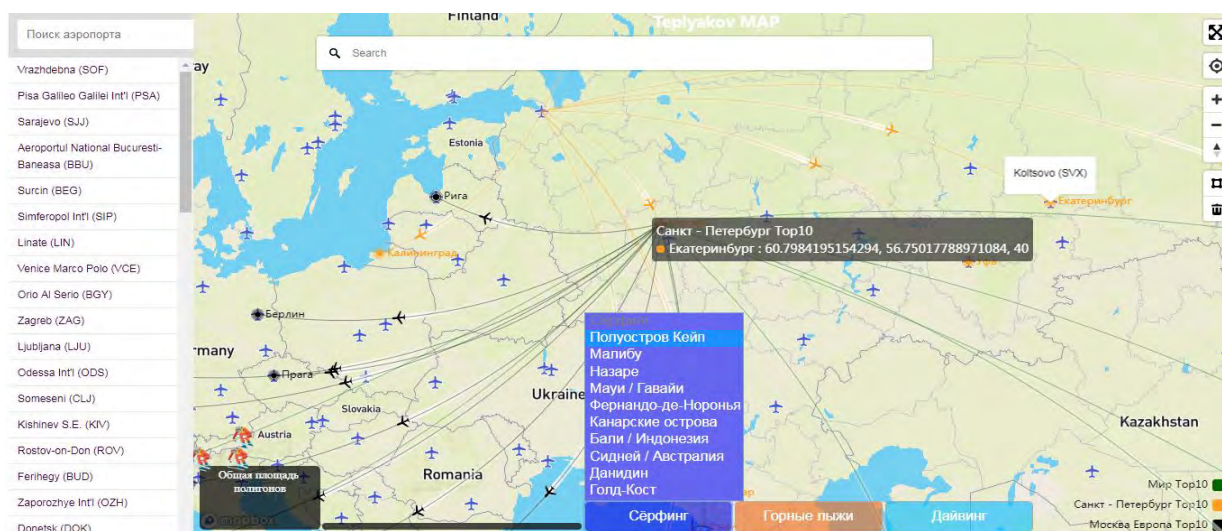


Рис. 3. Отображение интерфейса после запуска приложения в браузере

Проектирование и разработка новых, специализированных web-картографических сервисов является комплексной междисциплинарной программно-технологической задачей. На сегодняшний день технологии по созданию и разработке таких web-приложений совершенствуются, а конструирование на базе шаблонных модулей становится простым, интуитивно понятным. Развитие web-технологий определяет необходимость применения инфокоммуникационных технологий (ИКТ) во всех технологических процессах [3].

По данным социологических и профильных опросов, количество туристических маршрутов ежегодно увеличивается, так же увеличивается количе-

ство специализированных мест отдыха. Основная задача web-картографического сервиса GIS-Aero помощь в принятии решений на основе предоставляемой картографической информации. Предложить пользователю тематический выбор определённых мест в зависимости от выбранного направления, и визуализировать его положение на карте. Следующим шагом в развитии картографического сервиса становится предоставлении информации о стоимости и приблизительный расчёт затрат туристического сектора с графической визуализацией на карте. На данный момент, прототип сервиса подготовлен для регистрации в федеральном органе исполнительной власти по интеллектуальной собственности.

Библиографический список

1. Вагизов М.Р. Разработка интерактивных геоинформационных систем: принципы построения и конструирования системы. // Информационные системы и технологии: теория и практика. Сборник научных трудов научно-технической конференции института леса и природопользования. 2017. С. 21-27.
2. Хабаров С.П. Взаимодействия узлов сети по протоколу websocket. // Информационные системы и технологии: теория и практика Сборник научных трудов научно-технической конференции института леса и природопользования. 2017. С. 104-115.
3. Заяц А.М., Хабаров С.П. Настройка беспроводных соединений в системах мониторинга лесных территорий. // Леса России: политика, промышленность, наука, образование Материалы третьей международной научно-технической конференции. Под редакцией В.М. Гедьо. 2018. С. 80-83.

В.А. Горбачев, кандидат экономических наук, доцент
Кафедра информационных систем и технологий
СПбГЛТУ им. С.М. Кирова
ist@spbftu.ru

ПРОЕКТИРОВАНИЕ ПРОГРАММНОГО МОДУЛЯ «УЧЕТ ПОСЕЩАЕМОСТИ СТУДЕНТОВ»

Автоматизация управления учебным процессом в Санкт Петербургском лесотехническом университете базируется на комплексе программных средств (КПС), разработанных лабораторией математического моделирования и информационных систем (ММИС). Основными программными модулями, обеспечивающими автоматизацию учебного процесса комплекса, являются:

- Создание учебных планов ВПО – пакет «Планы»;
- Учёт контингента студентов – пакет «Деканат»;

- Расчёт нагрузки кафедр – пакет «Нагрузка ВУЗа;
- Учёт сессионной успеваемости – пакет «Электронные ведомости»;
- Студия визуального тестирования.

Эти программные средства используют данные накапливаемые и хранящиеся в базах данных (БД) «Деканат» и «Тест». Вопросы связанные с структурой БД «Деканат» рассмотрены в публикации «Автоматизация задач управления кафедрой» [1].

Функциональность этого КПС охватывает практически весь спектр информационных услуг в управлении учебным процессом и оценки его качества. Настройка на особенности ВУЗа осуществляются с помощью параметров (нормативы учебного процесса и т.д.). Однако, как и любое стандартное программное средство, оно не может учесть все тонкости управления в конкретном университете. В частности, появилась необходимость управления сессией, в задачи которой входят составление расписания экзаменов и назначение стипендий. Эти функции частично реализовывались в КПС лаборатории ММИС. Однако было признано более удобным разработать свой программный модуль для реализации этих функций. В результате была разработана программа «Управление сессией», характеристики которой были описаны в статье «Проектирование программного модуля «Управление сессией»» [2]. Аналогичные вопросы возникли и для нужд кафедр. В результате был разработан программный модуль «Управление кафедрой» [3].

В результате структурных изменений в управлении учебным процессом в СПбГЛТУ появилось отделение естественно-научного и гуманитарного образования, в функции которого вошло управление успеваемостью студентов 1 курса. Для повышения успеваемости студентов было принято решение автоматизировать учет посещаемости студентов 1 курса. В результате был разработан программный модуль «Учет посещаемости студентов», описание которого и предложено в данной статье.

Кратко постановка задачи для его проектирования выглядит следующим образом. В каждой группе старостами ведется журнал посещаемости, в котором отмечается посещаемость и активность студентов по всем предметам. Раз в неделю старосты сдают журналы в директорат. Данные из них должны заноситься в базу данных. А затем необходимо получать статистические оценки посещаемости по следующим критериям:

- Посещаемость студента;
- Средняя посещаемость группы;
- Посещаемость по преподавателям.

Информационной основой для реализации функций этого модуля явились следующие таблицы базы данных «Деканат»:

- Все_Ведомости
- Все_Группы
- Все_Студенты

В первой из них содержится множество полей характеризующих дисциплины изучаемые студентами в конкретном семестре учебного года. Для регистрации посещаемости потребуются только следующие из них: Год, Семестр, Курс, Код_Группы, Код_Факультета, Преподаватель, Дисциплина. Назначение полей понятно по их наименованиям. Таблица **Все_Группы** рассматривается как справочник, из которого используется наименование группы. Из таблицы **Все_Студенты** используются ФИО студентов относящихся к той или иной группе. Группа связанных полей этих таблиц определяет списки студентов изучающих некоторую дисциплину в выбранном семестре учебного года. А так же преподавателя ведущего эту дисциплину. Для регистрации посещаемости в базу данных была добавлена таблица АУППосещаемость со следующей структурой (рис. 1):

Имя поля	Тип данных
КодСтудента	Числовой
КодГруппы	Числовой
КодВедомости	Числовой
ДатаЗанятий	Дата/время
КодДисциплины	Числовой
Активность1	Текстовый
Активность2	Текстовый
Семестр	Числовой
Лек	Числовой
Практ	Числовой

Рис. 1. Структура таблицы АУППосещаемость

Помимо полей КодСтудента, КодГруппы, КодВедомости в таблицу добавлено поле ДатаЗанятий. Вместе они составляют ключ таблицы. Поле КодДисциплины в ключ не включается, так как между ним и полем КодВедомости установлена функциональная зависимость 1:1. Поле «Семестр» так же не включается в ключ таблицы, так как его роль берет на себя «КодВедомости». В полях «Активность1» и «Активность2» регистрируется посещаемость студентов на определенную дату. Предполагается, что это может быть либо лекция, либо практическое занятие. Два поля «Лек» и «Практ» конкретизируют, какой из типов занятий регистрировались в полях «Активность1» и «Активность2». Такая структура накладывает ограничения на регистрацию посещаемости. То есть на одну дату по одной дисциплине можно отметить посещаемость только для двух пар – либо лекций, либо практических занятий. Тип занятий отмечается в полях «Лек» и «Практ», тип поля логический. Эти же поля показывают, регистрировалась активность студента по какому либо виду занятий. Отметка активности студента имеет три значения:

- V – студент присутствовал на занятии;
- Н - студент отсутствовал на занятии;
- А – студент проявлял активность на занятии.

По умолчанию в колонках ставится V. Если в какой либо из них ставится Н или А, то считается, что произведена регистрация по выбранному предмету и дате и автоматически ставится «галочка» о регистрации посещаемости по лекциям или практикам. По мере накопления регистрационных данных по какой либо дисциплине появляется необходимость оценить посещаемость группы по выбранному предмету. Для этого в необходимо выбрать данные из таблицы АУППосещаемость с помощью перекрестного запроса следующего вида:

```

TRANSFORM First (IIf ([Лек]=1, [Активность1], [Активность2]))
AS Активность
SELECT TempПосещаемость.ФИО, TempПосещаемость.КодГруппы,
TempПосещаемость.КодВедомости,
Sum (IIf ([Лек]=1, IIf ([Активность1]='Н', 1, 0), IIf ([Активность2]
]='Н', 1, 0))) AS Отсутствовало
FROM TempПосещаемость
WHERE ((TempПосещаемость.Лек)=1)) OR
((TempПосещаемость.Практ)=1))
GROUP BY TempПосещаемость.ФИО, TempПосещаемость.КодГруппы,
TempПосещаемость.КодВедомости
PIVOT TempПосещаемость.ДатаЗанятий;


```

Данный запрос использует данные таблицы TempПосещаемость, в которую копируются сведения о посещаемости группы по выбранному предмету из таблицы АУППосещаемость. Это сделано для того, чтобы ускорить исполнение запроса, так как если использовать исходную таблицу АУППосещаемость выполнение запроса занимает значительное время. Экранная форма, отражающая результаты показывает посещаемость дисциплины по всем датам, по которым была проведена регистрация посещаемости. Здесь же отражается количество пропусков. Вид экранной формы показан на рисунке ниже. Поля формы позволяют настраивать занесение и просмотр данных в следующем порядке:

Выбирается учебный год и семестр;

- Выбирается институт;
- Выбирается группа и дисциплина;
- Выбирается или вводится дата регистрации посещаемости.

Выбор учебного года, семестра и института устанавливает фильтр по группам института. После выбора нужной группы, в выпадающем списке дисциплин появляются дисциплины изучаемые группой в выбранном семестре учебного года. Далее необходимо выбрать дату регистрации посещаемости. Если сведения по данной дате ранее были введены, то они отразятся в окне регистрации активности. Если необходимо зарегистрировать посещаемость на новую дату, то в этом окне появятся списки студентов с отметками «V» . В колонках «Лек» или (и) «Практ» отмечаются отсутствующие или активные студенты. Если выбрать из выпадающего списка уже существующую дату, то в окне откроется ранее заполненная форма регистрации активности.

Как уже отмечалось, кнопка  позволяет получить статистику посещаемости группы по выбранной дисциплине (рис. 2).

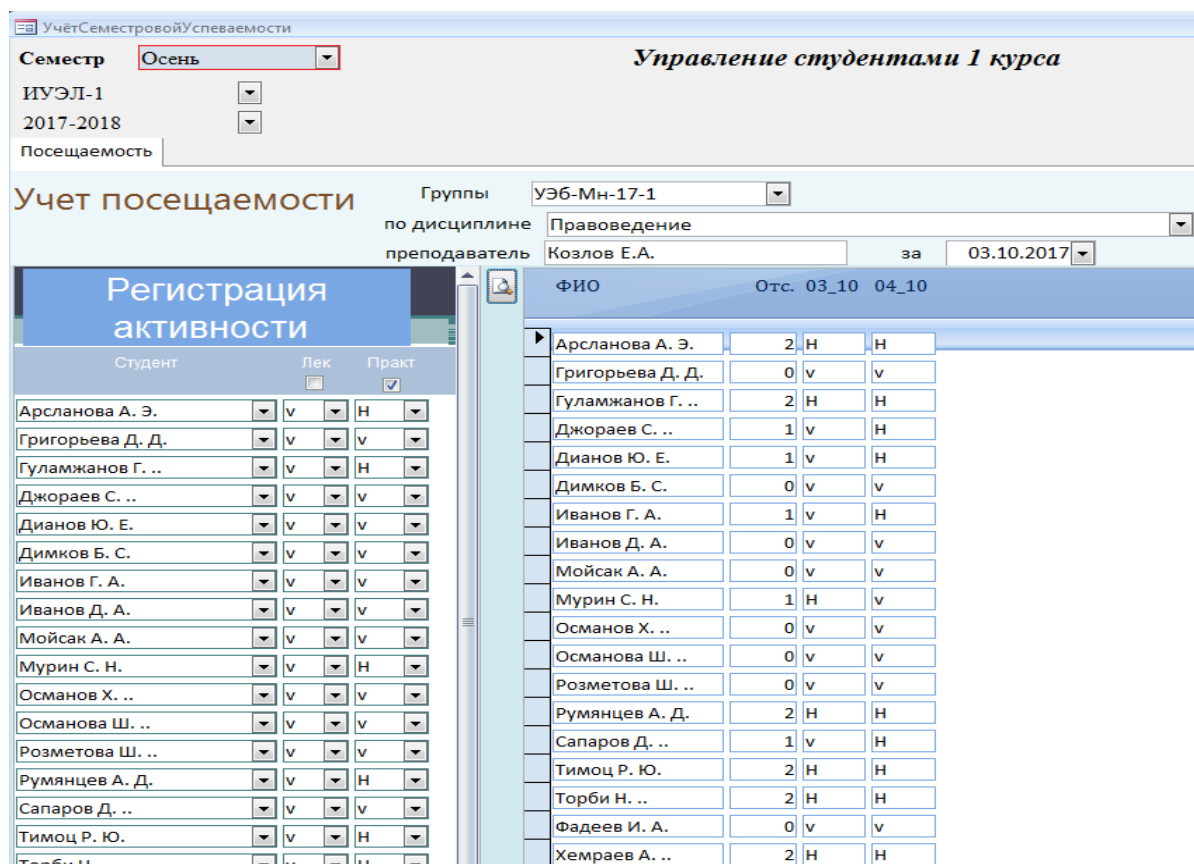
Рассмотренный фрагмент экранной формы позволяет вести учет посещаемости и активности групп по дисциплинам. Накопленные данные предназначены для анализа посещаемости в следующих разрезах.

- Свод посещаемости по каждому студенту группы;
- Обобщенная статистика посещаемости по группе, дисциплине и преподавателю;
- Посещаемость выбранного студента;
- Список студентов группы, процент посещаемости (активности) ниже заданной величины;
- Посещаемость по дисциплинам;
- Посещаемость по преподавателям.

Общий вид экранной формы со статистикой по группе, дисциплине и преподавателю имеет следующий вид.

Внизу экранной формы находятся три кнопки: «Студенты группы», «Дисциплины» и «Преподаватели» (рис. 3).

При нажатии кнопки «Студенты группы» появляется печатная форма – аналог свода посещаемости студентов показанной на рисунке выше. Нажатие кнопки «Дисциплины» приводит к появлению печатной формы показанной ниже (рис.4).



Студент	Лек	Практ
Арсланова А. Э.	v	Н
Григорьева Д. Д.	v	v
Гуламжанов Г. ...	v	Н
Джораев С. ...	v	v
Дианов Ю. Е.	v	v
Димков Б. С.	v	v
Иванов Г. А.	v	v
Иванов Д. А.	v	v
Мойсак А. А.	v	v
Мурин С. Н.	v	Н
Османов Х. ...	v	v
Османова Ш. ...	v	v
Розметова Ш. ...	v	v
Румянцев А. Д.	v	Н
Сапаров Д. ...	v	v
Тимоц Р. Ю.	v	Н
Торби Н. ...	v	Н

Ф.И.О.	Отс.	03_10	04_10
Арсланова А. Э.	2	Н	Н
Григорьева Д. Д.	0	v	v
Гуламжанов Г. ...	2	Н	Н
Джораев С. ...	1	v	Н
Дианов Ю. Е.	1	v	Н
Димков Б. С.	0	v	v
Иванов Г. А.	1	v	Н
Иванов Д. А.	0	v	v
Мойсак А. А.	0	v	v
Мурин С. Н.	1	Н	v
Османов Х. ...	0	v	v
Османова Ш. ...	0	v	v
Розметова Ш. ...	0	v	v
Румянцев А. Д.	2	Н	Н
Сапаров Д. ...	1	v	Н
Тимоц Р. Ю.	2	Н	Н
Торби Н. ...	2	Н	Н
Фадеев И. А.	0	v	v
Хемраев А. ...	2	Н	Н

Рис.2. Таблица Учет посещаемости

СводныеОтчетыПосещаемости

Свод посещаемости по институту **ИУЭЛ-1** группе **Уэб-ОУЛП-17-1** По институтам

дисциплине **Правоведение** Преподаватель **Козлов Е.А.**

Свод посещаемости студентов

Занятий

Имя	Отс	%Отс	Акт	%Акт
Волков Н. Б.	4	5%	0	0%
Гайнанов К. Р.	4	5%	0	0%
Голаев В. А.	12	15%	0	0%
Гречин Е. И.	34	44%	0	0%
Додохонов Ш. Ш.	7	9%	0	0%
Захаров К. С.	12	15%	0	0%
Зиганшин Д. А.	7	9%	0	0%
Клименко В. И.	0	0%	0	0%
Куликов К. Д.	20	26%	0	0%
Ли А. Д.	2	3%	0	0%

Свод посещаемости группы

Занятий Процент

Отсутствовало

Активных

Студент

Граница посещаемости

Граница активности

Посещаемость дисциплины

Занятий Процент

Отсутствовало

Активных

Посещаемость преподавателей

Занятий Процент

Отсутствовало

Активных

Записи: 1 из 26

Рис. 3. Экранная форма для учета посещаемости и активности групп по дисциплинам

Посещаемость дисциплин

№ п/п	Дисциплина	Занятий	Отсутствовало	%	Активных	%
1	Начертательная геометрия и инж	860	120	13,95%	0	0,00%
2	Общая и неорганическая химия	2084	321	15,40%	62	2,98%
3	Ботаника и дендрология	1175	194	16,51%	0	0,00%
4	Академический рисунок	224	38	16,96%	0	0,00%
5	Экология	1289	227	17,61%	50	3,88%

Рис.4. Печатная форма Посещаемость дисциплин

Печатная форма со статистикой посещаемости по каждому преподавателю появляется при нажатии кнопки «Преподаватели» (рис. 5).

Посещаемость преподавателей

Преподаватель	Занятий	Отсутствовало	Процент	Активных	Процент
	4225	1124	26,60%	130	3,08%
А.С. Кривоногова	117	19	16,24%	0	0,00%
Абакулина Л.Ю.	524	239	45,61%	1	0,19%
Адаева О.Е.	567	174	30,69%	0	0,00%
Алексеева В.Е.	336	71	21,13%	0	0,00%
Алексеева С.В.	468	69	14,74%	0	0,00%

Рис.5. Печатная форма Посещаемость преподавателей

Реализация представленных статистик осуществляется с использованием запросов. Не будем демонстрировать все запросы. Остановимся на запросе реализующем расчет статистики посещаемости студентами преподавателей. Он имеет следующий вид.

```
SELECT dbo_Все_Ведомости.Преподаватель,
Sum(СтатисткаСтудента.ВсегоЗанятий) AS Занятий,
Sum(СтатисткаСтудента.Отс) AS Отсутствовало,
Sum(СтатисткаСтудента.Акт) AS Активных
FROM dbo_Все_Ведомости INNER JOIN СтатисткаСтудента ON
dbo_Все_Ведомости.Код = СтатисткаСтудента.КодВедомости
GROUP BY dbo_Все_Ведомости.Преподаватель
```

Как видно из его структуры статистики посещаемости и активности «Отс» и «Акт» выбираются из некоторой таблицы с именем «Статистка Студента». Эта таблица порождается следующим запросом.

```
SELECT dbo_АУППосещаемость.КодСтудента,
dbo_АУППосещаемость.КодГруппы,
dbo_АУППосещаемость.КодВедомости,
Count(dbo_АУППосещаемость.Активность1) AS ВсегоЗанятий1,
Sum([Лек]+[Практ]) AS ВсегоЗанятий,
Sum((IIf([Активность1]="Н",1,0)+IIf([Активность2]="Н",1,0))
) AS Отс,
Sum((IIf([Активность1]="А",1,0)+IIf([Активность2]="А",1,0))
) AS Акт, Sum((IIf([Активность2]="Н",1,0))) AS Отс1,
Sum((IIf([Активность2]="А",1,0))) AS Акт1
FROM dbo_АУППосещаемость
WHERE
(( (dbo_АУППосещаемость.Семестр)=[Forms]![СеместровыеГрафики]
)![ПолеСоСписком4]))
GROUP BY dbo_АУППосещаемость.КодСтудента,
dbo_АУППосещаемость.КодГруппы,
dbo_АУППосещаемость.КодВедомости;
```

Разработка программного модуля «Учет посещаемости» позволило расширить функциональность подсистемы «Автоматизация управления учебным процессом» в СПб ГЛТУ. В настоящее время эта программа используется при учете посещаемости студентов первого курса. Однако, при соответствующих административных усилиях может быть применена для учета посещаемости старших курсов.

Библиографический список

1. В.А. Горбачев. Автоматизация решения задач управления кафедрой Информационные системы и технологии: теория и практика. Сборник научных трудов. Выпуск 4. СПб 2012 г. с. 98-109
2. В.А. Горбачев. Проектирование программного модуля «Управление сессией» Лесной сектор России: проблемы и пути решения. Сборник научных трудов факультета экономики и управления, СПбГЛТУ, 2014 г.

3. В.А. Горбачев. Проектирование программного комплекса «Управление кафедрой» Экономические проблемы лесного сектора. Сборник научных трудов», СПбГЛТУ, 2012 г.

В.А. Горбачев кандидат экономических наук, доцент
Кафедра информационных систем и технологий
СПбГЛТУ им. С.М. Кирова
ist@spbftu.ru

ПРОЕКТИРОВАНИЕ ПРОГРАММНОГО МОДУЛЯ ИМПОРТА И РАЗНЕСЕНИЯ ОПЛАТ ИЗ БУХГАЛТЕРСКОЙ СИСТЕМЫ «ПАРУС» В БАЗУ ДАННЫХ «ДЕКАНАТ»

В предыдущих статьях автора [1-4] рассматривались вопросы автоматизации управления учебным процессом СПбГЛТУ. И в частности работа с программным модулем «Деканат». В данной статье рассматривается решение вопросов реализации интерфейса между программой «Деканат» и бухгалтерской системой «Парус».

Одной из функций программы «Деканат» является учет расчетов с со студентами контрактной формы обучения. Она реализуется в форме программного функционала «Оплата», работа с которым ведется службами ПФУ. В рамках этого функционала решаются следующие задачи:

- Заключение договоров и дополнительных соглашений со студентами контрактниками.
- Начисление и учет оплат за семестр или год.
- Получение отчетных форм о расчетах со студентами.

Установка сумм начислений по различным направлениям обучения, курсам и другим уровням группировки осуществляется при зачислении и переводе студентов с курса на курс.

Разнесение оплат в рамках системы «Деканат» выполняется вручную при получении работниками ПФУ документов об оплате из бухгалтерии. Такой порядок действий является источником ошибок, которые в дальнейшем трудно обнаружить. Поскольку получаемые из бухгалтерии документы формируются в системе «Парус», то возникло предложение формировать в последней файл с реквизитами оплат по студентам, передавать его в ПФУ, и далее программным способом разносить в соответствующие таблицы базы данных «Деканат». Данная статья описывает программную реализацию этого функционала.

Постановка задачи формулируется следующим образом. В бухгалтерской системе «Парус» программным способом формируется текстовый файл (csv файл), содержащий информацию о прохождении оплат. Сведения об оплате содержат следующую информацию: фамилию, имя, отчество студента

(абитуриента); номер договора; сумму и дату оплаты; прочие реквизиты оплаты.

Форма этих данных имеет следующий вид:

Б*;Ш*;*;15-070;оплата;99400;31.07.2018;П/П;395;31.07.2018

Д*;Д*;*;17-201;оплата;44700;31.07.2018;П/П;8534;31.07.2018

Х*;М*;*;ХТи18-051;оплата;165250;31.07.2018;П/П;8688;31.07.2018

О*;Д*;*;ТМи18-102;оплата;30000;31.07.2018;П/П;320;31.07.2018

Поскольку системы «Парус» и «Деканат» эксплуатируются независимо и не имеют общего информационного пространства, принято решение передавать в ПФУ файл с указанными данными посредством электронной почты.

Далее возникает задача перенести данные файла в соответствующие таблицы базы данных «Деканат» или «Абитуриенты», структура которых будет рассмотрена далее. Принцип привязки строки оплаты к студентам (абитуриентам) основан на предположении, что номер договора является уникальным для студента или абитуриента. При этом должно учитываться обстоятельство, что если искомый номер договора есть в базе «Абитуриенты» и он отсутствует в базе «Деканат», то оплата разносится в таблицу базы «Абитуриенты». Если он есть и там и там, то только в таблицу базы «Деканат». Это обусловлено тем, наличие одного и того же договора в обеих базах говорит о том, что абитуриент зачислен в студенты и оплату нужно проводить студенту.

В базе данных «Деканат» сведения о начислениях и оплатах хранятся в таблице «Студенты_Оплата», которая связана с таблицей «Все_Договора» по коду студента. Структуры таблиц показаны ниже (рис. 1).

dbo_Студенты_Оплата		dbo_Все_Договора	
Имя поля	Тип данных	Имя поля	Тип данных
Код	Счетчик	Код_Договора	Счетчик
Код_Студента	Числовой	Код_Специальности	Числовой
Сумма_Оплаты	Денежный	Код_факультета	Числовой
Тип_Оплаты	Числовой	Код_Студента	Числовой
Дата_Оплаты	Дата/время	ФИО	Текстовый
Код_группы	Числовой	Номер_Договора	Текстовый
Удалена	Логический	Дата_Заключения	Дата/время
Дата_удаления	Дата/время	Год_Начала	Числовой
Код_пользователя	Числовой	Текущий	Логический
Курс	Числовой	Скидка	Числовой
Семестр	Числовой	Сумма_оплаты	Денежный
Квитанция	Текстовый	БывшаяФамилия	Текстовый
Примечания	Текстовый		
Код_договора	Числовой		
Учебный_год	Числовой		
КодПретензии	Числовой		
ВесНачисления	Числовой		
СтарыйГод	Числовой		

Рис.1. Структуры таблиц

Такая структура обуславливает следующий алгоритм заполнения таблицы «Студенты_Оплата».

Данные из текстового (csv) файла переносятся в промежуточную таблицу «ОплатаReception», которая имеет следующую структуру (рис.2.).

ОплатаReception	
Имя поля	Тип данных
Фамилия	Текстовый
Имя	Текстовый
Отчество	Текстовый
Договор	Текстовый
ТипОплаты	Текстовый
СуммаОплаты	Текстовый
ДатаОплаты	Дата/время
ПП	Текстовый
НомерДок	Текстовый
ДатаДок	Текстовый
Вкл	Логический

Рис.2. Таблица «ОплатаReception»

Последовательность полей таблицы полностью соответствуют структуре одной строки исходного файла, кроме последнего поля «Вкл». Назначение его будет разьяснено ниже.

Необходимо отметить, что подобные таблицы есть и базе «Абитуриенты». Это «Все_Абитуриенты», «Все_Заявления» и «ОплатаДоговора». Их структура имеет соответствующие для использования поля и в целях экономии места не рассматриваются.

Из каждой строки заполненной данными таблицы «ОплатаReception» выбирается значение поля «Договор», по которому в таблице «Все_Договора» отыскивается наличие студента с таким договором. Если договор (студент) найден, то в таблицу «Студенты_Оплата» должна быть добавлена запись об оплате, в которую входят поля: «Код студента», «Сумма Оплаты», «Тип оплаты», «Дата оплаты», «Курс», «Семестр», «Учебный год», «Код группы». Остальные поля носят вспомогательный характер и не заполняются. После этого в поле «Вкл» каждой строки таблицы «ОплатаReception» заносится значение «Истина», чтобы отметить, что данные договора для этой строки найдены в базе «Деканат». Строки таблицы, для которых не найдены соответствующие договора имеют в поле «Вкл» значение «Ложь» и будет сделана попытка найти эти договора в базе данных «Абитуриенты». Это общая идея алгоритма. В действительности, прежде чем разносить оплаты по таблицам баз данных, необходимо представить пользователю для контроля данные таблицы «ОплатаReception» в виде, отражающем условия поиска по базам данных. Для этого была создана промежуточная таблица «Оплата в базу», вид которой в экранной форме выглядит примерно так, как показано ниже (рис. 3).

Здесь кроме перечисленных ранее полей появляются следующие данные:

- «Статус» - в какой базе найден студент с данным номером;
- «Тек» - текущий договор, то есть договор действующий на данный МОМЕНТ;

Разнести оплату студентов									
Статус	ФИО	Договор	Курс	Сем	Сумма	Дата	Квитанция	Тек	
Студ	Овезов Тагандурды	15-120	4	7	136 700,00р.	03.07.2018	25743 03.07.20	Да	
Студ	Сомов Илья Сергеевич	ЛП16-34	3	5	3 941,00р.	03.07.2018	84028 03.07.20	Да	
Студ	Абрамова Виктория Андреев	УЭ16-44м			100,00р.	03.07.2018	320 03.07.2018	Нет	
Студ	Быковский Станислав Констан	УЭ17-22м	2	3	30 000,00р.	03.07.2018	82981 03.07.20	Да	
Студ	Лочаков Станислав Алексеев	ХПэ17-44	3	5	9 000,00р.	03.07.2018	84028 03.07.20	Да	
Студ	Суслова Оксана Сергеевна	ЛАэ18-01			56 000,00р.	03.07.2018	84028 03.07.20	Да	
Абит	Гринько Кирилл Васильевич	ХПэ18-01	1	1	49 400,00р.	03.07.2018	320 03.07.2018	Да	

Рис. 3. Промежуточная таблица «Оплата в базу»

На данном выше примере можно отметить следующие особенности:

- Все строки оплаты, которая пришла из бухгалтерии, относятся к студентам, кроме последней;
- Все, кроме одного, договора являются действующими;
- В не действующем договоре (помечен желтым цветом) не определены курс и семестр. Значит нужно выяснить, почему на него пришла оплата, а такое допустимо;
- В действующем договоре (вторая строка снизу) не задан курс и семестр. Скорее всего, в договоре не заполнены данные о начислениях студенту.

Анализ подобного рода необходимо выполнять перед разнесением оплат. Какие действия нужно выполнить пользователю, чтобы подготовить данные примера показанного выше, для разнесения оплаты?

- Для строки, в которой значение столбца «Тек» равно «Нет», нужно найти с помощью программы «Деканат» соответствующий договор. И если оплата в действительности проведена правильно, необходимо проставить правильные курс и семестр и изменить значение поля «Дек» на «Да».

- Для строки с текущим договором, но отсутствующими значениями столбцов «Курс» и «Семестр» необходимо найти и исправить этот договор. Скорее всего, в нем вообще отсутствуют начисления на оплату договора.

Поле подобного анализа и исправления ошибок можно запускать программу разнесения оплат по таблицам баз данных «Деканат» и «Абитуриенты». Понятно, что в обработку включаются только строки со значением «Да» в столбце «Тек».

Эту задачу в программе под управлением ACCESS можно решить с помощью SQL запроса следующего вида.

```
INSERT INTO ОплатаВБазу ( ФИО, КодСтудента, НомерДоговора, СуммаОплаты, ДатаОплаты, Квитанция, Курс, Семестр, КодЗагрузки, ГодКурса, ТипЗаписи, Текущий, СтудАбит )
```

```
SELECT (ОплатаReception.Фамилия & ' ' & ОплатаReception.Имя & ' ' & ОплатаReception.Отчество), Все_Договора.Код_Студента, ОплатаReception.
```

```

Договор, Iif(TunОплаты='Оплата', Val([СуммаОплаты]), -Val([СуммаОплаты]))
AS Оплачено, ОплатаReception.ДатаОплаты, [НомерДок] & ' ' & [ДатаДок]
AS Квитанция, Nz(DLookup('Курс', 'dbo_Студенты_Оплата', 'Код_Студента='
& [Код_Студента] & ' And Tun_Оплаты = -1 And Учебный_год = ' &
DLookup('Код', 'Учебный_Год', 'Год = ' & Year([ДатаОплаты]))-
IIf(Month([ДатаОплаты])<=6,1,0))) AS Курс1, [Курс1]*2-1 AS Семестр1, "
& Полеб & " AS Cod, DLookup('Код', 'dbo_Учебный_Год', 'Год = ' &
Year([ДатаОплаты]))-IIf(Month([ДатаОплаты])<=6,1,0) AS Год,
Iif(Все_Договора.Текущий=0,1,0) AS TunЗап, Все_Договора.Текущий, -1 AS
Студ
FROM Все_Договора INNER JOIN ОплатаReception ON
Все_Договора.Номер_Договора = ОплатаReception.Договор

```

Поэтому под управлением ACCESS был разработан прототип программы, а затем переведен на профессиональную платформу.

После всех этих подготовительных действий можно запускать программный код по разнесению оплат.

SQL запрос, реализующий разнесение оплат в базе «Деканат» имеет следующий вид.

```

INSERT INTO Студенты_Оплата ( Код_Студента, Код_договора, Курс, Семестр,
Tun_Оплаты, Сумма_Оплаты, Дата_Оплаты, Квитанция, Учебный_год )
SELECT ОплатаВБазу.КодСтудента, Все_Договора.Код_Договора, ОплатаВБазу.Курс,
ОплатаВБазу.Семестр, 1 AS Tun, ОплатаВБазу.СуммаОплаты, ОплатаВБазу.ДатаОплаты,
ОплатаВБазу.Квитанция, dbo_АУП_ОплатаВБазу.ГодКурса
FROM Договора INNER JOIN dbo_АУП_ОплатаВБазу ON Договора.Номер_Договора =
ОплатаВБазу.НомерДоговора
WHERE ОплатаВБазу.КодЗагрузки = " & Полеб & " AND ОплатаВБазу.Текущий=1 AND
ОплатаВБазу.СтудАбит=True

```

Обратим внимание, что данный запрос выполняется только для значений *Текущий=1* и *СтудАбит=True*, что означает обработку только договоров, помеченных как текущие и относящиеся к студентам. Аналогичный SQL запрос формируется и для базы «Абитуриенты». Только для этого обрабатываются строки, отобранные по критерию *СтудАбит=False*.

Программная реализация описанного алгоритма требует соответствующего интерфейса для пользователя. Для решения этой задачи предлагается следующий интерфейс (рис.4).

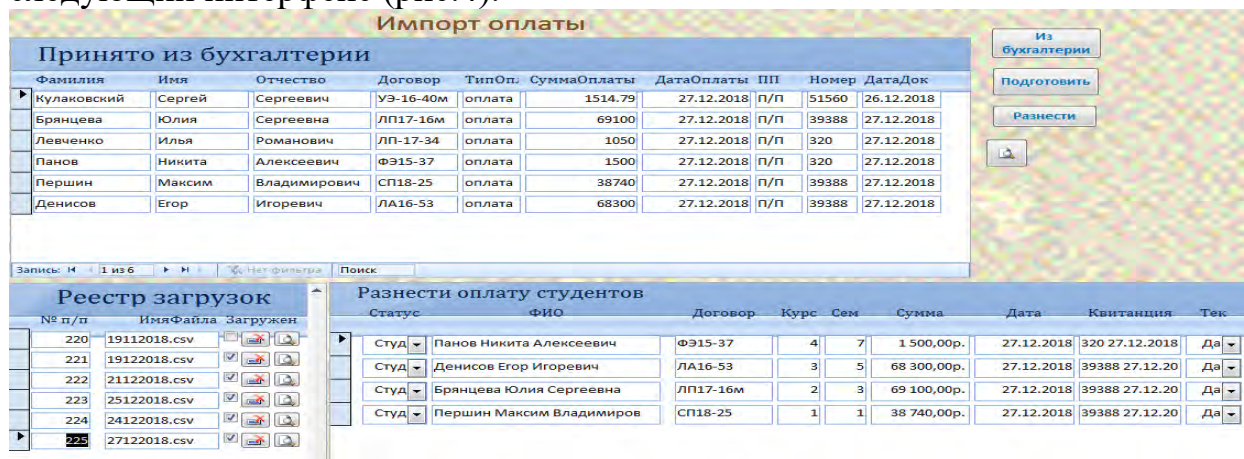


Рис. 4. Вид интерфейса для пользователя

Помимо данных, отражающих сведения принятые из бухгалтерии и подготовленные для разнесения по базам, в представленной экранной форме существует «Реестр загрузок» (рис. 5). Эта таблица хранит учетные данные о поступившей и подготовленной к разнесению в базы данных информации. Каждая строка этой таблицы содержит код загрузки, имя файла, поступившего из бухгалтерии, и отметку о разнесении оплат. Имя файла включает дату его формирования. Для того, чтобы сохранить историю разнесения оплат, в базу данных «Деканат» добавлены две таблицы: данные, поступившие из бухгалтерии (таблица «ОплатаResertion»); данные подготовленные к разнесению оплат («ОплатаВБазу»).

Изложенный выше алгоритм и структура данных проекта предполагают следующую последовательность действий пользователя.

При нажатии кнопки «Из бухгалтерии» появляется окно выбора файла для загрузки. После нажатия кнопки «Открыть» содержимое файла отражается в окне формы «Принято из бухгалтерии».

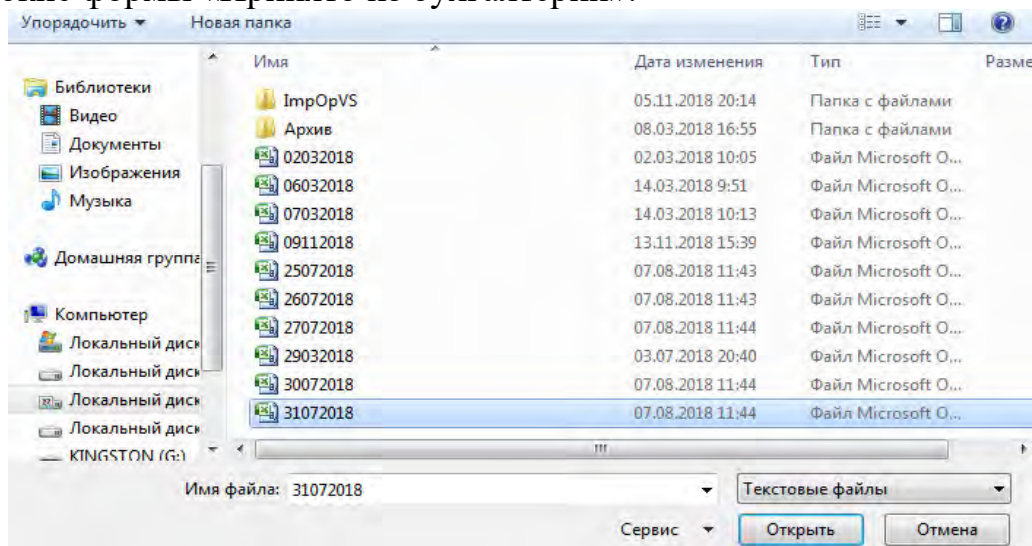


Рис. 5. Реестр загрузок

Нажав кнопку «Подготовить» пользователь запускает программу подготовки данных к разнесению оплат, как было описано выше. Результаты отражаются в окне «Разнести оплату студентов». После корректировки (если необходимо) данных формы, нажимается кнопка «Разнести». Все записи об оплате договоров разносятся по соответствующим таблицам.

Программа установлена в планово финансовом управлении и используется по назначению.

Библиографический список

1. В.А. Горбачев Автоматизация решения задач управления кафедрой Информационные системы и технологии: теория и практика. Сборник научных трудов. Выпуск 4. СПб 2012 г. с. 98-109
2. В.А. Горбачев Проектирование программного модуля «Управление сессией» Лесной сектор России: проблемы и пути решения. Сборник научных трудов факультета экономики и управления, СПбГЛТУ, 2014 г.

3. В.А. Горбачев Проектирование программного комплекса «Управление кафедрой» Экономические проблемы лесного сектора. Сборник научных трудов», СПбГЛТУ, 2012 г.

4. В.А. Горбачев Вопросы проектирования информационных технологий при создании электронной информационно-образовательной среды СПбГЛТУ, 2018 г., 11 с.

М.И. Думов, магистр
СПб ГЛТУ им. С.М.Кирова
ist@spbftu.ru

С.П. Хабаров, кандидат технических наук, доцент
Кафедра информационных систем и технологий
СПб ГЛТУ им. С.М.Кирова
serg.habarov@mail.ru

МОДЕЛИРОВАНИЕ РАБОТЫ УЗЛОВ WI-FI СЕТИ С УЧЕТОМ ИНТЕРФЕРЕНЦИОННЫХ ПОМЕХ ОТ СОСЕДНИХ УЗЛОВ

Большинство современных информационных систем различного прикладного назначения [1-4] имеют распределенную структуру, используя в своем составе как различные вычислительные устройства [5], так и широкий набор протоколов [6,7]. Отдельные узлы этих систем связываются между собой различными каналами передачи данных, образуя достаточно сложную сетевую структуру. На начальных этапах проектирования таких систем особую роль играет процесс моделирования их структуры, который позволяет выполнить анализ качества их функционирования и обоснованности выбора соответствующего оборудования.

При этом, наряду с методами физического и аналитического моделирования, широкое применение находит и имитационное моделирование, которое позволяет проводить исследования и получать практически те же результаты, что и на реальном оборудовании [8]. Исследование имитационных моделей позволяет сравнить различные сценарии работы системы, лучше изучить происходящие в ней процессы. Одной из наиболее популярных сред имитационного моделирования является пакет OMNeT++ [9,10], который позволяет достаточно просто строить и исследовать модели систем с беспроводными каналами связи между ее узлами.

Постановка задачи и объект исследования. В данной статье рассмотрен подход к построению и исследованию в среде OMNeT++ модели беспроводной сети, в которой на площади 500x500 метров присутствуют 5 беспроводных узлов с радиусом действия приемопередатчиков их сетевых адаптеров в 250 метров. Узлы hostA и hostB находятся на расстоянии 400 метров друг от друга, между ними размещены еще три узла: hostR1, hostR2 и hostR3,

которые могут при своей работе оказывать интерференционные помехи на работу соседей. Требуется построить и исследовать модель взаимодействия узлов hostA и hostB, когда один из них в случайные интервалы времени отправляет UDP-сообщения размером в 1 килобайт, а второй принимает эти сообщения.

Особенность исследуемой модели состоит в том, что приемопередатчики сетевых адаптеров отдельных узлов Wi-Fi сети обладают малой мощностью, что приводит к ограничениям на диапазон их работы. Может оказаться невозможной связь двух узлов сети между собой, если достаточно сильны внешние помехи или расстояние между ними выше допустимого. Однако соединение может быть установлено, если между этой парой узлов располагаются другие узлы, которые могут транслировать и передавать сетевые пакеты. Для этого все, или часть, промежуточных узлов должны уметь поддерживать маршрутизацию. В данной статье исследуется возможность UDP обмена между двумя Wi-Fi узлами, находящимися на расстояниях, превышающих их диапазон работы, за счет использования промежуточных узлов, которые в состоянии поддерживать обычную статическую маршрутизацию.

Формирование модели и запуск ее в работу. При первом запуске модели в работу в графической среде выполнения OMNeT++ (рис. 1) можно наблюдать, что узел hostA ведет передачу UDP-сообщений. При этом узлы hostR1 и hostR2 находятся в зоне его радиодоступа, а узлы hostR1 и hostR3 в зоне радиодоступа узла hostB. Узел hostB настроен на прием UDP-сообщений от узла hostA, однако не принимает их, так как находится вне радиодоступа. Находящиеся рядом узлы hostR1 и hostR2, наоборот, находятся в зоне радиодоступа узла hostA и могут принимать сообщения от узла hostA, однако, приходящие пакеты отбрасываются ими, так как они адресованы другому узлу.

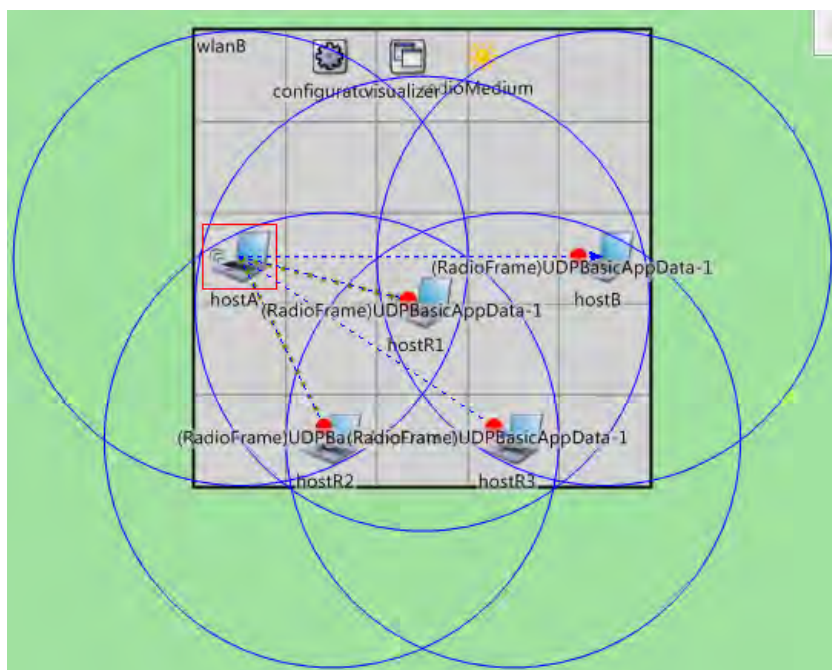


Рис. 1. Визуализация работы модели беспроводной сети

Из рисунка также видно, что узел hostR1 находится в зоне радиоприема как узла hostA, так и узла hostB, и потенциально способен передавать пакеты между узлами hostA и hostB.

Включение и настройка статической маршрутизации. Для того, чтобы узел hostR1 имел возможность перенаправлять пакеты, поступающие от узла hostA, к узлу hostB потребовалось в конфигурационном файле исследуемой модели Wi-Fi сети настроить статическую маршрутизацию для подключаемого в эту модель модуля IPv4NetworkConfiguration, который входит в состав фреймворка INET OMNeT++.

```
# Автоматическая настройка статических маршрутов
*.configurator.config = xml("<config>
  <interface hosts='**' address='10.0.0.x'
    netmask='255.255.255.0' />
  <autoroute metric='errorRate' /></config>")

# Отключение оптимизации записей таблицы маршрутизации.
# Для каждого хоста будет отдельная запись
#.Таблицы будут длиннее, но их гораздо легче понять
*.configurator.optimizeRoutes = false

# Отключение записей таблицы маршрутизации, созданных из #
# сетевой маски (также отключает маршруты по умолчанию).
**.RoutingTable.netmaskRoutes = ""
```

Эта настройка выполняется с помощью XML-строки, параметры которой сообщают конфигуратору, чтобы он назначал IP-адреса в диапазоне 10.0.0.x и использовал оценочную частоту ошибок передач между узлами сети для настройки статических маршрутов. В этом случае, конфигуратор рассматривает беспроводную сеть как полно связанный граф, дугам которого присвоено значение частоты ошибок по каждому из каналов канала связи. При использовании модели распространения волн типа IdealRadio:

- Узлы, которые находятся вне диапазона радиодоступа, будут иметь коэффициент ошибок 1;
- Узлы, находящиеся внутри диапазона радиодоступа, будут иметь коэффициент ошибок равный значению `configurator.minLinkWeight`, которое по умолчанию составляет $1E-3$.

Таким образом, маршруты будут формироваться так, чтобы минимизировать суммарные ошибки, что приводит к созданию правильно настроенной сети IPv4 без каких-либо дополнительных ручных настроек. Сформированные конфигуратором таблицы маршрутизации хранятся в параметре `routingTable` каждого из узлов сети, которые можно посмотреть, находясь в графической среде выполнения. Для примера, на рис. 2 приведена таблица маршрутизации, которая была сгенерирована для узла hostA.

Из этой таблицы видно, что узел hostA имеет прямой интерфейс для связи с узлами hostR1 и hostR2. Также, в таблице есть маршрут, который со-

общает, что до узла hostB можно добраться через узел hostR1, используя последний в качестве шлюза. В таблице маршрутизации узла hostA также существует запись о том, что узел hostR3 является доступным для узла hostA через шлюз hostR2.

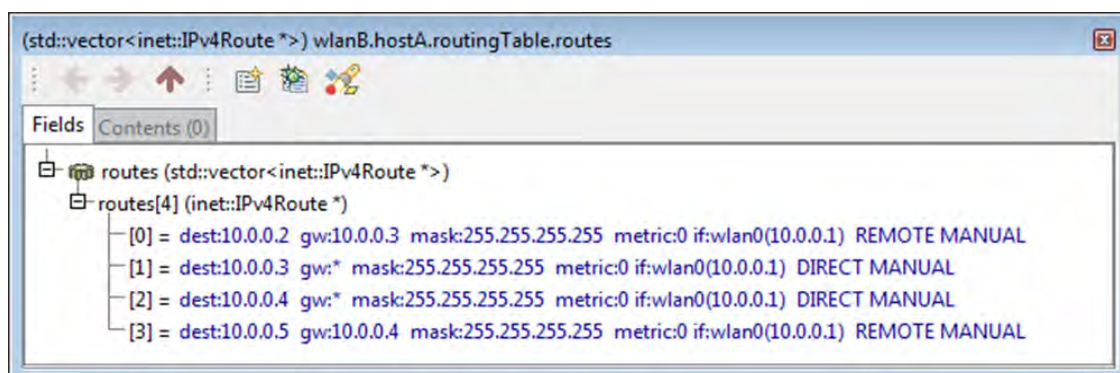


Рис. 2. Таблица маршрутизации узла hostA

После окончания конфигурирования модели исследуемой сети и запуска ее на исполнение, можно наблюдать следующую последовательность в работе исследуемой модели:

- Первый пакет, отправленный узлом hostA, прибывает на узел hostR1, между двумя узлами появляется пунктирная стрелка, указывающая на успешный обмен, выполненный на физическом уровне.
- Через несколько событий, пунктирная стрелка становится сначала желтой, а затем синей, что свидетельствует об успешном обмене между двумя каналами уже на уровне приложений.

В отличие от модели беспроводной сети, которая была рассмотрена в [10], сформированное на узле hostA UDP-сообщение для узла hostB, поступая на физический уровень отправляется не прямо в сеть по MAC адресу узла hostB, а проходя через сетевой и канальный уровни, адресуется для передачи в сеть по адресу hostR1 (рис. 3).

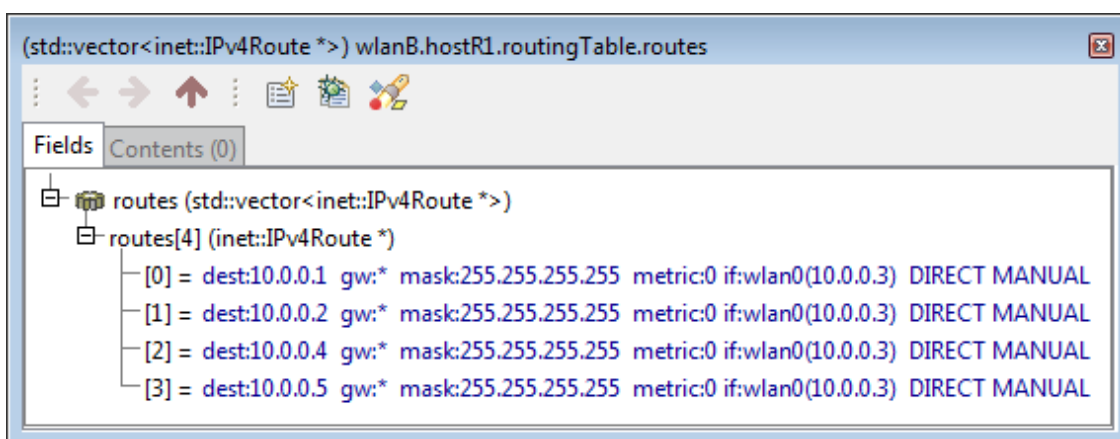


Рис. 3. Таблица маршрутизации узла hostR1

Это происходит в соответствии с таблицей маршрутизации узла hostA, и все пакеты, предназначенные для доставки на узел hostB, сначала будут отправляться на вход узла hostR1, подниматься до сетевого уровня и тут же отправляться обратно в сеть в соответствии с таблицей маршрутизации узла

hostR1 (рис. 3). Из нее видно, что узел hostR1 имеет прямой маршрут к узлу hostB, поэтому поступивший на его вход пакет, будет непосредственно транслирован узлу hostB. Как только принятый узлом hostB пакет поступит на уровень UDP, между узлом hostA и узлом hostB появится сначала пунктирная, а затем сплошная линия, свидетельствующая об успешной передаче UDP-сообщения.

Учет взаимных помех. Рассмотренный пример реализовывал идеализированные условия работы сети, при которых не учитывается такое физическое явление как интерференция радиоволн, которое возникает при поступлении на вход радиоприемника двух и более радиосигналов, из-за чего они сталкиваются, искажаются и сбивают нормальную работу приемника радиосигнала. В среде моделирования OMNeT++ реализовано четыре способа описания и представления сигналов:

- Первое представление называется range-based (основанное на диапазонах), которое реализовано в компоненте IdealRadioMedium. Преимущества этой структуры в компактности, предсказуемости и высокой производительности. Недостаток этой структуры заключается в том, что она не очень точно отражает реальность. Тем не менее, этого представления вполне достаточно для оценки характера работы сети или разработки нового протокола маршрутизации, когда точное моделирование потерь пакетов не имеет значения.

- Вторая структура представляет собой узкополосный сигнал со скалярной мощностью сигнала, несущей частотой и полосой пропускания. Преимущество этого представления состоит в том, что он позволяет вычислять отношение сигнала/шума, а также то, что в большинстве случаев его достаточно для моделирования сетей IEEE 802.11.

- Третья структура данных описывает сигнал, мощность которого изменяется со временем. В этом случае мощность сигнала представлена одномерным временным значением, которое точно следует за переданными импульсами. Это представление используется для моделей радиоволн стандарта IEEE 802.15.4a UWB.

- Четвертое представление использует многомерные значения для описания мощности сигнала, которая изменяется как по времени, так и по частоте. Модель IEEE 802.11b может использовать это представление.

В рассматриваемом примере модель физической среды базируется на модуле IdealRadioMedium, который использует диапазонное представление, где степень влияния сигнала на соседние узлы зависит от расстояния, на котором они находятся. При этом выделяется три диапазона:

- Communication range – диапазон уверенного приема и передачи радиосигнала.

- Interference range – диапазон, в котором нормальная связь невозможна, но он оказывает существенные воздействия на приемопередатчики других устройств.

- **Detection range** – диапазон, где нет воздействий узла на приемопередатчики других устройств, но есть возможность обнаружить факт наличия и работы этого устройства.

Результаты моделирования. Для исследования степени взаимного влияния радиоволн, в конфигурационном файле рассматриваемой модели, компонент `IdealRadioMedium` был настроен на интерференционный диапазон с расстоянием 500м. Присвоенные параметры описывают тот факт, что радиосигналы становятся слабее с расстоянием. При этом существует диапазон, в котором их нельзя уже принять правильно, но они еще достаточно сильны, чтобы мешать другим сигналам и приводить к сбою их приема.

В процессе проведения модельных экспериментов (при разных временах прогона модели и различных интенсивностях генерации UDP сообщений) было установлено, что узел `hostB` получал не все посланные ему узлом `hostA` сообщения. Это было вызвано тем, что пакеты на `hostB` могли поступить только с узла `hostR1`, но при их передаче продолжал свою работу и узел `hostA`, сигналы которого интерферировали с сигналами, исходящими из узла `hostR1`.

Установление этого факта потребовало проведения более подробного модельного эксперимента с возможностью получения более точного представления о сути процессов, происходящих в модели беспроводной сети. В этом модельном эксперименте:

- Время функционирования ограничивалось всего одной секундой.
- Задавалось, что длительность интервалов между моментами формирования UDP-сообщений на узле `hostA` описывается экспоненциальным распределением со средним значением равным 10 мс.

```
hostA.udpApp[0].sendInterval = exponential(10ms)
```

- До запуска модели в работу дополнительно включался режим регистрации и записи событий, происходящих во время работы модели.

Как было обнаружено из анализа полученных данных, посланки с узла `hostA` транслировались на все другие узлы, однако связь была установлена только с узлами `hostR1` и `hostR2`, а взаимодействие между узлами `hostR1` и `hostB`, произошло только по прошествии длительного времени. По окончании режима прогона модели были сформированы файлы отчетов, из анализа которых выяснилось, что:

- На уровне приложений узла `hostA` было сформировано за 1с модельного времени 92 UDP-сообщения размером 1000 байт каждое;
- Уровень UDP узла `hostA` прошла 91 датаграмма длиной 1008 байт;
- В сеть, пройдя уровень MAC, было отправлено 89 пакетов, размером уже в 1028 байт каждый.

Изменение размера исходного сообщения при прохождении его по протокольному стеку объясняется добавлением к нему соответствующих заголовков, а именно 8 байт от UDP и 20 байт от IP. Но все это относится к передающей стороне. Значительно больший интерес представляют результаты, полученные на принимающей стороне, то есть на узле `hostB`, где можно об-

наружить, что из 89 посланных ему пакетов уровень MAC и UDP прошел всего один.

Открыв в среде моделирования OMNeT++ файл временной диаграммы, можно увидеть, что отправка сообщения с узла hostA на узел hostB, была осуществлена в момент значительной паузы в генерации UDP_сообщений на узле hostA (рис. 4).

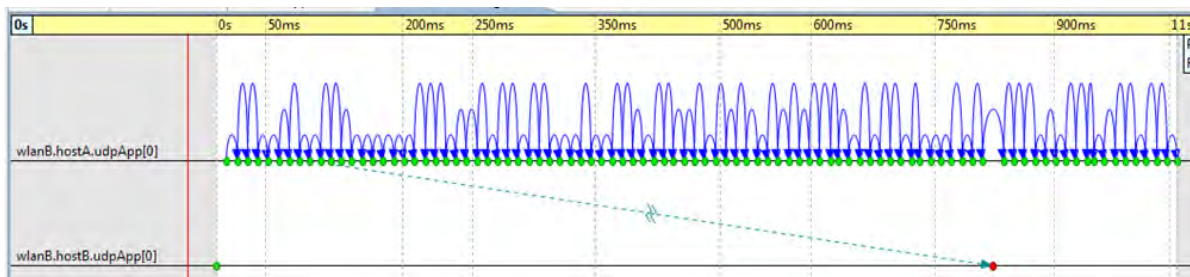


Рис. 4. Временная диаграмма взаимодействия UDP-уровней узлов hosA и hostB

Особый интерес на этой временной диаграмме представляет зеленая пунктирная стрелка с зигзагом на ней. Она обозначает повторную виртуальную пересылку сообщений. Откуда следует, что на вход прикладного уровня узла hostB (`wlanB.hostB.udpApp(0)`) поступило одно из первых сообщений, отправленных с узла hostA.

Это означает, что первое из сгенерированных сообщений не пропадает в сети, а где-то сохраняется. На это можно получить ответ, если вывести на экран график зависимости от времени длины очереди, которая сформировалась в буфере сетевого адаптера узла hostR1, играющего роль шлюза в передачах между узлами hostA и hostB (рис. 5).

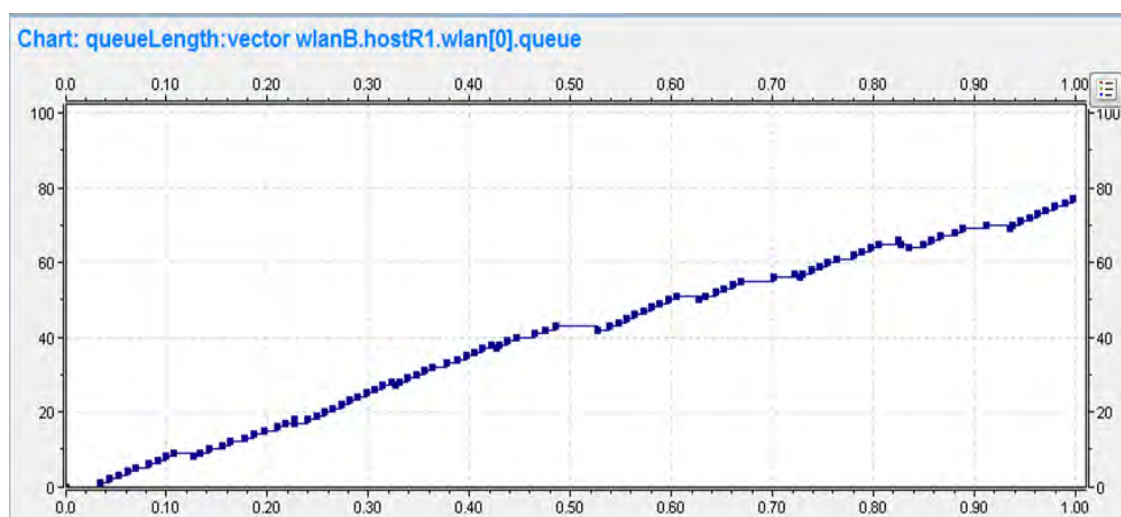


Рис. 5. Размер очереди в буфере сетевого адаптера шлюза hostR1

В рамках данного эксперимента никаких ограничений не накладывалось, и значение максимальной длины очереди буфера адаптера определено по умолчанию в соответствующем модуле фреймворка INET. В реальной беспроводной сети объем буфера сетевого адаптера играет важную роль, особенно если этот адаптер используется в таком узле, как шлюз.

Но что мешало в моделируемой беспроводной сети пакету со шлюза hostR1 дойти до узла hostB раньше? Ответ на этот вопрос можно получить, открыв на экране файл временной диаграммы (рис. 6), анализ которой показывает следующую последовательность происходящих в сети событий:

- Первый пакет (UDPData-0) поступает в сеть (событие #22) и через некоторый интервал времени достигает узла hostR1 (#23) и узла hostB (#26). Причем второе событие происходит позже, так как узел hostB расположен дальше.
- Узел hostB принимает (receptionTimer) радиосигналы от узла hostA (события #26 - #45), но не распознает их ввиду значительной удаленности от источника передачи.
- Узел hostR1 начинает успешный прием пакета от узла hostA (события #23 - #32).

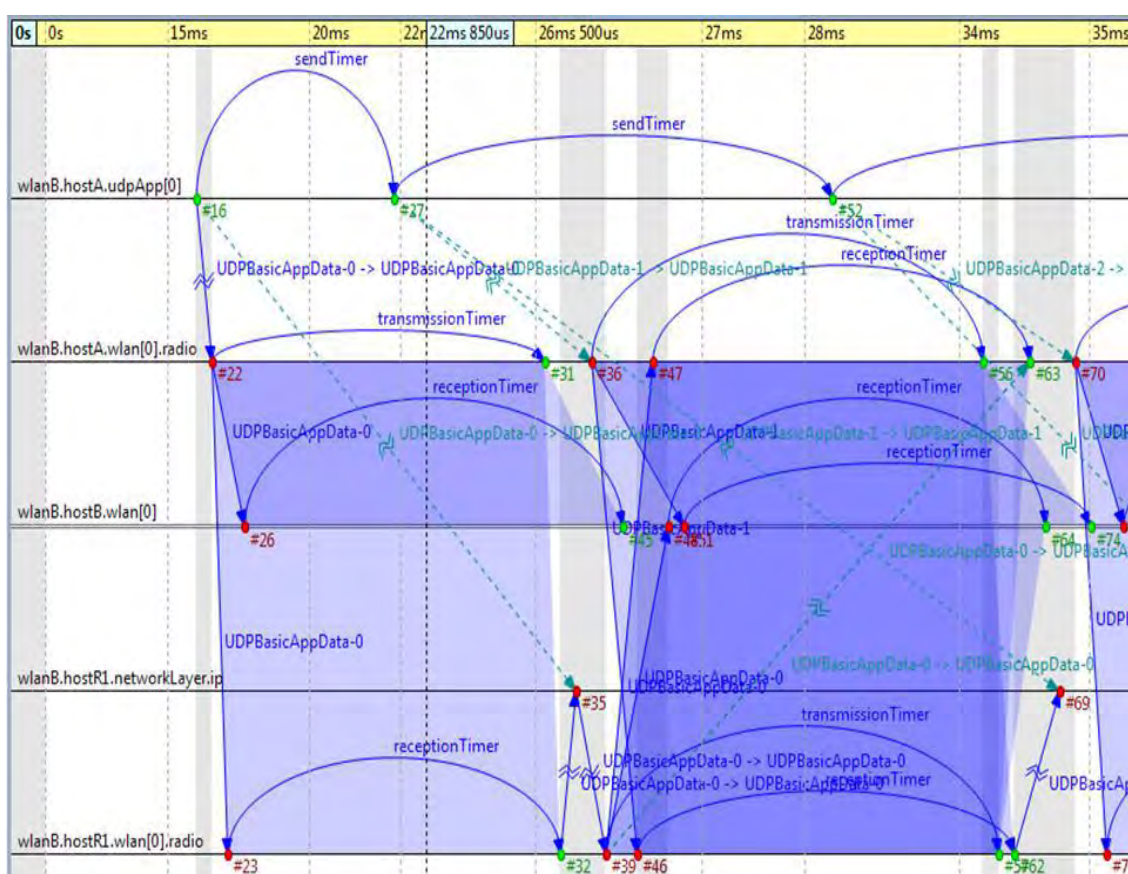


Рис. 6. Временная диаграмма работы исследуемой модели

- В процессе передачи первого пакета, приложение wlanB. hostA.udrApp(0) (событие #27) генерирует второе сообщение UDPBasicAppData-1, которое сразу не может быть отправлено в сеть. Оно помещается в буфер сетевого адаптера и будет находиться там до конца работы сетевого адаптера с первым пакетом. Этот момент описывает событие #36 и зеленая пунктирная стрелка, связывающая его с событием #27.
- Узел hostR1, успешно закончив прием первого пакета (#32), обращается к своей таблице маршрутизации (#35). На ее основе он переадресует принятый пакет на узел hostB и отправляет его в сеть (#39).

- Но чуть раньше (#32) узел hostA начал передачу второго пакета, который до этого хранился в буфере его сетевого адаптера.

- На вход приемника узла hostB поступают одновременно два радиосигнала (события #48 и #51), которые интерферируют между собой, не позволяя узлу hostB распознать пакет, идущий от шлюза.

Сместив временную диаграмму ближе к концу захваченного интервала работы модели сети (рис.7), можно увидеть, что выборка попадает на небольшую паузу в передаче пакетов узлом hostA (sendTimer), что позволяет узлу hostR1 выбрать из буфера сетевого адаптера пакет, ранее полученный им от узла hostA и отправить его в сеть.

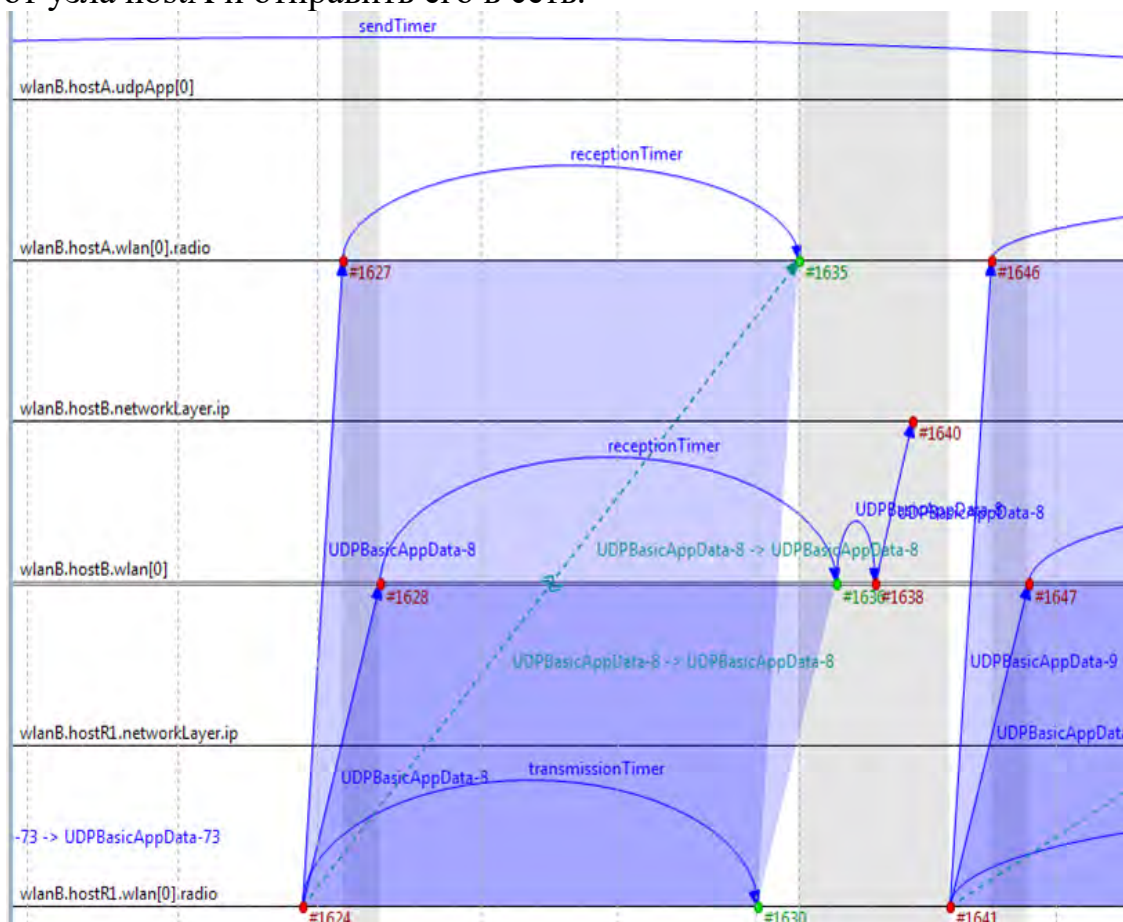


Рис. 7. Временная диаграмма работы исследуемой модели

Диаграмма хорошо показывает распространение сигнала в сторону узлов hostA и hostB, которые успешно принимают этот сигнал. Если узел hostA не выполняет с ним никакой обработки, так как он адресован другому узлу, то узел hostB, приняв сигнал, пропускает его через стек протоколов, реализованных в этом узле, поднимая его до прикладного уровня.

Подводя итог исследованию модели беспроводной сети в условиях учета возможных интерференционных помех от работы соседних устройств, можно сделать вывод, что в этих условиях отмечается существенное снижение производительности сети. Вызвано это тем, что большую часть времени передающий узел и шлюз работают одновременно, что вызывает приход на

трансивер принимающего узла сразу двух сигналов и их «столкновения» между собой.

Библиографический список

1. Заяц А.М., Хабаров С.П. Организация доступа к беспроводным АД НОС сетям информационных систем мониторинга лесных территорий из среды Windows 10. // Известия Санкт-Петербургской лесотехнической академии. – СПб, 2018, Вып. 223. - с. 285-299.
2. Хабаров С.П., Заяц А.М. Использование технологии websocket в клиент-серверных экспертных системах. //В книге: Леса России: политика, промышленность, наука, образование Материалы второй Международной научно-технической конференции. 2017, с. 278-280.
3. Заяц А.М., Хабаров С.П. Настройка беспроводных соединений в системах мониторинга лесных территорий. // Леса России: политика, промышленность, наука, образование / материалы третьей международной научно-технической конференции. Том 1 – СПб.: СПбГЛТУ, 2018. – с.80-83.
4. Заяц А.М., Думов М.И. Обзор беспроводных сенсорных сетей и технологий информационных систем оценки лесной пожароопасности и мониторинга лесов. //Сборник научных трудов «Информационные системы и технологии: теория и практика» - СПб.: СПбГЛТУ, 2016. №8. – с.9-21.
5. Хабаров С.П. Организация гетерогенных ЛВС с терминальным доступом между ее узлами. //Известия Санкт-Петербургской лесотехнической академии – СПб.:СПбГЛТУ, 2016, Вып. 216, с. 267-280.
6. Хабаров С.П., Колмогорцев Е.Л. Отказоустойчивый протокол надежной доставки данных. //Известия Санкт-Петербургской лесотехнической академии – СПб.:СПбГЛТУ, 2005, Вып. 174, с. 143-153.
7. Хабаров С.П. Взаимодействие узлов сети по протоколу WebSocket. //Сборник научных трудов «Информационные системы и технологии: теория и практика» - СПб.: СПбГЛТУ, 2017. №9. – с.109-119.
8. Сусь К.М., Хабаров С.П. Мониторинг беспроводной Wi-Fi сети с использованием сетевого монитора Microsoft Network Monitor. // Информационные системы и технологии: теория и практика: сб. научн. тр., Вып 10. Ч. 2. – СПб.:СПбГЛТУ, 2018. – с.73-82.
9. Хабаров С.П. Моделирование Ethernet сетей в среде OMNeT++ INET framework // Научно-технический вестник информационных технологий, механики и оптики. 2018. Т. 18. № 3. с. 462–472. doi: 10.17586/2226-1494-2018-18-3-462-472
10. Думов М.И., Хабаров С.П. Использование OMNeT++ для моделирования беспроводных Wi-Fi сетей. // Информационные системы и технологии: теория и практика: сб. научн. тр. Вып 10. Ч. 1. – СПб.:СПбГЛТУ, 2018. – с.44-53.

Г.В.Жижин, доктор физ.-мат. наук, профессор
Российская Академия Естественных наук
действительный член, академик,
genzhizhin@gmail.com

И.А.Обухова, кандидат технических наук, доцент
Кафедра информационных систем и технологий
СПбГЛТУ им.С.М.Кирова
lobukhova@inbox.ru

ИНФОРМАЦИОННАЯ МОДЕЛЬ ПЕРЕДАЧИ НАСЛЕДСТВЕННЫХ ПРИЗНАКОВ В ОПЫТАХ МЕНДЕЛЯ

Введение

В 1865г. Г.Мендель выступил с докладом на Броннском обществе естествоиспытателей об опытах над растительными гибридами [4]. Это выступление положило начало развитию генетики как науки, хотя далеко не сразу содержание выступления стало известно научной общественности и по достоинству оценено [1]. Понимание существа опытов Г.Менделя проходило в борьбе различных, часто противоположных научных течений, сопровождалось драматическими историями в личной жизни ученых. В разные времена казалось, что эти опыты доказывают абсолютно противоположные точки зрения. За минувшие 150 лет после выступления Г.Менделя, опираясь на его работы, в генетике получено много выдающихся результатов, в том числе удостоенных нескольких Нобелевских премий. В этой связи можно назвать имена многих выдающихся ученых, таких как А.Вейсман [8], В.Иоганнсен [2], Т.Г.Морган [5], Н.К.Кольцов [3], С.С.Четвериков [6, с.3-54], Дж.Уотсон [9] и многих, многих других. В результате были созданы хромосомная и молекулярная теории наследственности, определены структуры нуклеиновых кислот, структура гена.

Можно отметить также только что опубликованные работы, в которых доказывается, что составляющие молекул нуклеиновых кислот имеют высшую размерность [11]. Эти работы открывают новый этап генетических исследований, учитывающих высшую размерность молекул.

Трудно охватить в одной статье все многообразие результатов, полученных в генетике за прошедшие 150 лет. Эта задача и не ставилась перед данной работой. Мы предлагаем в честь 150-летия работ Менделя вновь обратиться к их рассмотрению, так как некоторые важные аспекты этих экспериментальных исследований все еще не получили соответствующего математического описания и обобщения. Вслед за Менделем мы не будем учитывать возможность более сложных явлений при скрещивании организмов, таких например, как открытый позже кроссингвор (перекрест генов), так как кроссингвор встречается всего в нескольких процентах взаимодействующих организмов [7].

Моногибридное скрещивание

Мендель для определения закономерностей наследования признаков растений тщательно отбирал их так, чтобы они устойчиво имели различающиеся признаки (константно-различающихся признаки). Сейчас мы можем сказать, что в соответствии с хромосомной теорией наследственности, что скрещивались два гомозиготных организма, гомологичные хромосомы каждого из которых содержат два одинаковых гена. Одна пара генов (AA) в одном организме отвечает одному признаку (доминантному), а другая пара генов (aa) в другом организме отвечает другому (рецессивному) признаку. Такое скрещивание называется моногибридным. В соответствии с первым законом Менделя при скрещивании гомозиготных родительских форм, отличающихся одной парой константно-различающихся признаков, все гибриды первого поколения будут единообразны как по генотипу (составу генов), так и по фенотипу (внешнему виду). При скрещивании же гибридов первого поколения в соответствии со вторым законом Менделя в потомстве будет наблюдаться преобладание одного из признаков (доминантного) над другим признаком (рецессивным) в соотношении 3:1 по фенотипу и 1:2:1 по генотипу. Это означает, что если обозначить A доминантный ген, a – рецессивный ген, то организмы первого поколения F_1 характеризуются парой генов Aa , а второе поколение F_2 характеризуется парами AA , Aa и aa в соотношении 1:2:1. Причем, организмы с генами AA имеют одинаковый фенотип, так как влияние рецессивных генов на фенотип сказывается тогда, когда их в гомологических хромосомах имеется два, а не один.

Законы Менделя выполняются тем точнее, чем больше имеется организмов, так эти законы имеют статистическую природу. Пусть в поколении F_1 имеется m организмов и их плодовитость равна v . Тогда в поколении F_2 имеется mv организмов. Учитывая второй закон Менделя, мы должны заключить, что mv кратно 4. Тогда в поколении F_2 имеется $mv/4$ гомозиготных организмов с генами AA , столько же гомозиготных организмов с генами aa и $mv/2$ гетерозиготных организмов с генами Aa :

$$F_2 = \frac{mv}{4} AA + \frac{mv}{2} Aa + \frac{mv}{4} aa .$$

Каждое из составляющих частей второго поколения дает потомство в следующих поколениях F_3 . При этом гомозиготные организмы имеют в потомстве тоже гомозиготные организмы с тем же признаком, но общее число соответствующих организмов увеличивается в v раз. Гетерозиготные же организмы, имеющие как доминантные, так и рецессивные гены порождают и гомозиготные и гетерозиготные организмы. С учетом численности гетерозиготных организмов в поколении F_2 и их плодовитости имеем

$$\frac{mv}{2} Aa = \frac{mv^2}{8} AA + \frac{mv^2}{4} Aa + \frac{mv^2}{8} aa.$$

В сумме гомозиготные и гетерозиготные организмы поколения F_2 дают в поколении F_3 следующее распределение организмов

$$F_3 = mv^2 \left(\frac{1}{4} + \frac{1}{8} \right) AA + \frac{mv^2}{4} Aa + mv^2 \left(\frac{1}{4} + \frac{1}{8} \right) aa.$$

Легко определить, что отношение количеств организмов двух фенотипов в этом поколении равно 5:3.

В поколении F_4 гетерозиготные организмы также порождают как гомозиготные, так и гетерозиготные организмы

$$\frac{mv^2}{4} aA \rightarrow \frac{mv^3}{16} AA + \frac{mv^3}{8} Aa + \frac{mv^3}{16} aa.$$

А гомозиготные организмы увеличивают свою численность в соответствии с их плодовитостью. В сумме в поколении F_4 получается следующее распределение потомства

$$F_4 = mv^3 \left(\frac{1}{4} + \frac{1}{8} + \frac{1}{16} \right) AA + \frac{mv^3}{8} Aa + mv^3 \left(\frac{1}{4} + \frac{1}{8} + \frac{1}{16} \right) aa.$$

Отношение численности организмов разных фенотипов в этом поколении 9:7.

В n -м поколении F_n , рассуждая аналогично, получим следующее выражение для распределения организмов

$$F_n = mv^{n-1} \left(\frac{1}{2^2} + \frac{1}{2^3} + \frac{1}{2^n} \right) AA + \frac{mv^{n-1}}{2^{n-1}} Aa + mv^{n-1} \left(\frac{1}{2^2} + \frac{1}{2^3} + \dots + \frac{1}{2^n} \right) aa.$$

Численность доминантного фенотипа в n -ом поколении равна $mv^{n-1} \left(\frac{1+2+2^2+\dots+2^{n-3}+2^{n-2}+2}{2^n} \right)$.

Численность рецессивного фенотипа в n -ом поколении равна

$$mv^{n-1} \left(\frac{1+2+2^2+\dots+2^{n-3}+2^{n-2}}{2^n} \right).$$

Числитель этой дроби есть геометрическая прогрессия

$$\sum_{k=1}^{n-1} 2^{k-1} = 2^{n-1} - 1.$$

Поэтому численность доминантного фенотипа и рецессивного фенотипа равны, соответственно,

$$mv^{n-1} \frac{2^{n-1} + 1}{2^n} \quad \text{и} \quad mv^{n-1} \frac{2^{n-1} - 1}{2^n}. \quad (1)$$

Их отношение есть

$$\delta_1(n) = \frac{2^{n-1} + 1}{2^n}. \quad (2)$$

Уравнение (2) представляют общее выражение для отношения численностей доминантного и рецессивного фенотипов при моногибридном скре-

щивании для любого номера поколения $n > 1$. (При $n = 1$ существует только один фенотип, поэтому отношение (2) неприемлемо). Легко видеть, что при $n = 2$ отношение (2) есть 3:1, что соответствует второму закону Менделя. Следует отметить, что соотношения (1) и (2) получены в результате математического вывода при условии расщепления гетерозиготных организмов. В работе Менделя [4] содержатся лишь данные опытов, математического вывода законов у Менделя и в последующих работах не было. Из (1), (2) следует, что численности доминантных и рецессивных фенотипов при увеличении номера поколения приближаются друг к другу. При этом численность гибридов $m \left(\frac{v}{2}\right)^{n-1}$ с ростом n , если считать $v > 2$ (обычно выполняется), возрастает. Таким образом, утверждение о том, что с ростом поколений число гибридов стремится к нулю, ошибочно.

Бигибридное скрещивание

Пусть организмы имеют две пары константно-различающихся признаков. В одной паре доминантному признаку соответствуют гены A_1A_1 и рецессивному признаку соответствуют гены a_1a_1 . В другой паре доминантному признаку соответствуют гены A_1A_2 , а рецессивному признаку соответствуют гены a_2a_2 . В каждой паре признаков независимо и одновременно действует второй закон Менделя. Этому отвечает произведение

$$(A_1A_1 + 2A_1a_1 + a_1a_1)(A_2A_2 + 2A_2a_2 + a_2a_2), \quad (3)$$

Так как вероятность произведения независимых событий равна произведению вероятностей этих событий. Результатом этого произведения являются 16 слагаемых

$$A_1A_1A_2A_2 + A_1A_1a_2a_2 + a_1a_1A_2A_2 + a_1a_1a_2a_2 + 2A_1A_1A_2a_2 + 2a_1a_1A_2a_2 + 2A_1a_1A_2A_2 + 2A_1a_1a_2a_2 + 4A_1a_1A_2a_2. \quad (4)$$

Поэтому, если в первом поколении F_1 имеется m гибридных организмов $A_1a_1A_2a_2$, то во втором поколении F_2 с учетом их плодовитости v имеется mv и это количество должно быть кратно 16. Следовательно, второе поколение есть сумма следующих организмов

$$F_2 = \frac{mv}{16} A_1A_1A_2A_2 + \frac{mv}{16} A_1A_1a_2a_2 + \frac{mv}{16} a_1a_1A_2A_2 + \frac{mv}{16} a_1a_1a_2a_2 + \frac{mv}{8} A_1A_1a_2a_2 + \frac{mv}{8} a_1a_1A_2a_2 + \frac{mv}{8} A_1a_1A_2A_2 + \frac{mv}{8} A_1a_1a_2a_2 + \frac{mv}{4} A_1a_1A_2a_2. \quad (5)$$

Выделим среди 16 слагаемых в (4) те слагаемые, которые соответствуют организмам с одинаковыми фенотипами по признаку с генами A_1A_1 . Таких слагаемых 12:

$$A_1A_1A_2A_2, A_1A_1a_2a_2, 2A_1A_1A_2a_2, 2A_1a_1A_2A_2, 2A_1a_1a_2a_2, 4A_1a_1A_2a_2.$$

В (4) есть слагаемые, соответствующие организмам с фенотипом по признаку с генами a_1a_1 . Это 4 слагаемых $a_1a_1A_2A_2$, $a_1a_1a_2a_2$, $2a_1a_1A_2a_2$.

Таким образом, отношение численности организмов в поколении F_2 (5) с фенотипом A_1A_1 и с фенотипами a_1a_1 такое же как и при моногибридном

скрещивании 3:1. Очевидно, что такое же отношение 3:1 у численности организмов с фенотипами $A_2 A_2$ и $a_2 a_2$.

В этом проявляется третий закон Менделя, открытый им в опытах при бигибридном скрещивании – закон независимого расщепления признаков в отношении 3:1 во втором поколении. В данной работе этот закон следует из математического рассмотрения бигибридного скрещивания. Из этого же рассмотрения следует и вторая часть третьего закона Менделя. Действительно, если из суммы (4) выделить слагаемые, соответствующие организмам с фенотипом по $A_1 A_2$, то таких слагаемых 9:

$$A_1 A_1 A_2 A_2, 2A_1 A_1 a_2 a_2, 2A_1 a_1 A_2 a_2, 4A_1 a_1 A_2 a_2.$$

Слагаемые в (4), соответствующие организмам с фенотипом по произведению $a_1 A_2$ есть три $a_1 a_1 A_2 A_2$, $2a_1 a_1 A_2 a_2$. Столько же слагаемых в (4) по фенотипу $F_1 a_2 A_1$. Это $A_1 A_1 a_2 a_2$, $2A_1 a_1 a_2 a_2$ и одно слагаемое в (4), соответствующее организмам с фенотипом $a_1 a_1$, это $a_1 a_1 a_2 a_2$. Таким образом, в поколении F_2 (5) отношение численностей организмов с одновременным присутствием каких-то двух признаков из четырех есть 9:3:3:1. Это есть вторая часть третьего закона Менделя, наблюдаемая им в опытах по бигибридному скрещиванию во втором поколении.

Менделю не удалось выяснить закономерности бигибридного скрещивания после второго поколения. Это можно сделать пользуясь математическим описанием, результативность которого уже доказана. При переходе в третье поколение гетерозиготные организмы $A_1 a_2, A_2 a_2$ в (3) расщепляются по второму закону Менделя. Если сконцентрировать внимание на отношении численности фенотипов по доминантному гену $A_1 A_1$ и рецессивному гену $a_1 a_1$, то после расщепления гетерозиготного организма $A_1 a_1$ в первой скобке произведения (3), это произведение принимает вид $(3A_1 A_1 + 4A_1 a_1 + 3a_1 a_1)f$, где для краткости f обозначает вторую скобку в (3). Отсюда следует, что отношение численностей организмов по фенотипам $A_1 A_1$ и $a_1 a_1$ в третьем поколении есть 7:3. Аналогично, в четвертом поколении произведение (3) принимает вид $(7A_1 A_1 + 8A_1 a_1 + 7a_1 a_1)f$, т.е. в четвертом поколении отношение численностей организмов по фенотипам $A_1 A_1$ и $a_1 a_1$ равно 15:7 и т.д. Раскрывая последовательность получаемым цифр в отношениях заключаем, что в n -ом поколении число организмов по фенотипу $A_1 A_1$ в произведении (3) есть геометрическая прогрессия

$$1+2+2^2+\dots+2^{n-1}=\sum_{k=1}^n 2^{k-1}=2^n-1,$$

а число организмов по фенотипу $a_1 a_1$ есть геометрическая прогрессия

$$1+2+2^2+\dots+2^{n-2}=\sum_{k=1}^n 2^{k-1}=2^{n-1}-1.$$

Таким образом, получаем, что отношение численностей организмов в n -ом поколении по фенотипам $A_1 A_1$ и $a_1 a_1$ при бигибридном скрещивании

$$\delta_2(n) = \frac{2^n - 1}{2^{n-1} - 1}. \quad (6)$$

Ясно, что эта же формула относится и для отношения численностей организмов по фенотипам A_2A_2 и a_2a_2 при бигибридном скрещивании в n -ом поколении ввиду равноправности скобок в произведении (3). Если необходимо рассмотреть отношение численностей организмов по фенотипам с одновременным присутствием двух пар признаков в поколении после второго, то необходимо последовательно расщеплять скобки в произведении (3). Обозначим организмы по фенотипу A_1A_1 символом φ_{A1} . В эти организмы входят организмы с генами A_1A_1 и A_1a_1 . Аналогично φ_{A2} обозначает организмы по фенотипу A_2A_2 , φ_{a1} – организмы по фенотипу a_1a_1 , φ_{A2} – организмы по фенотипу a_2a_2 . Тогда выражение (3) в третьем поколении принимает вид

$$(7\varphi_{A1} + 3\varphi_{a1})(7\varphi_{A2} + 3\varphi_{a2}) = 7^2\varphi_{A1}\varphi_{A2} + 7 \cdot 3\varphi_{a1}\varphi_{A2} + 7 \cdot 3\varphi_{A1}\varphi_{a2} + 3^2\varphi_{a1}\varphi_{a2}.$$

Из этого следует, что отношение численностей организмов по фенотипам A_1A_1 и A_2A_2 , a_1a_1 и A_2A_2 , a_2a_2 и A_1A_1 , a_1a_1 и a_2a_2 в третьем поколении есть $7^2:7 \cdot 3:7 \cdot 3:3^2$.

В четвертом поколении отношение численностей организмов по этим же феноменам получается аналогично при следующем расщеплении скобок в выражении (3)

$$15^2:15 \cdot 7:15 \cdot 7:7^2.$$

Конструкция этих отношений в следующем поколении сохраняются. Числа, участвующие в отношениях (1,3,7,15...), это есть числители и знаменатели (6) при соответствующих значениях n . Показатели и число повторяющихся произведений в отношениях указывают число независимых пар признаков (в данном случае 2). Это приводит к следующим отношениям численностей организмов по фенотипам с одновременным присутствием двух пар признаков при произвольном номере поколения n

$$(2^n - 1)^2 (2^4 - 1)(2^{n-1} - 1) : (2^n - 1)(2^{n-1} - 1) : (2^{n-1} - 1)^2. \quad (7)$$

Полигибридное скрещивание

Если скрещиваются организмы с числом различающихся пар признаков больше чем 2, например, t пар, то распределение организмов в потомствах можно получить из произведения t множителей

$$(A_1A_1 + 2A_1a_1 + a_1a_1)(A_2A_2 + 2A_2a_2 + a_2a_2) \dots (A_tA_t + 2A_ta_t + a_ta_t). \quad (8)$$

Если представляет интерес отношение численностей организмов фенотипов по доминантному и рецессивному членам одной из пар признаков i ($1 \leq i \leq t$), то во втором поколении произведение (8) можно записать в виде $(A_iA_i + 2A_ia_i + a_ia_i)f$, где f включает все остальные сомножители произведения (8). Из этого выражения следует, что во втором поколении отношение численностей организмов по доминантным и рецессивным генам пары i есть отношение 3:1. При расщеплении гетеризиготных организмов A_ia_i с образованием третьего поколения отношение численностей организмов по фенотипам A_ia_i и a_ia_i есть 7:3 и т.д.

Поэтому отношение численностей организмов в n -ом поколении по фенотипам $A_i A_i$ и $a_i a_i$ при полигибридном скрещивании имеет такой же вид, как и при бигибридном скрещивании

$$\delta_t(n) = \frac{2^n - 1}{2^{n-1} - 1}. \quad (9)$$

Это отношение относится к любой паре признаков i из t пар в силу их независимости.

Если необходимо рассмотреть отношение численностей организмов по фенотипу с одновременным присутствием нескольких пар признаков, то последовательно расщеплять все скобки в произведении (8). Например, в случае трех пар признаков ($t=3$), используя обозначения, введенные в предыдущем разделе, произведение (8) принимает вид

$$(3\varphi_{A1} + \varphi_{a1})(3\varphi_{A2} + \varphi_{a2})(3\varphi_{A3} + \varphi_{a3}) = 27\varphi_{A1}\varphi_{A2}\varphi_{A3} + 9\varphi_{a1}\varphi_{A2}\varphi_{A3} + 9\varphi_{A1}\varphi_{a1}\varphi_{A3} + 9\varphi_{A1}\varphi_{A2}\varphi_{a2} + 3\varphi_{A3}\varphi_{a1}\varphi_{a2} + 3\varphi_{a1}\varphi_{A2}\varphi_{a3} + 3\varphi_{A1}\varphi_{a2}\varphi_{a3} + \varphi_{a1}\varphi_{A2}\varphi_{a3}. \quad (10)$$

Из равенства (10) следует, что в случае гибридизации с тремя парами признаков во втором поколении отношение между численностями организмов по фенотипам различных возможных пар генов имеет вид

$$3^3:3^2:3^2:3^2:3:3:3:1.$$

В третьем поколении произведение (8) имеет вид

$$(7\varphi_{A1} + 3\varphi_{a1})(7\varphi_{A2} + 3\varphi_{a2})(7\varphi_{A3} + 3\varphi_{a3}) = 7^3\varphi_{A1}\varphi_{A2}\varphi_{A3} + 7^2 \cdot 3\varphi_{A1}\varphi_{A2}\varphi_{a3} + 7^2 \cdot 3\varphi_{A1}\varphi_{a2}\varphi_{A3} + 7 \cdot 3^2\varphi_{a1}\varphi_{A2}\varphi_{A3} + 7 \cdot 3^2\varphi_{a1}\varphi_{a2}\varphi_{A3} + 7^3 \cdot 3\varphi_{a1}\varphi_{A2}\varphi_{a3} + 7 \cdot 3^2\varphi_{A1}\varphi_{a2}\varphi_{a3} + 3^3\varphi_{a1}\varphi_{a2}\varphi_{a3}.$$

Из последнего выражения следует, что отношение между численностями организмов с различными генами при скрещивании организмов с тремя парами признаков в третьем поколении имеет вид

$$7^3:(7^2 \cdot 3):(7^2 \cdot 3):(7^2 \cdot 3):(7 \cdot 3^2):(7 \cdot 3^2):(7 \cdot 3^2):3^3.$$

Отсюда можно получить соответствующие отношения численностей организмов при скрещивании организмов с t парами признаков в n -ом поколении.

Индексируя повторяющиеся произведения числом пар признаков эти отношения имеют вид

$$\begin{aligned} & (2^n - 1)^t : [(2^n - 1)^{t-1}(2^{n-1} - 1)]_1 : [(2^n - 1)^{t-1}(2^{n-1} - 1)]_2 : \dots \\ & : [(2^n - 1)^{t-1}(2^{n-1} - 1)]_t : [(2^n - 1)^{t-2}(2^{n-1} - 1)^2]_1 : [(2^n - 1)^{t-2} \\ & (2^{n-1} - 1)^2]_2 : \dots : [(2^{n-1} - 1)^{t-2}(2^{n-1} - 1)^2]_t : \dots \\ & : [(2^n - 1)(2^{n-1} - 1)^{t-1}]_1 : [(2^n - 1)(2^{n-1} - 1)^{t-1}]_2 : \dots : [(2^n - 1)(2^{n-1} - 1)^{t-1}]_t \\ & : [(2^{n-1} - 1)^t]. \end{aligned}$$

Выводы

Показано, что закономерности наследования признаков организмов при их скрещивании основаны на специального рода математических соотноше-

ниях, позволяющих алгебраическим путем вывести экспериментальные закономерности, изученные еще 150 лет назад Г.Менделем. Ядром этих соотношений являются геометрические прогрессии. С их помощью можно предсказывать результаты скрещивания организмов вне интервалов исходных данных экспериментальных работ Г.Менделя. В частности, при любом числе пар константно-различающихся признаков и любом числе поколений.

Получены общие формулы для отношений количеств организмов при полигибридном скрещивании, отличающихся фенотипом, для любого поколения. Эти формулы непрерывно переходят в соотношения Г.Менделя в области условий проведенных им экспериментов.

Библиографический список

1. Гайсинович А.Е. Зарождение и развитие генетики. М.:Наука, 1988, 423 с.
2. Иоганнсен В.Л. Элементы точного учения об изменчивости и наследственности с основами биологической вариационной статистики. Л.:Сельхозгиз, 1933, 410 с.
3. Кольцов Н.К. Наследственные молекулы. Наука и жизнь, 1935,N5, с.4-13.
4. Мендель Г. Опыты над растительными гибридами. М.:Наука,1965, 66 с.
5. Морган Т.Г. Избранные работы по генетике. М.-Л.:Сельхозгиз, 1937, 285 с.
6. Четвериков С.С. О некоторых моментах эволюционного процесса с точки зрения современной генетики. Журн.эксперим.биологии.Сер.А.1926, т.2,с.3-54.
7. Villee C.A., Dethier V.G. Biological Principles and Processes Philadelphia-London-Toronto, W.B.Saunders Company, 1971, 822 с.
8. Weismann A. Die Continuitat des Keimplasmas als Grundlage einer Theorie der Vererbung.Jena, Fisher, 1885, 112 p.
9. Watsons D., Crick H.C. Molecular structure of nucleic acids.Nature, 1953, v.171, pp.738-740.
10. Zhizhin G.V. The structure, Topological dimension of biomolecules. J.Chemoinformatics and Chemical Engineering, 2016, 5(3), 44-58.
11. Zhizhin G.V. Chemical Compound Structures and the Higher Dimension of Molecules: Emerging Research and Opportunities.2017. Heishey PA, USA, 221 с.
12. Zhizhin G.V. Dimensions of compounds in supramolecular chemistry. Int. J. Chem. Modeling. 2015,7(2), 215 – 238.
13. Zhizhin G.V., Diudea M.V. . Space of nanoworld. In M.V. Putz, C.M. Marius (Eds.), Sustainable nanosystems, development, properties, and applications. 2016, New York: IGI Global. pp. 221 – 245.
14. Zhizhin G.V. About the True Equilibrium in the Population of the Plant. Journal of Biochemistry and Molecular Biology Research, 2018,3(1), 206 - 224.

СИСТЕМА ЭЛЕКТРОННОГО ДОКУМЕНТООБОРОТА И УПРАВЛЕНИЯ ВЗАИМОДЕЙСТВИЕМ «DIRECTUM» в СПбГЛТУ

Активное применение цифровых технологий во всех сферах профессиональной деятельности предполагает их активное использование и в образовании в виде электронного и дистанционного обучения, а также электронного документооборота.

Переход на электронную форму документооборота предполагает использование новых подходов в формировании, перемещении, использовании и контроле информации.

Эволюционное внедрение и развитие этого процесса в вузах имеют положительные предпосылки, заключающиеся в том, что:

- в вузах давно и достаточно успешно эксплуатируются системы автоматизации образовательного процесса и управления всеми видами деятельности;
- использование средств автоматизации поддерживается аппаратно - программными компонентами инфокоммуникационных систем вузов;
- в настоящее время в вузах сформировались коллективы преподавателей и сотрудников выросших и получивших образование в годы интенсивного внедрения информационных технологий (ИТ), они активно используют компьютеры и самостоятельно занимаются разработкой необходимых им электронных информационных документов для профессиональной деятельности.

Всем этим требованиям отвечает отечественная система электронного документооборота (СЭД) Directum) развернутая в СПбГЛТУ [1]. Система электронно объединяет все здания университета, колледж и филиалы вуза. Она обладает необходимым функционалом и обеспечивает реализацию следующих операций:

- обработку и регистрацию документов, рассмотрение входящей корреспонденции, выдачу поручений и контроль исполнения, ознакомление с документами, поиск и контроль местонахождения;
- выдачу заданий сотрудникам, контроль хода их исполнения и проверка выполненных работ с помощью механизма поручений;
- быстрый поиск любого документа по реквизитам, тексту содержимого или штрихкоду, а бумажного экземпляра с помощью специального сканера;
- создание электронных документов на основе готовых шаблонов или получение их из почты или сервиса обмена;
- хранение документов в едином хранилище, обеспечивающем пользователям работу со ссылками на документы и сохранение их в разных папках,

структура папок задается исходя из организационной структуры, процессов или личных потребностей сотрудника.

- создание неограниченного количества версий позволяющих отслеживать изменения в документе и появление дублей.

Тестовая эксплуатация СЭД Directum показала, возможность ее интегрирования в электронную информационно-образовательную среду СПбГЛТУ.

При этом отмечается относительная независимость реализации функций СЭД от операционной платформы (аппаратной и программной), то есть возможность ее функционирования в мультиплатформенной среде корпоративной информационной системы СПбГЛТУ, компоненты которой отличаются своей архитектурой, функциональностью и характеристиками, территориальной разнесенностью рабочих станций и серверов СЭД, предоставлением пользователям и приложениям возможность работы в системе, независимо от того, где, как и когда происходит их взаимодействие.

Проявилась способность СЭД и процессов, реализуемых в ней, адаптироваться к увеличению рабочей нагрузки путем добавления ресурсов без полной или частичной ее реструктуризации.

Таким образом, короткий опыт эксплуатации СЭД Directum в СПбГЛТУ показывал, что она обеспечивает хранение, поиск, обработку и выдачу информации в виде совокупности фактов, документов, данных и знаний, отражающих реальное и изменяющееся во времени состояние как в вузе, так и вне его. В дальнейшем рассматривается возможность удаленного доступа пользователей к системе, в том числе и с мобильных устройств.

Распределенность архитектуры СЭД обеспечивает: фактически неограниченное количество организационно и физически распределенных пользователей, одновременно работающих с общими данными и располагающимися на различных рабочих станциях, с различными полномочиями и задачами, независимо от того как, где и когда происходит их взаимодействие, единообразную работу пользователей и приложений в системе, которая относительно легко масштабируется.

Библиографический список

1. Системы электронного документооборота и управления бизнес процессами. <https://directum.ru>
2. Заяц А.М. Электронная информационно-образовательная среда – платформа агрегации средств управления образовательным процессом ВУЗа, информационных образовательных ресурсов и технологий. // Информационные системы и технологии: теория и практика: сб. научных трудов - СПб.: СПбГЛТУ, 2018, №10, с. 5-14.

WEBSOCKET КАК ИНСТРУМЕНТ ДЛЯ УДАЛЁННОГО УПРАВЛЕНИЯ

WebSocket разработан для использования в веб-браузерах и веб-серверах, но он может быть использован для любого клиентского или серверного приложения [1.2]. Протокол WebSocket — это независимый протокол [3], основанный на протоколе TCP, он делает возможным более тесное взаимодействие между браузером и веб-сайтом, способствуя распространению интерактивного содержимого и созданию приложений реального времени. Он обеспечивает двунаправленность потоков данных через Интернет и постоянную связь между клиентом и сервером. Таким образом, этот протокол становится оптимальным решением для построения распределённых систем.

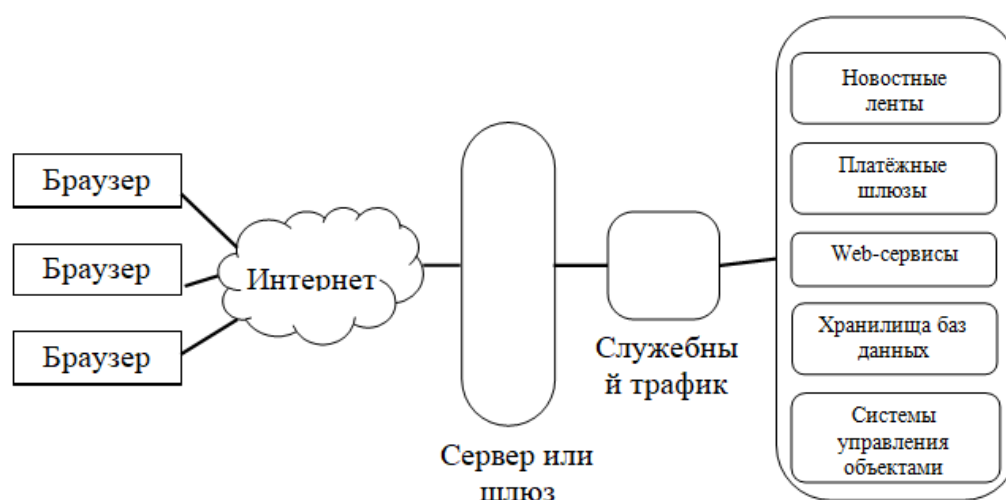


Рис 1. Пример работы websocket приложения

Постановка задачи и объект исследования. Необходимо разработать надёжную и эффективную систему управления элементами умного дома. В данной статье рассмотрены возможности построения web-приложения для удалённого управления какими-либо объектами на базе протокола websocket [4]. Протокол позволяет производить подключение к серверу сразу большому количеству клиентов, соответственно, нужно разработать такую систему, которая будет отображать актуальный статус работы объектов.

При этом система должна обеспечивать управление объектов используя для этого лишь стандартный браузер (рис 1). В рассматриваемом примере речь пойдёт о команде включения и выключения электроприборов, таких как: кухонная техника, предметы домашнего интерьера. Серверная часть обработки websocket запросов будет использовать утилиту websocketd, которая обеспечивает передачу запросов к серверному коду на языке C#, и в тоже время формирует исходящие сообщения от сервера для браузеров клиентов,

интерфейс которых представляет собой html-страницу с набором элементов управления и скриптов на javascript.

Описание работы серверной части. Сервером, в данном случае, выступит обычный компьютер с запущенной на нём утилитой websocketd [5]. К этому серверу с помощью Wi-Fi модуля [6] подключен объект управления электропитанием, который в зависимости от поданных на него сигналов, либо подаёт электропитание, либо отключает его. Стоит учесть, что при создании приложений подобного рода серверная часть должна отслеживать все изменения на каждом из клиентов, и постоянно обновлять состояния объектов в зависимости от изменений состояния этих объектов. Для этой задачи используется скрипт на стороне клиента. Он обновляет параметры каждые 5 секунд, а так же, каждый раз при включении или выключении какого-либо устройства скрипт проверяет статус устройства в текущий момент времени, и если они совпадают с желаемым, то сообщает об этом пользователю.

Для доступа к компьютеру из внешней сети ему был присвоен статический ip-адрес, при переходе на который через браузер нам откроется страница управления элементами умного дома. Ниже приведён фрагмент кода на языке C#, в котором сервер начинает свою работу, затем прослушивает входящие к нему подключения. В зависимости от прибора, который захочет включить пользователь, на клиентской стороне браузер отправит серверу ту или иную команду. Например, если требуется включить свет, то клиент посылает команду «LightOn». Получив эту команду, сервер запускает событие «Action.LightOn()». Фрагмент кода сервера приведен ниже.

```
// Создание TCP/IP сокета.
Socket listener = new
Socket (IPAddress.AddressFamily,
        SocketType.Stream, ProtocolType.Tcp );
// Сервер прослушивает подключения
try {
    listener.Bind(localEndPoint);
    listener.Listen (10);
    // Сервер принимает запрос на подключение
    while (true) {
        Console.WriteLine ("Waiting for a connection...");
        Socket handler = listener.Accept();
        data = null;
        // Входящие подключения начинают обрабатываться
        while (true) {
            int bytesRec = handler.Receive(bytes);
            data += Encoding.ASCII.GetString(bytes, 0, bytesRec);
            if (data.Equals("<LightOn>") ) {Action.LightOn();}
            . . .
            if (data.Equals("<exit>") ) { break; }
            . . .
        }
    }
}
```

Описание работы клиентской части. Клиентская часть реализована на языке html и javascript. Работоспособность этой страницы во многом зависит от поддержки конкретным браузером протокола websocket. Рабочая страница содержит список доступных устройств умного дома и элементы управления для включения или отключения их функций, интерфейс управления представлен на рис. 2.

Подключение к websocket серверу происходит автоматически в момент загрузки главной страницы интерфейса пользователя. Если подключение не произойдет, то будет выдано сообщение о недоступности сервера. Для подключения из локальной сети используется локальный адрес компьютера сервера в этой сети - 192.168.31.103, для доступа к этому же серверу, но из внешней сети должен использоваться статический адрес выданный Интернет-провайдером – 93.92.206.67.

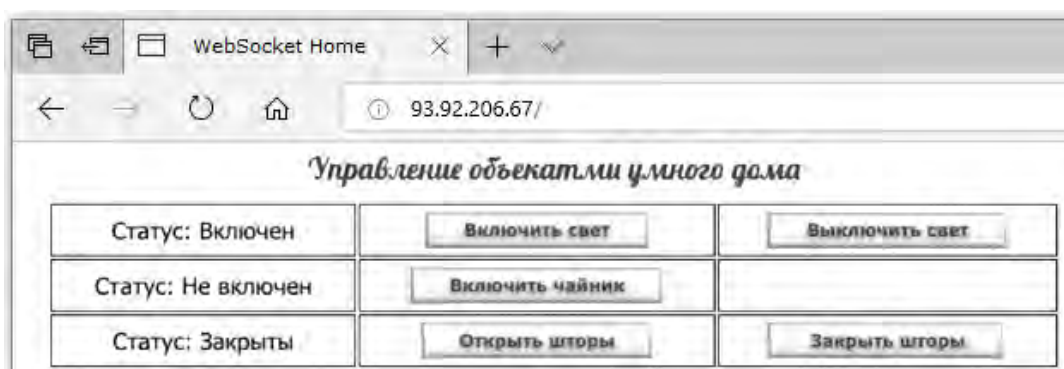


Рис 2. Интерфейс управления для пользователей

Заключение. Таким образом, с помощью предложенной технологии, возможно организовать управление любым объектом, имеющим связь с сервером, необходимо лишь добавить изменения в интерфейс и программный код сервера. Большое количество входящих подключений, а так же отсутствие ограничений в виде и количестве устройств позволяют использовать приложения подобного рода практически в любой сфере.

Система приложений, рассмотренных в статье, свидетельствует о простоте, надёжности и эффективности применения протокола websocket при работе с подобными задачами. Протокол обеспечивает короткие сессии передачи информации без лишнего трафика, и позволяет быстро передать нужную информацию в текущий момент времени. Разработанное приложение может быть усовершенствовано путем добавления авторизации и прав доступа. Такие меры помогут ограничить управление конкретными объектами или ввести режим управления нужными объектами.

Библиографический список

1. Хабаров С.П., Заяц А.М. Использование технологии websocket в клиент-серверных экспертных системах. //В книге: Леса России: политика, промышленность, наука, образование Материалы второй Международной научно-технической конференции. 2017, с. 278-280.

2. Хабаров С.П. Взаимодействие узлов сети по протоколу WebSocket. // Сборник научных трудов "Информационные системы и технологии: теория и практика" — СПб.:СПбГЛТУ, 2017. № 9. — с.109-119
3. WebSocket [Электронный ресурс] // Wikipedia 2019. URL: <https://ru.wikipedia.org/wiki/WebSocket>,
4. Хабаров С.П., Красовский И.А., Киев А.Х.-А. Удаленное управление на базе технологии websocket. // Сборник научных трудов "Информационные системы и технологии: теория и практика" — СПб.:СПбГЛТУ, 2017. № 9. — с.124-130
5. Хабаров С.П. Использование утилиты websocketd для удаленного выполнения программ. // Сборник научных трудов "Информационные системы и технологии: теория и практика" — СПб.:СПбГЛТУ, 2017. № 9. — с.94-108
6. Сусь К.М., Хабаров С.П. Мониторинг беспроводной Wi-Fi сети с использованием сетевого монитора Microsoft Network Monitor. // Информационные системы и технологии: теория и практика: сб. научн. тр., Вып 10. Ч. 2. — СПб.:СПбГЛТУ, 2018. — с.73-82

Е.В. Тетюев, студент
СПбГЛТУ им.С.М.Кирова
hooves.xyz@gmail.com

Ю.А. Жук кандидат педагогических наук, доцент
Кафедра информационных систем и технологий
СПбГЛТУ им.С.М.Кирова
zhuk_yua@mail.ru

ИСПОЛЬЗОВАНИЕ ТРЁХМЕРНОЙ ГРАФИКИ НА ОСНОВЕ WEBGL ДЛЯ СОЗДАНИЯ ОБУЧАЮЩИХ И ДЕМОСТРАЦИОННЫХ МАТЕРИАЛОВ

Введение

Трудно представить себе современный мир без компьютерных технологий или без Интернета. Сегодняшние школьники ходят с мобильными телефонами, которые потом меняют на яркие смартфоны, многие семьи имеют компьютер или планшет. Эти вещи уже крепко засели в наш быт, и есть практически в любой отрасли. Информатизация не прошла и мимо образовательной сферы: компьютерные технологии сильно оживили процесс обучения, упростили его и сделали гораздо доступнее. Студент, конспектирующий лекции на ноутбуке, уже не вызывает удивления, как и преподаватели, которые используют на лекциях проектор. Использование цифровых наглядных пособий стало отличным дополнением к «классическим» лекциям.

Однако, презентации, схемы и диаграммы представляют собой лишь набор статических изображений с минимальными возможностями для взаи-

модействия. На их фоне явно выделяются обучающие материалы, реализованные с помощью HTML и JavaScript, выгодно отличаясь своими интерактивными возможностями. Сайты дистанционного обучения и тестирование онлайн становятся всё популярнее, поэтому дальнейшее использование веб-браузеров и их возможностей видится перспективным направлением развития. Одной из таких возможностей является рендеринг трёхмерной графики прямо в браузере.

Стандарт WebGL и .js библиотеки

Какие ассоциации вызывает слово у нас слово «OpenGL»? Рядовые пользователи вряд ли о таком слышали. Поклонники старых компьютерных игр вспомнят, что эта опция когда-то встречалась в настройках. Разработчики знают правильный ответ: OpenGL – программный интерфейс для написания приложений, использующих двумерную и трёхмерную компьютерную графику. То же самое и с WebGL[1]: многие знают, что это такое, но сама технология гораздо ближе, чем кажется. Что же такое WebGL?

WebGL (Web-based Graphics Library) — кроссплатформенный интерфейс прикладного программирования (API) для отображения 3D-графики в браузере, разрабатываемый некоммерческой организацией Khronos Group[1,2]. API построен на основе OpenGL и даёт возможность реализации 3D-графики прямо в браузере, используя HTML5-элемент canvas и взаимодействуя с объектной моделью документа (DOM). Таким образом, использование WebGL позволяет перейти от двумерной графики к полноценной трёхмерной визуализации, а средства JavaScript открывают огромные возможности для интерактива.

Для упрощения разработки WebGL-приложений существуют различные библиотеки. Первой общедоступной стала библиотека WebGLU. Со временем количество библиотек выросло, а самые популярные из них - Three.js, PlayCanvas и BabylonJS - распространяются по лицензии открытого программного обеспечения MIT[2]. Библиотеки, обладая не самым большим весом, сильно упрощают будни разработчика. Например, код для вставки трёхмерной сцены в HTML с помощью Three.js до невозможного прост (рис.1):

```
<script type="text/javascript"
src="../../libs/three.js">
</script>
<script type="text/javascript"
src="../../libs/jquery-1.9.0.js">
</script>
<div id="WebGL-output">
</div>
```

Рис. 1. Три строки для создания пустой, но полностью рабочей сцены

А вот пример того, что будет, если добавить к этому коду пару строк с описанием простейших геометрических объектов (рис.2) [3].

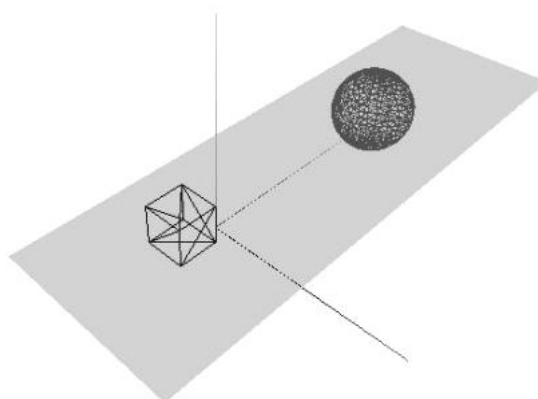


Рис. 2. Простейшая сцена из геометрических примитивов: сферы, куба, плоскости и осей x, y, z .

Примеры, возможности и направления

WebGL создавался для решения более сложных и интересных задач. Как было сказано ранее, примеры использования этой технологии гораздо ближе, чем может показаться. Наверняка, каждый из нас хоть раз в жизни использовал сервисы Яндекс или Google карт [4]. Если увеличить масштаб карты до максимального, на ней будут видны объёмные модели зданий. А что будет, если в `maps.google` сильно уменьшить масштаб? Карта постепенно выгибается, и в наименьшем масштабе пользователь видит на экране земной шар в виде полноценной трёхмерной сферы с текстурой (рис. 3).



Рис. 3. Земля на картах Google в минимальном масштабе

Пример показывает, что WebGL может использоваться в повседневной жизни, причём пользователь необязательно догадывается об этом. А вот пример того, как WebGL может использоваться в профессиональной сфере (рис.4) [5].



X-TAON: Rally Car #XTAONartcarcontest ©

Рис. 4. Модель машины на сайте SketchFab

Сайт представляет собой хранилище и торговую площадку для разработчиков 3D моделей. Одна из функций – просмотр моделей прямо в браузере. Добавим ещё пару примеров для наглядности и демонстрации возможностей WebGL (рис. 5, 6) [6, 7].

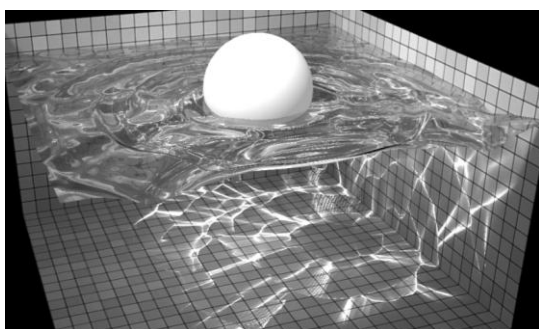


Рис. 5. Демонстрация работы освещения и текстур.




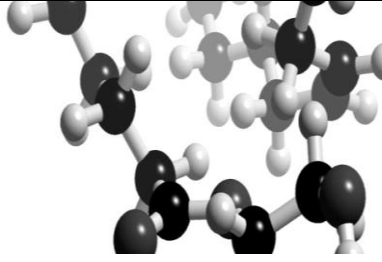

Рис. 6. Пример использования сплайнов и трёхмерного текста [7]

Основные возможности использования WebGL.

- Загрузка и отображение объектов и сцен, как простой геометрии, так и сложных объектов в формате obj, json;
- Взаимодействие с объектами в пределах загруженной сцены: перемещение, вращение, масштабирование и т.д.;
- Работа с текстурными картами, поддержка составных многослойных материалов и основных свойств, таких как opacity, bump, reflection, glossiness и др.;
- Работа с источниками света, поддержка теней, glow-эффектов и тумана;
- Наличие вспомогательных средств, например, построение лучей, отрезков, отслеживание пересечений объектов и т.д.

Таким образом, использование модуля ограничено лишь фантазией разработчика (заказчика) и решением поставленной задачи. В каких же случаях использование трёхмерной графики может быть полезнее и нагляднее двумерной графики?

В первую очередь, это любые предметы, где нужна наглядность и возможность посмотреть на объект с нескольких ракурсов. Вот несколько простейших примеров: детали машин и механизмов, химические формулы или процессы, а также объекты строительства и ландшафтный дизайн (рис. 7–9).

		
<p>Рис. 7. Трёхмерная модель подшипника</p>	<p>Рис. 8. Трёхмерная модель химического соединения</p>	<p>Рис. 9. Трёхмерная модель участка парка</p>

Отдельно стоит отметить электронное обучение, которое активно набирает популярность. Поскольку обучающиеся сильно ограничены в плане исследуемых материалов, возможность взять какой-либо объект и рассмотреть его в подробностях со всех сторон может существенно способствовать процессу обучения. Интерактивная 3D модель отлично справится с этой задачей.

Достоинства и недостатки

Конечно, как и любой API, WebGL имеет свои плюсы и минусы. Так основными достоинствами являются:

- простота использования для конечного пользователя: нет необходимости устанавливать специальное приложение или среду для просмотра;
- простота исполнения: встраивание в HTML-страницу позволяет вставлять несложные трёхмерные сцены и модели прямо в текст, как обычные картинки;
- кроссбраузерность и кроссплатформенность: WebGL поддерживается Chrome, Firefox, Safari, Opera, IE версии 11 и выше, в «революционном» Edge, а также во многих мобильных браузерах;
- использование JavaScript позволяет создать полноценную интерактивную среду и тесное взаимодействие между элементами трёхмерной сцены и остальными элементами интернет страницы.

Из недостатков можно отметить:

- в связи с ограничениями политики безопасности у браузеров, для загрузки трёхмерных объектов рекомендуется использование локального или интернет сервера;
- при использовании интернет сервера загрузка сложных объектов может потребовать значительного времени и интернет трафика;
- браузерная реализация накладывает ограничения на количество и сложность моделей. Разработчикам следует воздержаться от избыточно сложных объектов и стремиться к оптимизации моделей и сцен. Для сцен с множеством объектов идеальным решением будет красочный и низкополигональный минимализм.

Пример использования технологии

В качестве примера, для данной статьи была разработана простейшая программа, которая наглядно демонстрирует порядок и схему сборки системного блока компьютера. Программа представляет собой список возмож-

ных комплектующих с индикацией установки и окно для вывода трёхмерного изображения. При клике на любой из элементов списка, он будет установлен, снят или останется без изменений (в зависимости от уже установленных модулей), при этом процесс установки или демонтажа отобразится в графике, наглядно показывая место крепления компонента.

Для модели реализована возможность повернуть камеру, а также возможность приблизить/отдалить модель, чтобы рассмотреть сцену с разных ракурсов (рис. 10).

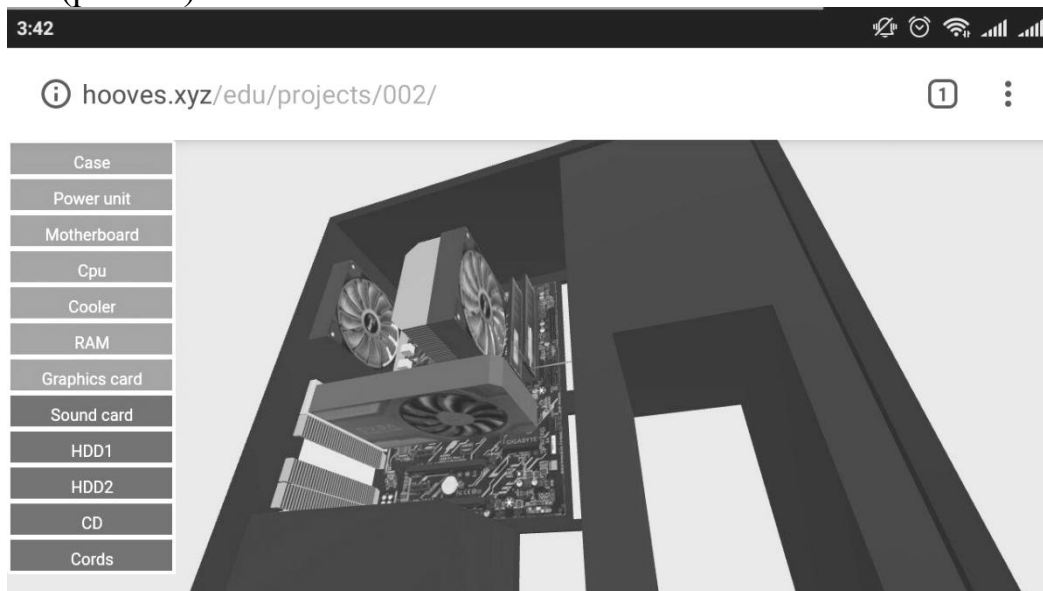


Рис. 10. Интерфейс обучающей мини-программы.
Скриншот с мобильного устройства

Программа [8] реализована на базе HTML-страницы, взаимодействие с WebGL идёт на основе библиотеки Three.js. Сцена выглядит достаточно простой, и это абсолютно верно. Однако, сама идея не так принципиальна, как возможности её использования. Усложняя задачу, можно создавать программы, в которых будет больше деталей, больше интерактивных возможностей, лучше наглядность, и т.д.

Программа доступна для ознакомления по адресу:
hooves.xyz/edu/projects/002/

Отдельно стоит отметить, что веб-страница, файл модели, текстуры и необходимые библиотеки в сумме занимают около 1300Кб, что весьма скромно, особенно с учётом того, что данная программа запускается без установки и требует для работы только браузер и более-менее современный компьютер или телефон. Код программы без труда встанет в HTML-статью об устройстве компьютера, наряду с двумерными изображениями, и станет отличным дополнением к визуальному ряду.

Заключение

Простейший пример показывает, насколько удобной и востребованной может быть данная технология. С точки зрения разработчика, кроссплатформенность и кроссбраузерность однозначно говорят в пользу WebGL, а код, основанный на Javascript, упрощает понимание работы модулей и компонентов, тем самым открывая перспективы для создания сложных интерактивных

приложений, которые можно использовать в качестве полноценного учебного материала.

Таким образом, использование WebGL и трёхмерной графики в целом расширяет горизонты привычных методов преподавания, а также делает процесс обучения гораздо интереснее и доступнее для обучаемых. Кроме того, использование технологии позволяет образовательным учреждениям предоставлять качественные образовательные услуги и на расстоянии с помощью дистанционных технологий. Перспективы использования 3D уже сейчас просто поразительны и хорошо видны на примерах, представленных в статье. Очевидно, что дальнейшее развитие мультимедиа устройств и устройств вывода информации положительно скажется как на технологии в целом, так и на возможностях для её применения.

Библиографический список

1. WebGL // Khronos Group // URL: <https://www.khronos.org/webgl/>
2. Web-based Graphics Library // Википедия // URL: <https://ru.wikipedia.org/wiki/WebGL>
3. Изучаем Three.js - Создание сцены // habr URL: <https://habr.com/ru/post/224509/>
4. Google карты // URL: <https://www.google.com/maps>
5. Sketchfab URL: <https://sketchfab.com/>
6. Пример физики воды и освещения // URL: <http://madebyevan.com/webgl-water/>
7. Пример с базовой геометрией и отрисовкой примитивов // Armsglobe URL: <http://armsglobe.chromeexperiments.com/>
8. Программа-пример: сборка компьютера // URL: <http://hooves.xyz/edu/projects/002/>

М.П. Филяев, доктор технических наук
Кафедра информационных систем и технологий
СПбГЛТУ им. С.М. Кирова,
mastkon@yandex.ru

А.С. Якшин, кандидат технических наук
Научно-исследовательский институт ВА МТО имени генерала армии А.В. Хрулёва,
yakshin_as@mail.ru

АКТУАЛЬНЫЕ ВОПРОСЫ ИМИТАЦИОННОГО ИССЛЕДОВАНИЯ ТРАНСПОРТНЫХ СИСТЕМ

Современные экономические реалии и проблемы обеспечения безопасности государства ставят перед российской транспортной системой важные задачи, связанные не только с развитием транспортных коммуникаций, но и

повышением эффективности перевозочных процессов. Для Российской Федерации, территория которой раскинулась от Балтийского и Черного морей до Тихого океана, транспорт является определяющим фактором развития производственного потенциала страны, организации международных торговых связей и повышения транспортной мобильности населения.

Для решения задач, сформулированных в Транспортной стратегии Российской Федерации на период до 2030 года [1], необходим целостный взгляд на существующую транспортную систему и ее взаимосвязь с экономикой страны. Эффективное решение как общей проблемы прогнозирования траектории развития транспортной системы, так и частных задач, связанных с организацией перевозочных процессов с учетом значительного числа составляющих их элементов и причинно-следственных связей, невозможно без использования современных информационных систем и технологий. При этом основным направлением их применения является создание и внедрение в управление транспортными процессами систем поддержки принятия решений (СППР).

СППР – это компьютерная система, которая помогает лицу, принимающему решение, в условиях неопределенности в процессе прямого взаимодействия с использованием данных и моделей [2]. Человеко-машинная процедура принятия решений с помощью СППР представляет собой циклический процесс взаимодействия человека и компьютера. Цикл состоит из фазы анализа и постановки задачи для компьютера и фазы оптимизации (поиска решения и обеспечения его характеристик), реализуемой компьютером. Таким образом, СППР обеспечивают следующее:

- Помогают произвести оценку обстановки (ситуаций), осуществить выбор критериев и оценить их относительную важность.
- Генерируют возможные решения (сценарии действий).
- Осуществляют оценку сценариев (действий, решений) и выбирают лучший.
- Моделируют принимаемые решения (в тех случаях, когда это возможно).
- Осуществляют динамический компьютерный анализ возможных последствий принимаемых решений.
- Производят сбор данных о результатах реализации принятых решений и осуществляют оценку результатов.

Таким образом, построение и применение средств поддержки принятия решений в транспортных системах (ТС) основывается на моделировании протекающих в них процессов. В современных условиях функционирования ТС, содержащих значительное число элементов, поведение каждого из которых может описываться довольно сложными аналитическими соотношениями, их адекватное моделирование фактически невозможно без применения методов имитационного моделирования [3].

Для уточнения роли, места и особенностей имитационного моделирования в исследовании ТС конкретизируем понятия «транспортной системы» и «транспортного процесса».

Являясь сложной организационно-технической системой, ТС может быть рассмотрена как совокупность связанных потоков транспортных средств, пассажиров и грузов, взаимодействующих в пространстве и времени на основе транспортной инфраструктуры с учетом реализуемых алгоритмов функционирования.

В наиболее общем случае, транспортный процесс – это совокупность операций с ресурсами и транспортными средствами, в результате выполнения которых ресурсы изменяют своё положение в пространстве. В качестве ресурсов могут рассматриваться грузы (материальные средства) и людские ресурсы (пассажиры). Сущность транспортной работы заключается в изменении места нахождения ресурсов. Структура транспортного процесса включает три основных элемента (подпроцесса, этапа): погрузку, перевозку и разгрузку. Некоторые транспортные процессы могут включать дополнительно этап перегрузки ресурса. Тем не менее - основной элемент транспортного процесса – перевозка, все другие элементы подчинены ему. Перевозочный процесс включает работу подвижного состава с момента подачи под погрузку, его движение с ресурсом до постановки под разгрузку. Процессы погрузки и разгрузки состоят из возможного ожидания погрузки (разгрузки) и обслуживания. Ожидание погрузки (разгрузки) грузов может быть связано с опозданием транспортных средств, занятостью погрузочно-разгрузочных средств и выделяемых для выполнения этих операций мест (участков) транспортной инфраструктуры. Обслуживание включает собственно погрузку (разгрузку), а также оформление документов, если эта операция полностью не осуществляется во время ожидания погрузки (разгрузки) и обслуживания. Транспортные процессы инициируются и выполняются в соответствии с потребностями промышленности, аграрного производства, обороны и отдельного человека (групп людей). Транспортный процесс является также одним из видов производственного процесса и включает операции приема, транспортировки, хранения, перевалки, сдачи. В транспортном процессе, в отличие от промышленного процесса, нового материального продукта не производится. Каждая операция, как и процесс, состоит из предметов труда, средств производства и производственного персонала, взаимодействие которых направлено на создание транспортной продукции:

- предмет транспортного производства – перевозка ресурсов;
- средства транспортного производства – транспортное пространство и транспортная техника;
- исполнители транспортного производства – персонал транспортного производства.

Таким образом, в процессе моделирования транспортных систем решаются следующие основные задачи: формирование прогнозов транспортной ситуации в зависимости от изменения внешних и внутренних параметров системы; подготовка рекомендаций для оптимальной организации и управ-

ления дорожным движением (перевозочным процессом), обоснование решений органами управления об инвестициях в области транспортной инфраструктуры для повышения эффективности функционирования ТС и др. В связи с этим актуально рассмотрение следующих проблемных вопросов, связанных с указанными задачами, и определение основных направлений их решения.

Выбор подхода к имитационному моделированию транспортной системы. Современное состояние методологии имитационного моделирования свидетельствует о применении для исследования сложных систем в основном трех исторически сложившихся подходов: системно-динамическом, дискретно-событийном и агентном. Использование того или иного подхода определяется особенностями функционирования исследуемой системы, что было рассмотрено в работах [4, 5, 6] и представлено на рис. 1.

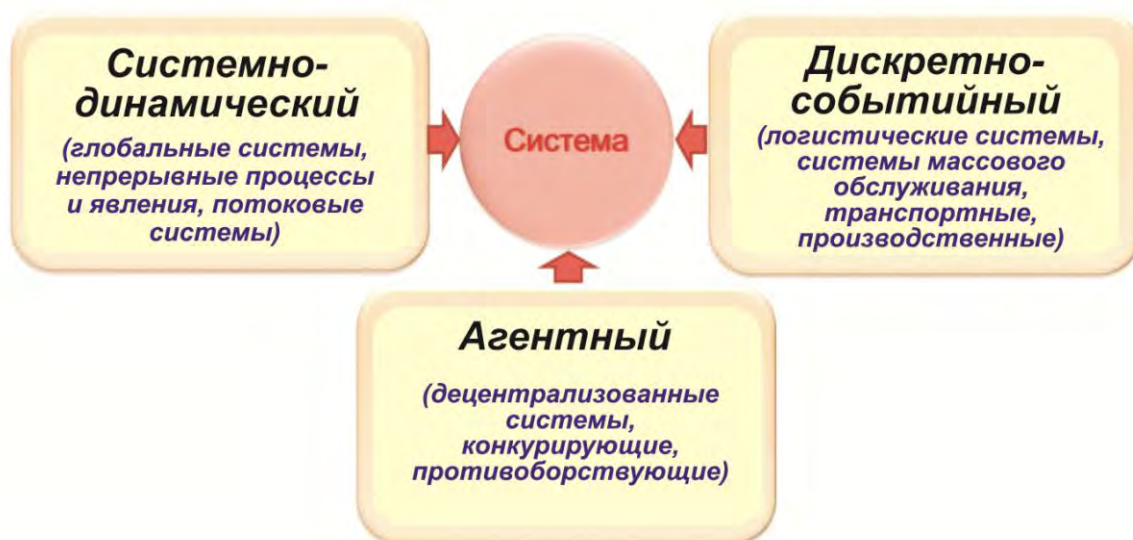


Рис.1 Современные подходы к имитационному исследованию сложных систем

Выбор методов имитационного моделирования зависит также от требуемых уровней абстрагирования и детализации исследуемых процессов при разработке соответствующих моделей (рис. 2).



Рис.2 Влияние уровней абстрагирования и детализации на выбор подхода к имитационному моделированию

Под системной динамикой понимается метод моделирования, направленный на исследование причинных связей и глобальных влияний между объектами и явлениями. Для него характерен очень высокий уровень абстракции. В отношении исследования транспортных систем он может рассматриваться только при исследовании транспортных проблем в государственном или региональном масштабе. Уровень детализации здесь ограничивается усредненными оценками объемов грузопотоков или пассажиропотоков и возможностей их реализации различными видами транспорта. При таком подходе для разработки модели транспортной системы возможно, как правило, и применение альтернативного аналитического метода.

С повышением уровня детализации исследуемых транспортных систем и процессов чаще всего используется дискретно-событийный или агентный подходы. При дискретно-событийном непрерывный процесс функционирования системы аппроксимируется последовательностью мгновенных переходов, определяющих изменения её состояний. Этот подход применяется, как правило, при разработке централизованных систем, в которых поведение моделируемых объектов заранее известно и регламентируется принятыми правилами, расписаниями, нормативными документами. Ярким примером тому является имитационное моделирование перевозок железнодорожным транспортом, а также всех видов перевозок, относящихся к военной сфере.

Агентное моделирование представляет собой наиболее универсальный способ представления систем, так как может описывать как очень детализированные, так и предельно абстрактные процессы. Основным направлением применения относительно транспортных систем является, например, исследование пропускной способности транспортной инфраструктуры в условиях децентрализованного использования. Каждый объект (агент) при этом обладает собственными правилами поведения, которые в основном не задаются директивно, а формируются в процессе взаимодействия с другими агентами. Примером может служить моделирование движения независимых транспортных средств в городской среде с учетом загруженности и пропускной способности автомобильных магистралей.

В современных условиях при разработке имитационных моделей ТС все больше проявляется тенденция комплексного применения рассмотренных подходов, в том числе, и с использованием аналитического моделирования.

Выбор технологии разработки имитационной модели исследуемой транспортной системы. С развитием информационных технологий решение этого вопроса становится всё более многовариантным. В работе [6] систематизированы и представлены возможные схемы разработки имитационных моделей в современных условиях, акцентировано внимание на то, что наиболее востребованным сегодня является направление разработки имитационных моделей на основе применения инструментальных программных сред моделирования.

При рассмотрении технологии разработки имитационных моделей транспортных систем первоочередное внимание уделяется процедуре прове-

дения самого имитационного исследования. В общем виде схема проведения имитационного исследования представлена на рис. 3.

Таким образом, процесс моделирования транспортной системы можно укрупнено представить в виде выполнения четырех основных этапов:

1. Формирование задания.
2. Создание модели.
3. Исследование модели.
4. Реализация результатов.

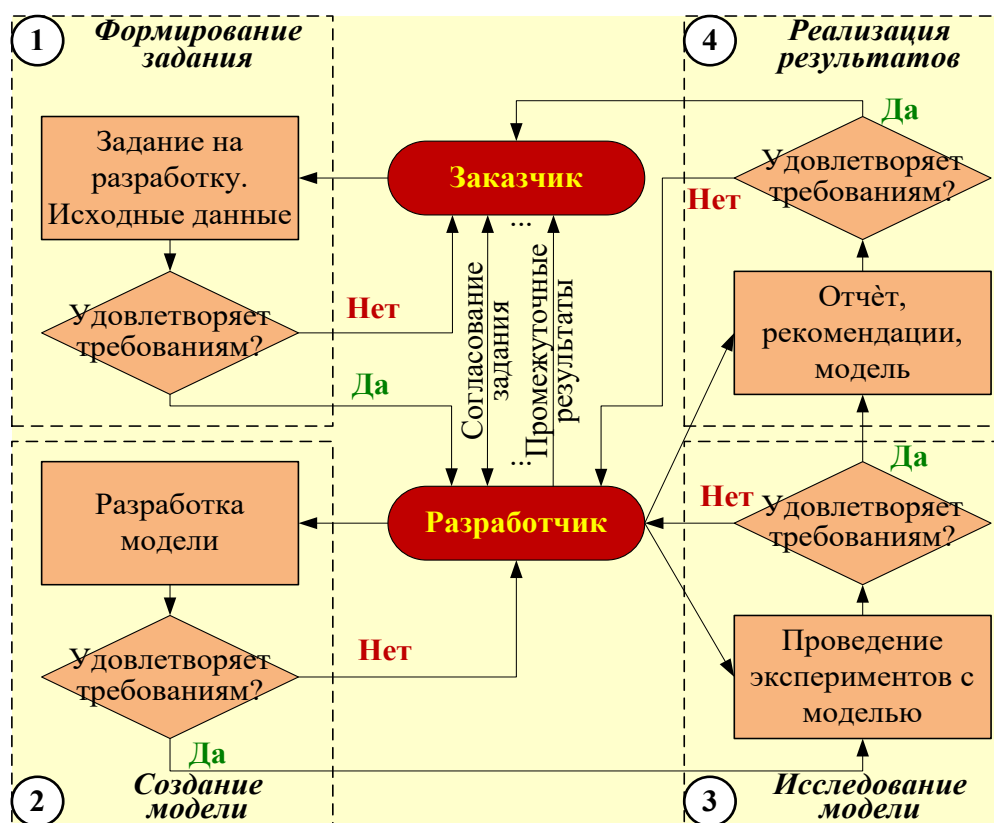


Рис.3 Схема проведения имитационного исследования

Сам процесс проведения имитационного исследования является итерационным, т.е., в зависимости от результатов проверок может потребоваться повторное прохождение уже завершённых этапов.

Формирование задания - это первый, наиболее сложно формализуемый этап, результатом которого становятся цели и задачи исследования, определяются наиболее значимые характеристики моделируемых процессов, требуемая степень детализации, определяется состав и структура транспортной системы и её имитационной модели.

На этапе создания модели осуществляется разработка логической схемы в среде имитационного моделирования, а также формируется информационное пространство системы. Этап завершается оценкой адекватности модели, что позволяет определить, насколько полученная модель отражает наиболее существенные характеристики исследуемой системы.

Содержание этапа исследования модели во многом определяется поставленными целями и задачами моделирования. После статистической обработки экспериментальных данных проводится анализ результатов. По итогам моделирования выдаются рекомендации по поводу использования разработанной модели, и оформляется техническая документация.

Выбор программной среды имитационного моделирования для создания модели и обеспечение ее адаптации к исследуемой транспортной системе. В настоящее время существующие программные среды имитационного моделирования можно классифицировать как универсальные и специализированные. Представителями первой группы являются, например, широко распространенные AnyLogic [7] и GPSS Studio [8]. Представителями второй группы могут служить среды имитационного моделирования сетей передачи данных OMNET++ [9, 10] и технических систем SimInTTech [11].

Применение программных сред имитационного моделирования позволяет существенно повысить массовость разработки имитационных моделей для исследования, в том числе, и транспортных систем. Это обусловлено, прежде всего, тем, что к разработчику снижаются требования как к специалисту в программировании, поскольку большинство моделируемых объектов и их поведение реализованы в программной среде в виде типовых процедур, модулей, блоков. Во многих случаях для настройки таких процедур достаточно лишь установить значения требуемых параметров и характеристик. Очевидно, в специализированных средах моделирования эта проблема решена наиболее полно, чем в универсальных.

Относительно транспортных систем, доступных для применения специализированных сред моделирования в настоящее время фактически нет. При этом доступность использования предполагает возможность приобретения лицензии на применение, а так же всестороннюю поддержку при использовании, включая возможность обновления программного продукта в рамках его совершенствования и модернизации. К сожалению, и выбор доступных универсальных программных сред для моделирования транспортных систем оказывается не велик, и на российском рынке ограничивается тремя программными продуктами. Среди них наиболее известные и распространенные AnyLogic и GPSS Studio, а так же относительно новая среда iWebSim. В работе [5] представлены результаты оценки возможностей рассматриваемых инструментальных средств по ряду критериев, а также востребованность применения.

Программная среда AnyLogic по большинству анализируемых критериев на данный момент безусловно превосходит альтернативные варианты, чем и обусловлено её широкое применение разработчиками имитационных моделей в различных сферах деятельности. Отличительными особенностями рассматриваемой среды являются: поддержка всех трех известных и наиболее распространенных в настоящее время рассмотренных выше подходов имитационного моделирования; возможность создавать многоподходные модели; развитость библиотеки моделирования логистических процессов.

Опыт ее применения для разработки имитационных моделей транспортных процессов, в частности, процессов оперативных перевозок войск железнодорожным транспортом, подтвердил ряд ее преимуществ. Основными преимуществами являются: возможность логического описания перевозочных процессов реализованными стандартными процедурами; оперативное изменение входных параметров; развитая визуализационная поддержка модельных экспериментов и представления выходных результатов; взаимодействие с географическими информационными системами и ряд других.

Вместе с тем, опыт разработки и применения имитационных моделей перевозочных процессов позволил выявить ряд проблемных вопросов, решение которых позволит существенно расширить возможности использования программных сред моделирования для исследования транспортных систем. Суть одного из них заключается в следующем.

В имитационных моделях ТС учитывается большое число различных параметров и характеристик перевозочных процессов и транспортной инфраструктуры, при этом для заказчика модели на панель управления выносятся лишь незначительная часть из них для обеспечения возможности варьировать значениями исходных данных при поиске рациональных вариантов организации перевозочного процесса. В ходе имитационных исследований часто возникают ситуации, требующие анализа влияния на выходные характеристики перевозочного процесса и параметров, не вынесенных на панель управления. Решение этого вопроса в настоящее время доступно только разработчику модели и в целом затрудняет ее применение.

Вынесение же, изначально, всех параметров и характеристик моделируемого процесса на панель управления, с одной стороны, усложняет анализ и поиск рационального варианта, а, с другой стороны, не решает проблему повышения доступности модели заказчику с точки зрения возможности ее адаптации для исследования однотипных перевозочных процессов на различных транспортных направлениях.

Предлагаемый подход основан на возможности загрузки в имитационную модель исходных данных через общедоступный формат файлов Excel (реализован в среде AnyLogic) и применения внешнего по отношению к среде моделирования интерфейса для настройки и управления моделью. Таким образом, имитационная модель для заказчика является полностью «черным ящиком», настраиваемым и управляемым через общедоступный интерфейс или дружественный специализированный интерфейс. Этот подход, кроме того, обеспечивает хранение исходных данных имитационных моделей в общедоступном формате. Суть предлагаемого подхода поясняется рис. 4.

Применение рассматриваемого подхода предполагает и хранение, при необходимости, выходных параметров и характеристик моделируемого перевозочного процесса в формате Excel файлов. Это обеспечивает возможность распараллеливания разработки сложных имитационных моделей путем декомпозиции моделируемого процесса на ряд взаимосвязанных подпроцессов, в том числе, с различными уровнями детализации.

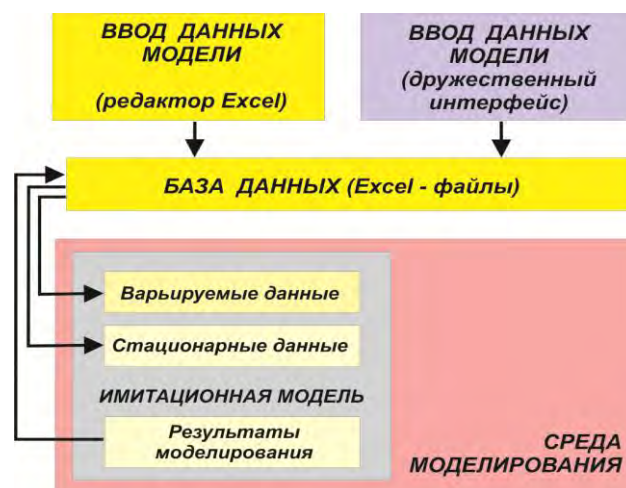


Рис.4 Схема загрузки исходных данных в имитационную модель

В результате логического описания и разработки модели какого-либо типового транспортного процесса (системы) при использовании предлагаемого подхода может быть фактически сформировано специализированное инструментальное средство имитационного моделирования. При этом применение данного средства, очевидно, ограничено исследованием именно рассматриваемых типовых процессов, но возможность «настройки» моделей через структурированный внешний интерфейс существенно упрощает их адаптацию на исходные данные различных заказчиков. Привлечение специалистов с подготовкой, соответствующей уровню разработчиков имитационных моделей, в этом случае, теряет свою актуальность.

Таким образом, имеющийся отечественный и зарубежный опыт показывает, что одним из важнейших резервов повышения эффективности функционирования транспортных систем является их всестороннее комплексное исследование, что сегодня практически невозможно без применения технологии имитационного моделирования.

Проведенный анализ основных проблемных вопросов имитационного моделирования транспортных систем позволил определить рациональные подходы к разработке их имитационных моделей в современных условиях. Предложенная схема взаимодействия имитационной модели с данными, представленными в формате Excel файлов, фактически обеспечивает реализацию концепции разработки специализированных сред имитационного моделирования типовых транспортных процессов (систем) на основе применения известных универсальных инструментальных средств разработки имитационных моделей.

Библиографический список

1. Транспортная стратегия Российской Федерации на период до 2030 года // Собр. законодательства РФ. – 29.12.2008. – № 52. – Ч. 2.
2. Башмаков А.И., Башмаков И.А. Интеллектуальные информационные технологии: учеб. пособие. – М.: Изд-во МГТУ им. Н.Э. Баумана, 2005. – 304 с.

3. Девятков В. В. Опыт системного анализа транспортных систем с использованием метода имитационного моделирования / В. В. Девятков // Вестник НЦБЖД, 2017. – № 1 (31). – С. 18-28.
4. Девятков В. В. Методология и технология имитационных исследований сложных систем: современное состояние и перспективы развития [Текст]: монография / В. В. Девятков – М.: Вузовский учебник: ИНФРА-М, 2013. – 448 с.
5. Филяев М. П. Имитационное моделирование процессов материально-технического обеспечения войск: методы и инструментальные средства / М. П. Филяев, И. В. Фесенко, А. В. Венедиктов// Научные проблемы материально-технического обеспечения Вооружённых Сил Российской Федерации / СПб: Изд-во Политехнического университета, выпуск 1(7) 2018. – С. 83 – 89.
6. Филяев М.П. Повышение эффективности процессов материально-технического обеспечения на основе применения современных инструментальных средств имитационного моделирования // В сборнике: Информационные системы и технологии: теория и практика. Сборник научных трудов. Ответственный редактор А.М. Заяц. – Том. Выпуск 10. Часть 2. 2018. С. 45-50.
7. Официальный сайт компании AnyLogicCompany – Режим доступа <http://www.anylogic.ru/> (дата обращения 01.02.2019).
8. Официальный сайт компании ООО «Элина-Компьютер» – Режим доступа <http://www.elina-computer.ru/> (дата обращения 02.02.2019).
9. Думов М.И., Хабаров С.П. Использование OMNET++ для моделирования беспроводных WI-FI сетей // В сборнике: Информационные системы и технологии: теория и практика. Сборник научных трудов. Ответственный редактор А.М. Заяц. – Том. Выпуск 10. Часть 1. 2018. С. 44-53.
10. Хабаров С.П. Моделирование Ethernet сетей в среде OMNET++ INET FRAMEWORK // Научно-технический вестник информационных технологий, механики и оптики. 2018. Т. 18. № 3. С. 462-472.
11. Официальный сайт компании ООО «3В Сервис» – Режим доступа <http://www.symintech.ru/> (дата обращения 02.02.2019).

С.П. Хабаров, кандидат технических наук, доцент
Кафедра информационных систем и технологий
СПб ГЛТУ им. С.М.Кирова
serg.habarov@mail.ru

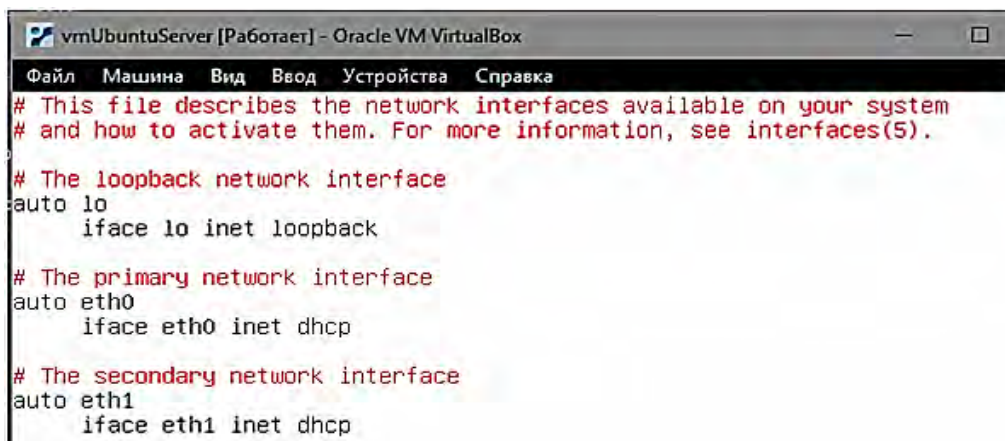
ИСПОЛЬЗОВАНИЕ УТИЛИТЫ WEBSOCKETD В СРЕДЕ ОС UBUNTU SERVER

В современных сетевых информационных системах различного назначения [1, 2] довольно часто стал применяться протокол WebSocket [3], под-

держивающий технологию двунаправленной связи между браузером и сервером. Его используют для выполнения параллельных вычислений, при построении распределенных систем и Web-приложений. Такие системы по своей природе являются гетерогенными, объединяя в себе вычислительные устройства разного класса с различными типами операционных систем (ОС) [4, 5]. При этом серверные устройства, как правило, используют Linux подобные ОС. В этих условиях представляется актуальной задача исследования возможности настройки и использования утилиты WebSocketD в среде серверной реализации ОС Ubuntu.

Для решения поставленной задачи предлагается использовать созданную в среде Oracle VM VirtualBox виртуальную машину с ОС Ubuntu Server 14.04, не имеющей графической оболочки, но с установленным на ней терминальным сервером ssh. В сетевых настройках виртуальной машины предусмотрено два сетевых адаптера: "Виртуальный адаптер хоста" и "NAT". Первый из них используется для связи базового Windows компьютера с виртуальной машиной. Второй должен обеспечить доступ гостевой ОС к ресурсам Интернет для возможности установки в нее необходимых программных средств.

При первом запуске виртуальной машины следует проверить правильность настройки ее сетевых интерфейсов для доступа, как из среды гостевой ОС к базовому компьютеру, так и к Интернету, выполнив для этого команду `ifconfig | more`. При отсутствии таких возможностей потребуется провести необходимые изменения в файле `/etc/network/interfaces` (рис. 1).



```
vmUbuntuServer [Работает] - Oracle VM VirtualBox
Файл  Машина  Вид  Ввод  Устройства  Справка
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).
# The loopback network interface
auto lo
    iface lo inet loopback
# The primary network interface
auto eth0
    iface eth0 inet dhcp
# The secondary network interface
auto eth1
    iface eth1 inet dhcp
```

Рис. 1. Исходный вид файла `/etc/network/interfaces`

После внесения необходимых изменений, следует сохранить файл и выполнить перезагрузку ОС Ubuntu Server (`sudo reboot`). При наличии нормального сетевого доступа и после того, как будет определен IP адрес виртуальной машины (например, 192.168.56.3) появляется возможность удаленного терминального доступа к ней по протоколу SSH.

Для этого на базовом компьютере с ОС Windows надо запустить утилиту PuTTY, ввести соответствующий адрес и выполнить подключение. В результате откроется окно терминала PuTTY, где после ввода логина и пароля будет выполнен вход в ОС Ubuntu Server (рис. 2).

```
serp@ubuntu-server: ~
login as: serp
serp@192.168.56.3's password:
Welcome to Ubuntu 14.04 LTS (GNU/Linux 3.13.0-24-generic i686)

* Documentation:  https://help.ubuntu.com/

System information as of Wed Oct 31 15:10:21 MSK 2018

System load:  0.4                Processes:            76
Usage of /:   13.2% of 6.99GB    Users logged in:    0
Memory usage: 3%                IP address for eth0: 192.168.56.3
Swap usage:  0%                IP address for eth1: 10.0.3.15

Graph this data and manage this system at:
https://landscape.canonical.com/

Last login: Wed Oct 31 15:10:21 2018
serp@ubuntu-server:~$
```

Рис. 2. Удаленный вход в среду ОС Ubuntu Server с основного компьютера

Из системной подсказки видно, что мы находимся в домашнем каталоге пользователя `serp` на компьютере `Ubuntu-server`. На этом этап настройки среды виртуализации и гостевой ОС следует считать законченным, и можно перейти к установке утилиты `websocketD`. Для работы с ней желательно создать новый каталог: `mkdir ws`

```
cd ws.
```

Именно в этот каталог будет загружена утилита `websocketD`, и там же будут сохраняться файлы примеров. Для загрузки утилиты следует обратиться на <https://github.com/joewalnes/websocketd/releases>, где представлены различные варианты ее реализации (рис. 3)



Рис. 3. Вид сайта с различными реализациями `websocketD`

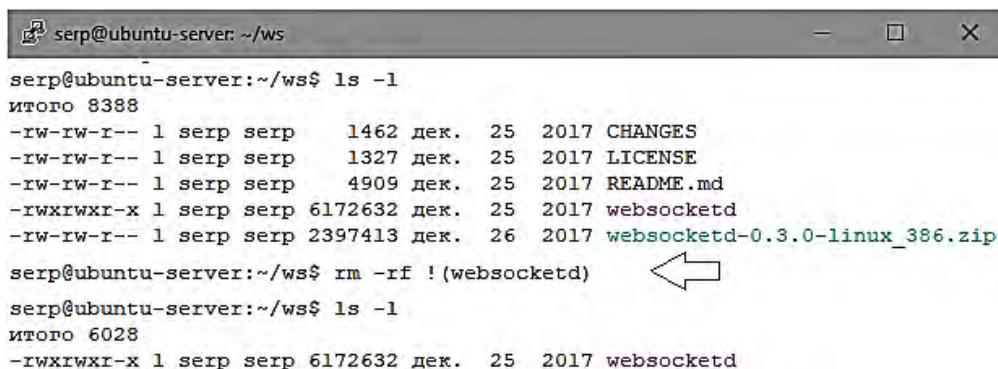
Виртуальная машина в качестве гостевой использует 32-битную версию ОС `Ubuntu Server`, которая является Linux подобной ОС. Поэтому загрузить нужную версию можно, если, находясь в каталоге `ws`, выполнить команду:

```
wget
https://github.com/joewalnes/websocketd/releases/download/v0.3.0/websocketd-0.3.0-linux_386.zip
```

В результате выполнения этой команды в каталоге `ws` появится новый архивный файл, разархивировать который можно командой:


```
unzip websocketd-0.3.0-linux_386.zip
```

При этом в рабочем каталоге появится несколько файлов, которые не нужны для дальнейшей работы, поэтому имеет смысл удалить из рабочего каталога все файлы за исключением файла `websocketd` (рис. 4).



```
serp@ubuntu-server: ~/ws
serp@ubuntu-server:~/ws$ ls -l
итого 8388
-rw-rw-r-- 1 serp serp 1462 дек. 25 2017 CHANGES
-rw-rw-r-- 1 serp serp 1327 дек. 25 2017 LICENSE
-rw-rw-r-- 1 serp serp 4909 дек. 25 2017 README.md
-rwxrwxr-x 1 serp serp 6172632 дек. 25 2017 websocketd
-rw-rw-r-- 1 serp serp 2397413 дек. 26 2017 websocketd-0.3.0-linux_386.zip
serp@ubuntu-server:~/ws$ rm -rf !(websocketd)
serp@ubuntu-server:~/ws$ ls -l
итого 6028
-rwxrwxr-x 1 serp serp 6172632 дек. 25 2017 websocketd
```

Рис. 4. Удаление файлов командой `rm -rf !(websocketd)`

Как видно из рис. 4 файл `websocketd` является исполняемым и присутствует в рабочем каталоге. Для проверки того, что он работоспособен можно выполнить, например, команду просмотра версии утилиты `websocketD`:

```
serp@ubuntu-server:~/ws$ ./websocketd --version
websocketd 0.3.0 (gol.9.2 linux-386) --
```

На этом подготовительный этап можно закончить и перейти к тестированию работы утилиты `websocketD` в среде ОС Ubuntu, начав с простого примера, когда клиентом является стандартный браузер с поддержкой WebSocket API. Он будет запущен на основном компьютере. В него будет подгружаться html страница для связи с WebSocket сервером, который реализуется на виртуальной машине с ОС Ubuntu. Так как сервер работает в среде Linux подобной ОС, то в качестве языка кода серверной обработки выбирается Bash, который входит в стандартную поставку всех Linux подобных систем.

Пусть для начала файл кода для серверной обработки представляет собой однострочный эхо-сервер, имеет имя `EchoLine` и состоит всего из трех строк:

```
serp@ubuntu-server:~/ws$ cat > EchoLine
#!/bin/bash
read line
echo $line
```

Для окончания ввода файла используется `Ctrl+D`, что вызывает сохранение файла в рабочем каталоге. Проверить его работоспособность можно путем ввода команды `bash EchoLine`, результат выполнения которой будет:

```
serp@ubuntu-server:~/ws$ bash EchoLine
Hello, world!
Hello, world!
```

Чтобы использовать этот файл, как код серверной обработки, он должен быть исполняемым. Для этого надо выполнить команду:

```

serp@ubuntu-server:~/ws$ chmod 744 EchoLine
serp@ubuntu-server:~/ws$ ./EchoLine
Hello, world!
Hello, world!

```

После этого появляется возможность использовать файл EchoLine в качестве кода серверной обработки, связав его входные и выходные потоки с утилитой websocketD. Предполагая, что на клиентской стороне пока отсутствует какая-либо html страница с JavaScript обработкой WebSocket API, запустим на серверной стороне утилиту websocketd с поддержкой отладочного режима:

```

./websocketd --port=1234 -devconsole ./EchoLine
Sat, 03 Nov 2018 14:14:36 +0400 | INFO | server | |
Serving using application : ./EchoLine
Sat, 03 Nov 2018 14:14:36 +0400 | INFO | server | |
Starting WebSocket server : ws://ubuntu-server:1234/
Sat, 03 Nov 2018 14:14:36 +0400 | INFO | server | | De-
veloper console enabled : http://ubuntu-server:1234/

```

Это позволит проверить работу кода серверной обработки в ОС Ubuntu, подключаясь к ней из браузера основного Windows компьютера (рис. 5).

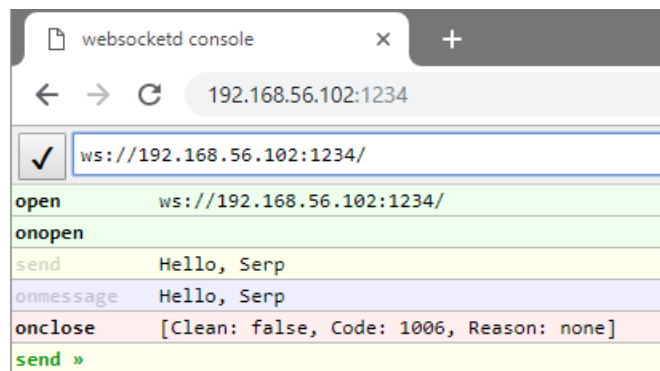


Рис. 5. Доступ к WebSocket серверу со стороны клиента

Следует отметить, что в этом примере при каждой передаче данных осуществляется новое подключение клиента к серверу, установление нового соединения (open), обмен данными (onopen) и разрыв соединения (onclose). Поэтому для новой передачи потребуется повторно нажимать на кнопку "Connect" с пиктограммой в виде птички. Следует отметить, что даже после разрыва соединения сервер продолжает свою работу, прослушивает порт 1234 и готов к соединению. Прервать работу сервера можно нажатием Ctrl+C в терминале виртуальной машины.

Чуть усложним рассматриваемый пример и разработаем эхо-сервер, который будет представлять собой бесконечный цикл по перенаправлению пользовательских потоков. Для его реализации также воспользуемся скриптовым языком BASH и на основе файла EchoLine создадим новый файл с именем EchoServer:

```
#!/bin/bash
do
    read line
    echo $line
done
```

После создания этого файла необходимо определить его как исполняемый и проверить работоспособность, путем запуска в терминале на выполнение:

```
serp@ubuntu-server:~/ws$ ./EchoServer
Line number 1
Line number 1
Line 2
Line 2
^C
```

Если проверка завершилась удачно, то это позволяет использовать его в качестве кода серверной обработки, и запустить на виртуальной машине WebSocket сервер, используя команду:

```
./websocketd --port=1234 --devconsole ./EchoServer.
```

Для проверки работоспособности WebSocket сервера имеет смысл на основном компьютере открыть браузер и выполнить подключение по HTTP к соответствующему порту виртуальной машины (рис. 6)

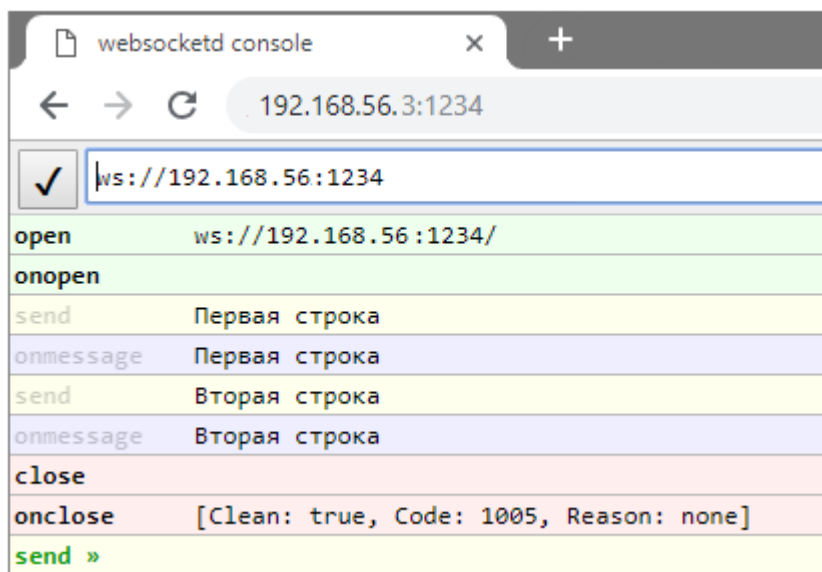


Рис. 6. Подключение с Windows узла к эхо-серверу Ubuntu машины

Если принять во внимание, что ранее была разработана и уже существует html страница с поддержкой WebSocket API [6], которая позволяет изменять адрес и порт WebSocket сервера, к которому предполагается выполнять подключение, то после выполнения команды:

```
./websocketd --port=1234 ./EchoServer ,
```


появляется возможность подключения с основного Windows компьютера к виртуальной машине с ОС Ubuntu Server, используя для этого браузер, в который подгружается эта html страница (рис. 7).

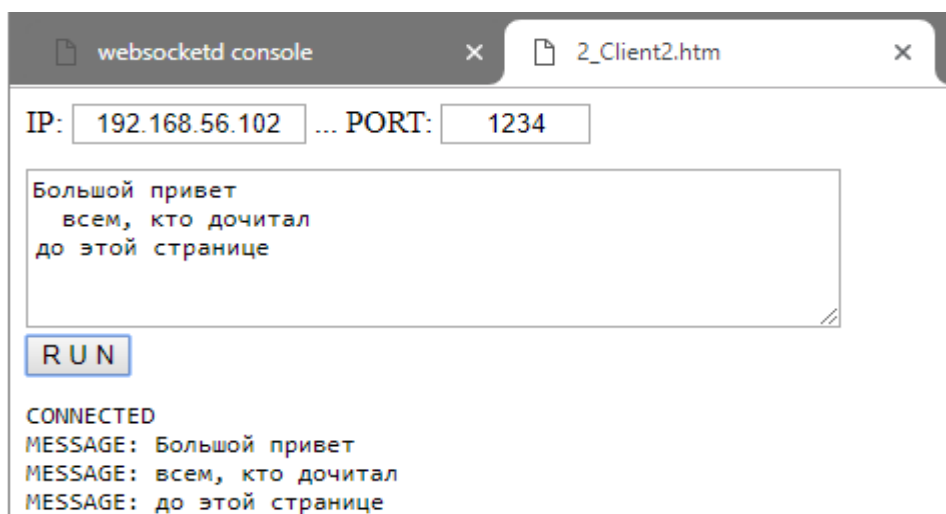


Рис. 7. Подключение с Windows узла к эхо-серверу Ubuntu машины

После рассмотрения этих простейших примеров создания WebSocket серверов в виртуальной машине с ОС Ubuntu Server, в которой отсутствует графическая оболочка, и в которую можно удаленно войти через PuTTY, имеет смысл сделать несколько важных замечаний.



Замечание.

1. Используя для удаленного доступа утилиту PuTTY, запуская виртуальную машину с ОС Ubuntu Server, и после ее загрузки не обязательно входить в среду ОС, а можно сразу запускать PuTTY и из ее консоли входить в ОС Ubuntu Server.

2. Так как виртуальная машина использует "Виртуальный адаптер хоста" с включенным DHCP сервером, то в соответствии с файлом `/ect/network/interfaces` гостевая ОС может получать разные адреса, что необходимо учитывать при выполнении примеров.

3. Если последнее замечание принципиально, то можно внести изменения в файл сетевой конфигурации:

```
iface eth0 inet static
address 192.168.56.10
netmask 255.255.255.0
```

назначив виртуальной машине уникальный статический адрес.

Отдельный интерес представляет использование утилиты `websocketD` совместно с активными серверными кодами, которые начинают генерировать выходной поток клиенту сразу же после установления его связи с сервером. Этот процесс базируется на том, что сама попытка со стороны клиента установить связь с WebSocket сервером уже является запросом о том, что клиенту что-то надо от сервера.

В качестве примера генерации сервером последовательного выходного потока рассмотрим запуск на сервере счетчика, который генерирует числа от

1 до 10 с заданным временным интервалом. В составе Linux подобных ОС присутствует, как правило, несколько встроенных скриптовых языков, таких как Bash, Python и Perl. Для иллюстрации возможности работы websocketD с кодами серверной обработки на любом из этих языков программирования реализуем один и тот же алгоритм активного серверного кода на каждом из отмеченных выше языков. С этой целью в каталоге ws создадим новый каталог с именем sample, а в нем четыре файла: counter.pl, counter.sh, counter.py и counter.php, используя для этого примеры с сайта <http://websocketd.com>. Так, например, файл counter.py будет иметь вид:

```
#!/usr/bin/python
from sys import stdout
from time import sleep
for count in range(0, 10):
    stdout.flush()
    sleep(1.0)
done
```

Для проверки работоспособности вновь созданных файлов имеет смысл выполнить следующую последовательность команд:

```
perl counter.pl
bash counter.sh
python counter.py
php counter.php
```

При этом возможна ситуация, что в ОС используемой виртуальной машины не установлен интерпретатор PHP. Его можно установить, выполнив:

```
sudo apt-get install php5-cli
```

Для того чтобы утилита websocketD могла использовать программные файлы в качестве кодов серверной обработки необходимо, чтобы они были исполняемыми. Этого можно добиться, выполнив, например, следующий набор команд:

```
serp@ubuntu-server:~/ws/samples$ chmod +x counter.sh
serp@ubuntu-server:~/ws/samples$ chmod +x counter.py
serp@ubuntu-server:~/ws/samples$ chmod +x counter.pl
serp@ubuntu-server:~/ws/samples$ chmod +x counter.php
```

и убедиться в правильности произведенных действий путем просмотра содержимого текущего каталога:

```
serp@ubuntu-server:~/ws/samples$ ls -l
```

После этого появляется возможность активации WebSocket сервера. Для этого, находясь в каталоге sample, достаточно выполнить команду:

```
../websocketd --port=8081 --devconsole ./counter.sh
```

Для проверки работы сервера надо на основном Windows компьютере открыть браузер и ввести адрес <http://192.168.56.10:8081> (рис. 8).

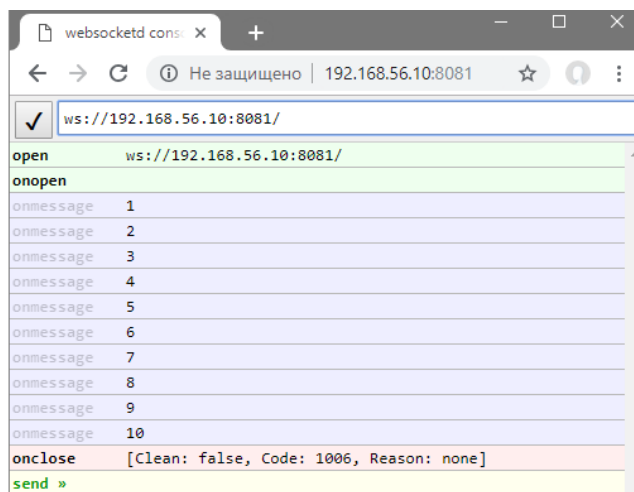


Рис. 8. Результат подключения к WebSocket серверу

Возможны и другие варианты тестирования WebSocket сервера, создаваемого на виртуальной машине с ОС Ubuntu. В частности, сервер может быть запущен командой

```
../websocketd --port=8081 --devconsole --dir=.
```

При этом доступ к нему из браузера может выполняться путем указания разных адресных строк

```
http://192.168.56.10:8081/counter.sh
http://192.168.56.10:8081/counter.py
http://192.168.56.10:8081/counter.pl
http://192.168.56.10:8081/counter.php
```

в зависимости от желания запустить тот или иной код серверной обработки. Возможен и иной подход, при котором выбор нужного кода серверной обработки осуществлялся бы из html страницы, загружаемой в браузер клиента с сервера. Для этого можно создать достаточно простой файл counter.html:

```
<!DOCTYPE html>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<title>Test Counter</title>
Выберите нужный Вам серверный код
<select id="cod">
  <option value="sh">Bash</option> <option value="py">Python</option>
  <option value="pl">Perl</option> <option value="php">PHP</option>
</select> . . . .
<input id="btn" type="button" value="Выполнить"/>
<div id="Nn" style="text-align:center; font:bold 150px arial"></div>

<script>
btn.addEventListener('click', function(e) {
  var file = 'counter.' + document.getElementById('cod').value,
```

```

ws = new WebSocket('ws://192.168.56.10:8081/' + file);
ws.onopen = function() {
    document.body.style.backgroundColor = '#cfc';
};
ws.onclose = function() {
    document.body.style.backgroundColor = null;
};
ws.onmessage = function(e) {
    document.getElementById('Nn').textContent = e.data;
};
});
</script>

```

Этот файл следует разместить на сервере в каталоге `samples`, а затем инициировать работу WebSocket сервера, выполнив команду:

```

samples$ ../websocketd --port=8081 --dir=./ --
staticdir=./

```

После этого на клиентской стороне, открыв браузер, следует указать адрес соответствующей html страницы, выбрать язык нужного серверного кода и нажать кнопку "Выполнить" (рис. 9).

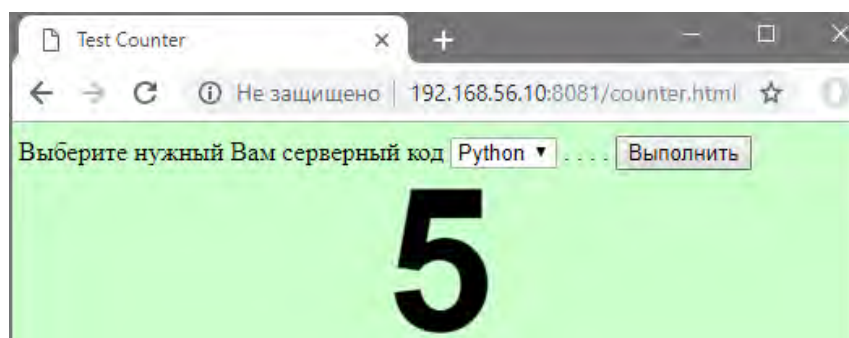


Рис. 9. Загруженная с сервера html страница в процессе выполнения серверного кода с передачей результатов клиенту по WebSocket протоколу

Следует отметить, что все серверные коды реализуют один и тот же алгоритм, поэтому результаты, получаемые на данной странице, будут идентичны между собой. В реальных информационных системах серверные коды в меньшей степени будут отличаться языками, а в большей степени алгоритмами, которые они реализуют.

Отдельно следует рассмотреть пример, который иллюстрирует возможность реализации на одном компьютере сразу нескольких одновременно работающих серверов, обменивающихся по WebSocket протоколу входными и выходными потоками с одним или несколькими клиентами. Для реализации этого простого примера необходимо:

- Запустить в работу виртуальную машину с ОС Ubuntu Server.
- После ее загрузки, не входя в ядро ОС, запустить утилиту PuTTY и выполнить удаленное подключение к этой машине.
- Введя логин и пароль, войти в консоль ОС и перейти в каталог, где находятся коды серверной обработки (`~/ws/samples$`).

- Активировать работу первого WebSocket сервера, используя для этого, например, 8081 порт и серверный код на языке Python.

```
samples$ ../websocketd --port=8081 --devconsole ./counter.py
```

- После этого следует запустить вторую копию утилиты PuTTY и выполнить подключение к уже работающей виртуальной машине.
- Пройти начальную аутентификацию и перейти в каталог примеров кодов серверной обработки.
- Находясь во второй копии утилиты PuTTY, активировать работу еще одного WebSocket сервера на этой же виртуальной машине, выполнив для этого команду, которая будет использовать другие порт и код серверной обработки, например, на языке Bash:

```
samples$ ../websocketd --port=8082 --devconsole ./counter.sh
```

- Для тестирования работы WebSocket серверов следует открыть две вкладки или две копии браузера на основном Windows компьютере, и выполнить подключение по одному и тому же IP адресу, но к разным портам (рис. 10).

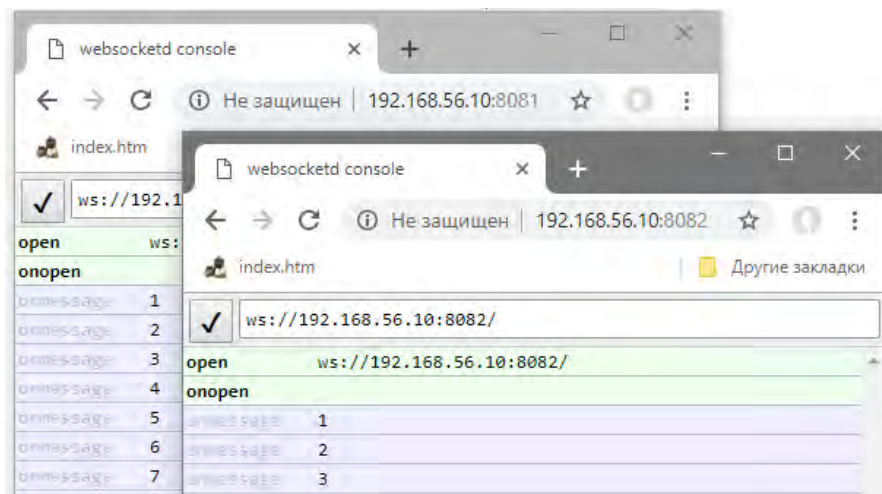


Рис. 10. Доступ из браузера к разным серверам с разными серверными кодами

Следует отметить, что для ряда практических реализаций более важной является возможность построения таких html страниц, которые позволяют на одной странице открыть несколько WebSocket соединений с разными серверами, то есть создать распределенную систему.

Так, например, в биллинговых системах различные сервера или разные серверные коды могут отвечать за разные операции: операцию пополнения счета, операцию оплаты картой покупки (снятия средств со счета) и операцию получения остатка средств на карте.

В технологических системах и системах умного дома одни сервера или серверные коды отвечают за управление исполнительными устройствами, а другие за сбор информации с датчиков. При этом вся эта информация будет сосредотачиваться в одном приложении, обеспечивая как мониторинг, так и управление системой.

Библиографический список

1. Хабаров С.П., Заяц А.М. Использование технологии WebSocket в клиент-серверных экспертных системах. // Леса России: политика, промышленность, наука, образование / материалы научно-технической конференции. Том 1 – СПб.:СПбГЛТУ, 2017. – с.278-280
2. Заяц А.М., Хабаров С.П. Организация доступа к беспроводным AD НОС сетям информационных систем мониторинга лесных территорий из среды Windows 10. // Известия Санкт-Петербургской лесотехнической академии. – СПб, 2018, Вып. 223. - с. 285-299.
3. Хабаров С.П. Взаимодействие узлов сети по протоколу WebSocket. // Сборник научных трудов "Информационные системы и технологии: теория и практика" – СПб.:СПбГЛТУ, 2017. № 9. – с.109-119
4. С.П.Хабаров. Организация гетерогенных ЛВС с терминальным доступом между ее узлами. // Известия СПбЛТА-СПб., 2016, № 216, с. 267-280
5. С.П.Хабаров. Терминальный доступ между UBUNTU и WINDOWS компьютерами. //Информационные системы и технологии: теория и практика: сб. научн. тр. - СПб.:СПбГЛТУ, 2016, № 8, с. 30-43.
6. Хабаров С.П. Использование утилиты websocketd для удаленного выполнения программ.//Сборник научных трудов "Информационные системы и технологии: теория и практика" — СПб.:СПбГЛТУ, 2017. № 9. — с.94-108.

С.П. Хабаров, кандидат технических наук, доцент
Кафедра информационных систем и технологий
СПб ГЛТУ им. С.М.Кирова
serg.habarov@mail.ru

ИСПОЛЬЗОВАНИЕ SIMINTECH ДЛЯ АНАЛИЗА СИСТЕМ АВТОМАТИЧЕСКОГО УПРАВЛЕНИЯ

Одним из широко используемых направлений при построении и анализе динамических систем управления является имитационное моделирование, для проведения которого в последнее время наиболее часто используется пакет SimInTech [1,2]. Он поставляется фирмой «ЗВ Сервис» и предназначен для исследования нестационарных процессов в системах автоматического управления [3,4], а также в следящих приводах и роботах, фактически в любых технических системах [5,6,7], описание динамики которых может быть представлено в виде системы дифференциально-алгебраических уравнений и/или реализовано методами структурного моделирования.

Основным направлением использования SimInTech является создание моделей объектов управления и проектирование алгоритмов управления, с возможностью их отладки с учетом имеющихся моделей объектов, а также генерация исходного кода на языке Си для программируемых контроллеров.

Постановка задачи. В статье описывается технология использования пакета SimInTech для проектирования и анализа простейшей системы управления. Для этого разрабатывается модель объекта управления, алгоритм управления этим объектом и проводится анализ моделирование их совместной работы. С целью иллюстрации реализованного в SimInTech структурно-иерархического подхода к построению моделей предполагается алгоритм управления представить в виде отдельного функционального блока, для которого автоматически средствами SimInTech будет сгенерирована dll-библиотека на языке Си. Достоверность ее работы будет проверена путем проведения в той же среде SimInTech процесса моделирования работы модели объекта совместно с этой dll-библиотекой, реализующей алгоритм управления этим объектом.

В качестве объекта управления рассматривается простейший возвратный клапан, при подаче на который сигнала управления он начинает открываться с некоторой заданной из его технических характеристик скоростью. При снятии сигнала управления он с той же скоростью возвращается в закрытое состояние. При этом предполагается, что в клапан встроены дискретные датчики его положения в двух конечных состояниях. Алгоритм управления рассматривается в его простейшей реализации, которая заключается в том, что он принимает сигнал от кнопки «Открыть/Закрыть» и передает его на исполнительный орган клапана. Кроме этого в алгоритме управления должно быть предусмотрено формирование сигналов о процессе открытия или закрытия клапана, а также сигналы неисправностей, возникающие при невыполнении этих процессов в течении заданного промежутка времени.

Создание нового проекта. Решение поставленной задачи предполагает создание нового проекта в среде схемы автоматики SimInTech, для чего в главной панели инструментов необходимо нажать кнопку «Новый проект», в выпадающем меню выбрать пункт «Схема автоматики». При этом откроется новое окно, в котором может быть создана структурная схема модели системы автоматики. Следует отметить, что при создании нового или редактировании уже существующего проекта среда SimInTech должна быть переведена в режим разработчика («Вид → Режим разработчика»).

После создания нового проекта можно приступить к построению общей модели системы управления клапаном, выделив на ней в качестве отдельного блока модель исполнительного органа (клапана). Для этого на вновь созданную схему следует поместить блок "Субструктуры/Субмодель". Двойной клик по этому блоку откроет новое окно, где надо создать модель клапана, которая должна описывать не только динамику перемещения клапана, но и работу датчиков конечных положений с возможностью имитации их отказов (рис. 1).

В модели использованы блоки "Порт входа" и "Порт выхода" (из состава вкладки "Субструктуры"), блок "Константа" (вкладка "Источники"), блоки "Сумматор" и "Множитель" (вкладка "Операторы") и блок "Кнопка" (вкладка "Ключи"). Динамику перемещения клапана моделирует блок "Интегратор с

ограничениями" (вкладка "Динамические"). Настройка его свойств должна соответствовать параметрам работы конкретного типа клапана.

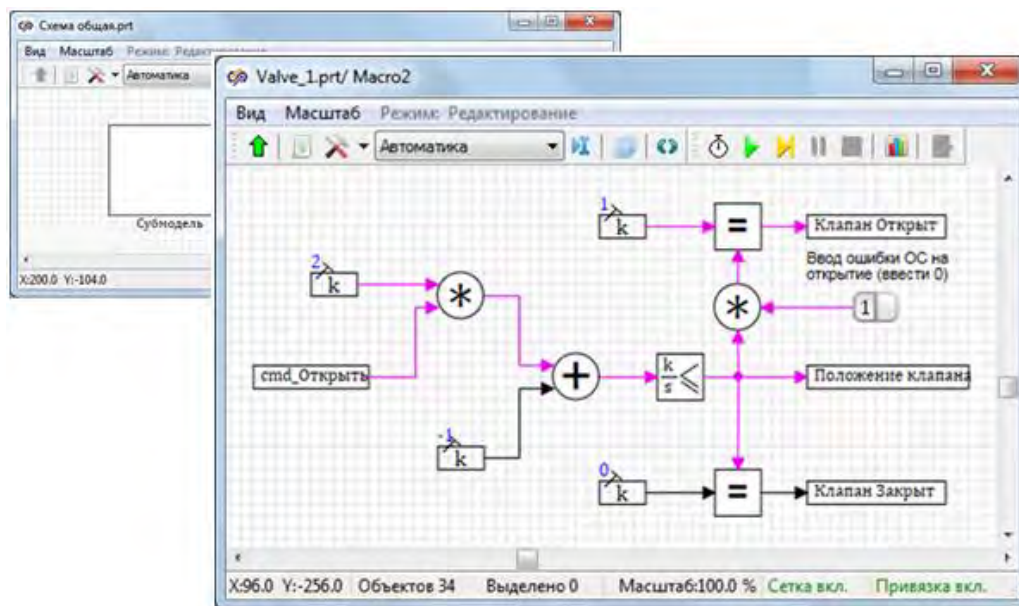


Рис. 1. Схема системы и модель работы возвратного клапана

Свойства: LimitIntegrator4			
Свойства	Общие	Порты	Визуальные слои
Название	Имя	Значение	
Коэффициенты усиления	k	0.2	
Минимальное значение	Ymin	[0]	
Максимальное значение	Ymax	[1]	
Начальные условия	x0	[0]	

Рис. 2. Определение параметров работы конкретного клапана

Приведенные на рис. 2 параметры определяют клапан, закрытое состояние которого характеризует значение равно 0 (Y_{min}), полностью открытое значение равно 1 (Y_{max}), а скорость перемещения определяет $k=0.2$, что соответствует полному перемещению клапана за 5 сек. Закончив построение модели работы клапана, надо вернуться в основную модель, нажав в панели инструментов окна субмодели кнопку с зеленой стрелкой вверх. Теперь необходимо разработать алгоритм управления этим клапаном (рис. 3).

Данная реализация предполагает, что сигнал от кнопки "Открыть/Закрыть" передается не только на субмодель клапана, но определяет и работу алгоритма управления, который от этой субмодели также получает сигналы о конечных состояниях. На их основе алгоритм управления должен формировать сигналы о том, что клапан находится в процессе открытия или закрытия, а также сигналы неисправностей при невыполнении этих процессов за заданный промежуток времени. В данном примере он выбран равным 7 сек, что больше времени полного открытия клапана.

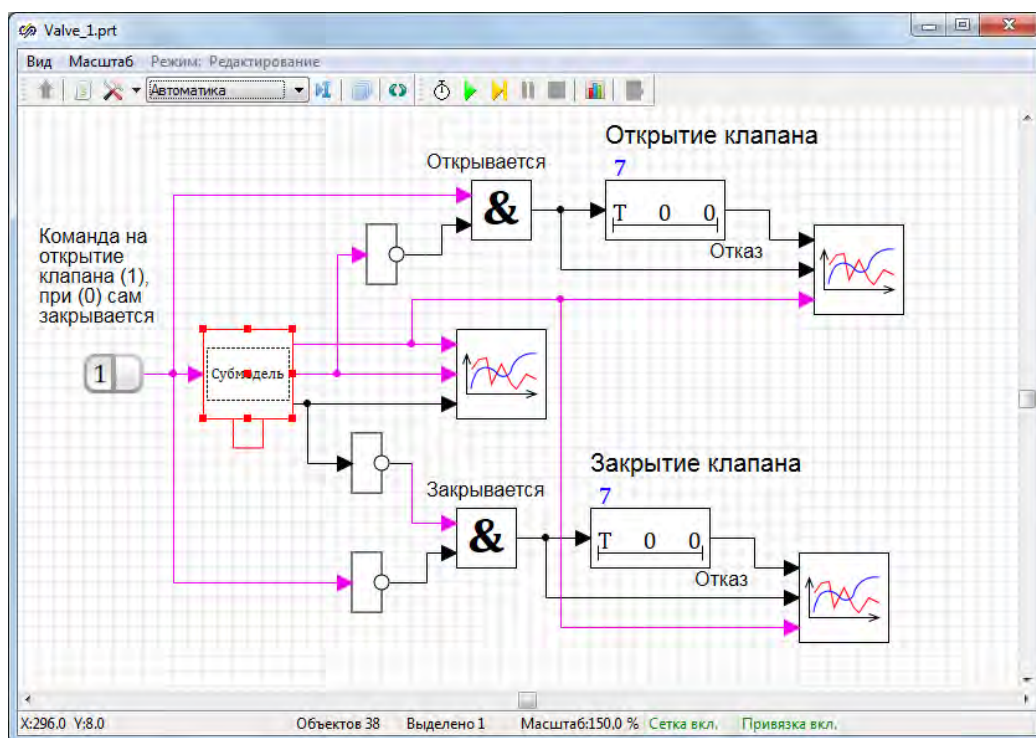


Рис. 3. Модель функционирования системы управления клапаном

На схеме алгоритма управления использованы такие типовые компоненты и блоки SimInTech, как: "Кнопка" (вкладка "Ключи"), "Оператор И" и "Оператор НЕ" ("Логические"), блок "Задержка по включению" ("Задержки и импульсы"), блок "Временной график" ("Данные"). Чтобы блок "Временной график" имел 3 входа надо правой клавишей мыши щелкнуть по этому блоку, в контекстном меню выбрать "Свойства объекта", на вкладке "Свойства" в графе "Количество входных портов" в поле "Значение" ввести 3 и нажать "Применить" и "Ок".

Для блока "Задержка по включению" также надо определить параметры его работы. Для этого, открыв окно свойств этого компонента, следует в графе "Массив времён подтверждения, сек" в поле "Значение" ввести число 7, то есть семь секунд будет задержка на изменение выходного сигнала этого блока с 0 на 1 при поступлении на его вход единичного сигнала.

Результаты моделирования системы приведены на рис. 4, где представлены как сигналы модели исполнительного органа (текущее положение клапана), так и сигналы от системы управления (работа клапана и его неисправность). Из этого графика видно, что на нулевой секунде поступила команда на открытие клапана, что вызвало его перемещение и срабатывание системы управления по формированию сигнала "Работа клапана на открытие". В районе пятой секунды клапан достиг своей конечной точки. Сработал конечный выключатель, что привело к снятию системой управления сигнала "Работа клапана на открытие".

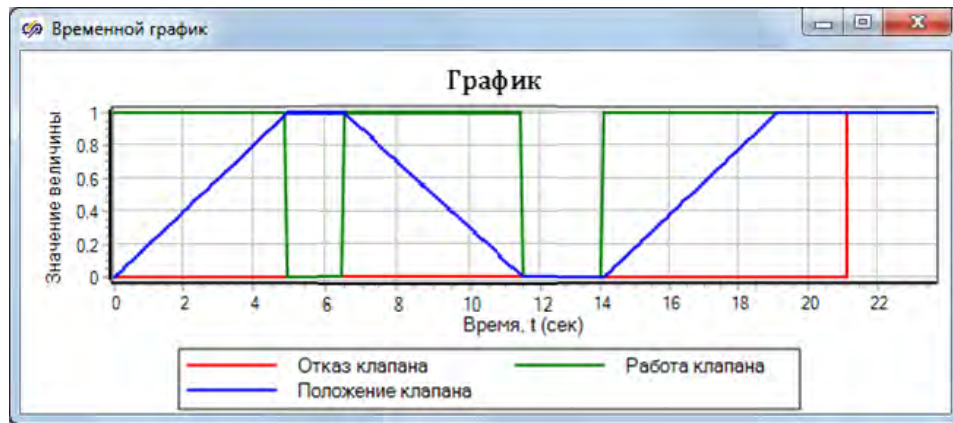


Рис. 4. Результаты моделирования системы управления клапаном

В районе шестой секунды была снята команда на открытие клапана, что вызвало его перемещение в закрытое состояние и, как следствие, формирование системой управления сигнала «Работа клапана на закрытие». Через пять секунд клапан закрылся, сработал его конечный выключатель, и система управления сняла сигнал «Работа клапана на закрытие». В районе 14-й секунды вновь поступила команда на открытие клапана. Клапан начал открываться, но в процессе его открытия был смоделирован отказ в его работе путем перевода кнопки «Ввод ошибки ОС на открытие» на субмодели клапана из состояния 1 в состояние 0). Это приводит к тому, что в систему управления в течение семи секунд от момента появления команды на открытие клапана не поступит сигнал от его конечного выключателя. На это алгоритм управления будет реагировать путем выдачи сигнала неисправности работы клапана.

Убедившись в работоспособности созданной модели, но рассматривая при этом возможные технологии использования SimInTech для анализа систем управления, поставим новую задачу по преобразованию модели таким образом, чтобы алгоритм управления представлял собой отдельный функциональный блок. Для этого сохраним разработанную модель в файле Valve_1.prt, а ее копию — в файле Valve_2.prt.

Модификация модели системы управления клапаном. С этой целью следует открыть файл Valve_2.prt и в основную схему проекта добавить еще одну субмодель, куда надо перенести схему алгоритма управления клапаном. При этом модель клапана будет составлять основу общей модели, но с некоторыми незначительными изменениями (рис. 5).

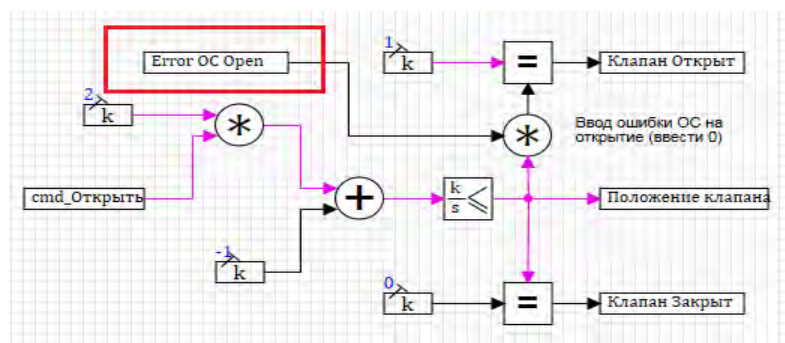


Рис. 5. Отказ датчика стал внешним параметром

Тогда общая модель системы управления клапаном будет иметь вид, аналогичный тому, что представлен на рис. 6.

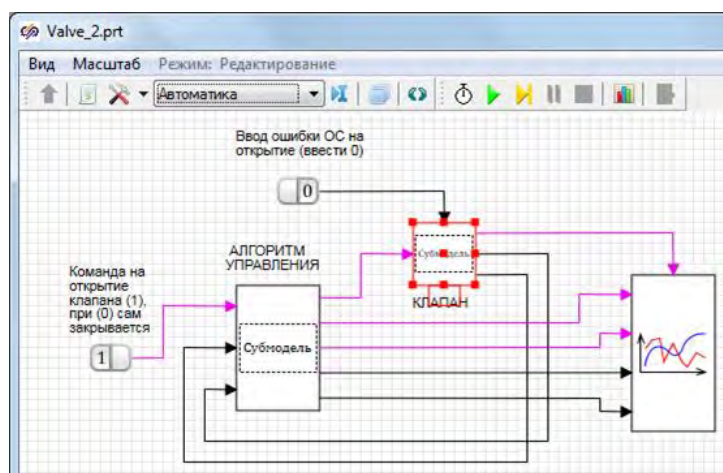


Рис. 6. Модифицированная модель системы управления клапаном

При этом модель алгоритма управления, которая перенесена в отдельную субмодель, также претерпит некоторые изменения и будет иметь вид как на рис. 7. Теперь у нее три входных и пять выходных параметров.

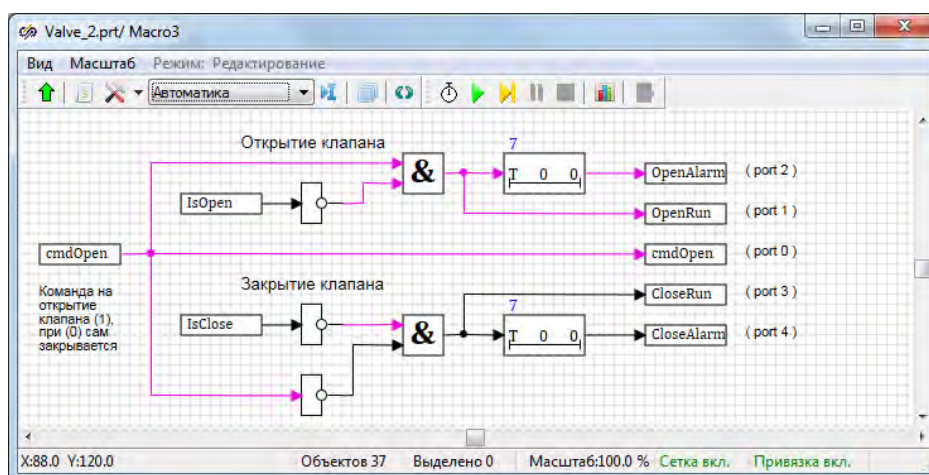


Рис. 7. Модель алгоритма управления клапаном

Выполнив соединение между собой двух субмоделей и их подключение к управляющим кнопкам и временным графикам, получаем законченную модель системы управления клапаном, которую можно запустить на выполнение и получить результаты, аналогичные полученным ранее. Это подтверждает работоспособность модуля алгоритма управления клапаном, входящим в состав системы. А если это так, то можно поставить задачу генерации для этого модуля кода на языке Си, а также внешней, доступной для подключения, dll-библиотеки.

Генерация Си-кода и внешней dll-библиотеки для алгоритма управления клапаном. Для начала в исходном проекте надо активировать окно субмодели алгоритма управления и сохранить алгоритм управления в виде файла нового проекта, например, Valve_control_1.prt. Затем, открыв этот файл, надо добавить в его схему ряд надписей-комментариев (рис. 8), которые будут полезны далее. Таким образом, модель управления клапаном будет иметь

3 порта входа и 5 портов выхода, кроме этого будет зафиксирована связь каждого порта с соответствующей переменной. После чего нужно сохранить все добавления в файле, например, Valve_control_2.prt.

На следующем этапе необходимо выполнить подготовку проекта к кодогенерации. Для этого надо открыть проект Valve_control_2.prt и выполнить в нем ряд преобразований:

- Заменить используемые в схеме блоки «Порт входа/Субструктуры» на блоки «Входной контакт S3/Данные», а «Порт выхода» — на блоки «Выходной контакт S3»;
- В свойствах «Имя контакта» новых входных блоков нужно указать ключевые слова «input:0» для первого входа, «input:1» для второго входа и т.д.;
- В свойствах «Имя контакта» новых выходных блоков нужно указать ключевые слова «out:0», «out:1» и т.д.;
- В свойствах «Тип контакта» для всех блоков дополнительно надо выбрать тип переменной, соответствующий контакту данного блока (рис. 8).

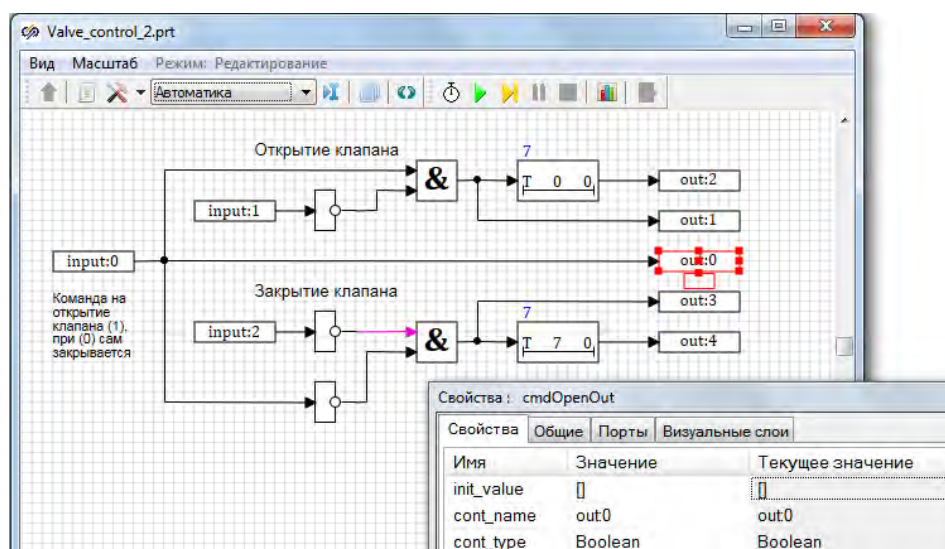


Рис. 8. Определение параметров входных и выходных контактов S3

Далее надо определить название будущей dll-библиотеки. Для этого следует открыть окно параметров проекта, используя кнопку в левой части панели инструментов и задать имя алгоритма. Пусть оно будет, например, «cmdValve» (рис. 9).

На последнем этапе необходимо сохранить текущий проект и в главном меню SimInTech выбрать «Инструменты» → «Сгенерировать программу» (рис. 10).

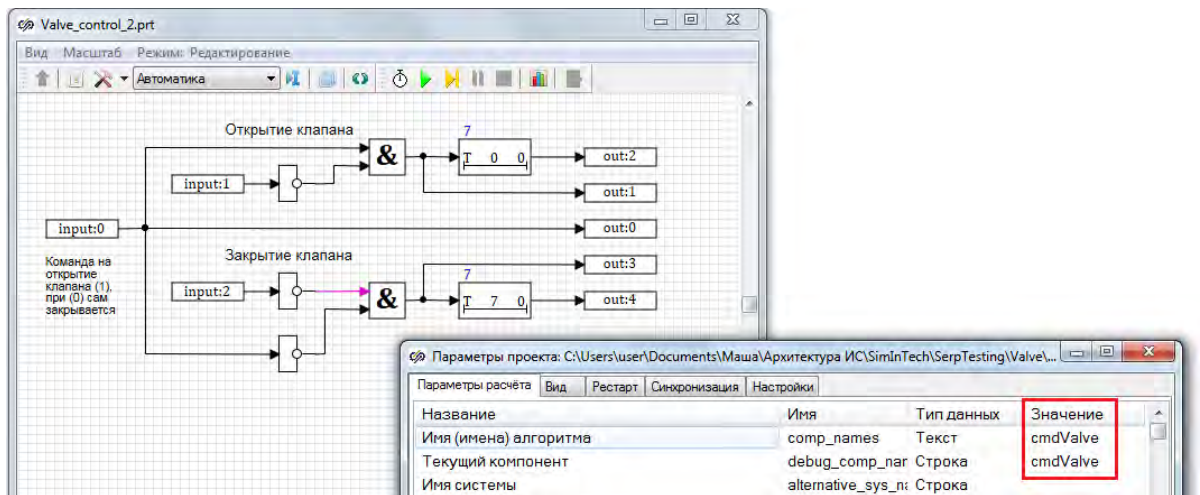


Рис. 9. Определение имени dll-библиотеки

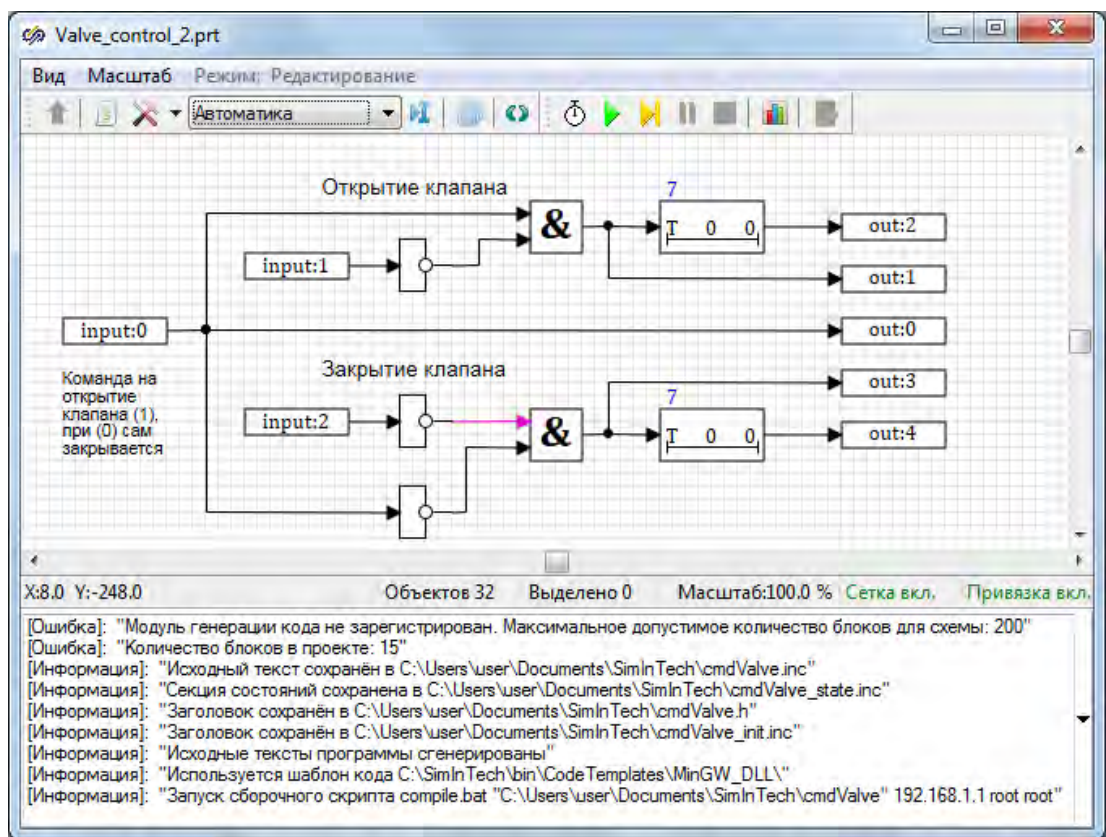


Рис. 10. Процесс генерации cmdValve.dll

Исследование работы клапана с алгоритмом управления из подключаемой внешней dll-библиотеки. Для этого надо скопировать файл проекта Valve_2.prt в файл Valve_3.prt. Открыв его, следует заменить в нем submodule алгоритма управления на только что сформированную внешнюю библиотеку. Для этого необходимо выполнить следующие действия:

- Из ранее разработанной модели удалить submodule, описывающую алгоритм управления (рис. 11);

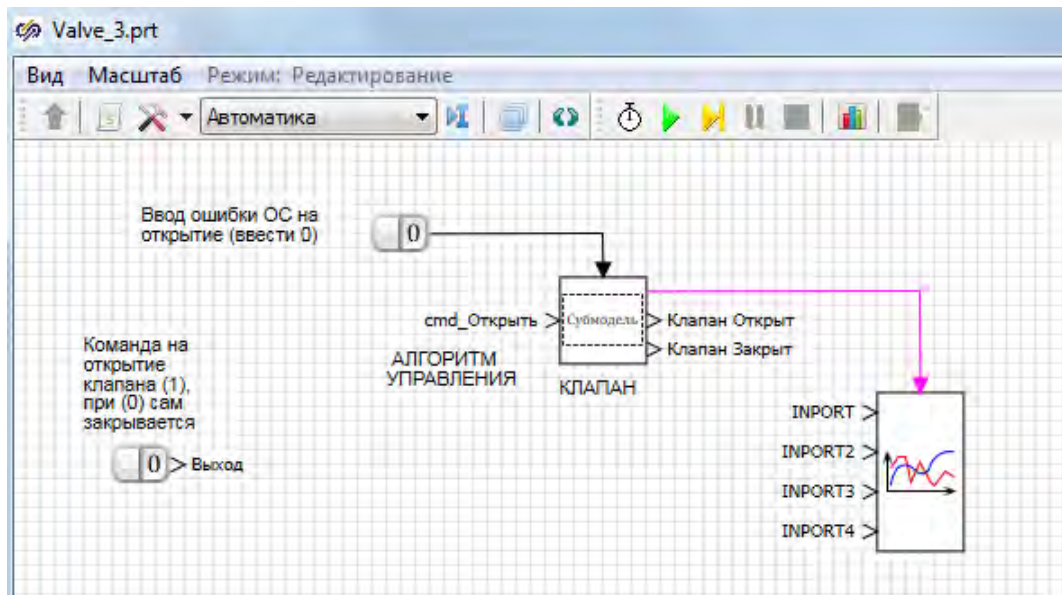


Рис. 11. Исходная модель проекта Valve_3.prt

- На ее месте следует вставить блок «Внешняя dll» из вкладки «Субструктуры». Далее надо настроить параметры этого блока в соответствии с параметрами, которые были определены при создании библиотеки (рис. 12).

Свойства	Имя	Значение	Текущее значение
Количество портов	port	3	0
Массив размерностей выходов	outdims	[1,1,1,1]	[]
Имена загружаемых DLL	dllnames	cmdValve.dll	test.dll
Имена оборудования	componentnames	a1	a1
Имена файлов проектов для отладки	prjnames	Valve_control_2.prt	
Создавать по умолчанию не существующие	useemptyvars	Нет	Нет
Количество потоков	nthread	1	1

Рис. 12. Определение параметров подключаемой библиотеки

- После определения параметров блока «Внешняя dll» в схеме на ее графическом отображении появится слева 3 входных порта и справа 5 выходных портов. Эти порты следует соединить с остальными блоками модели, как и предыдущих проектах (рис. 13).

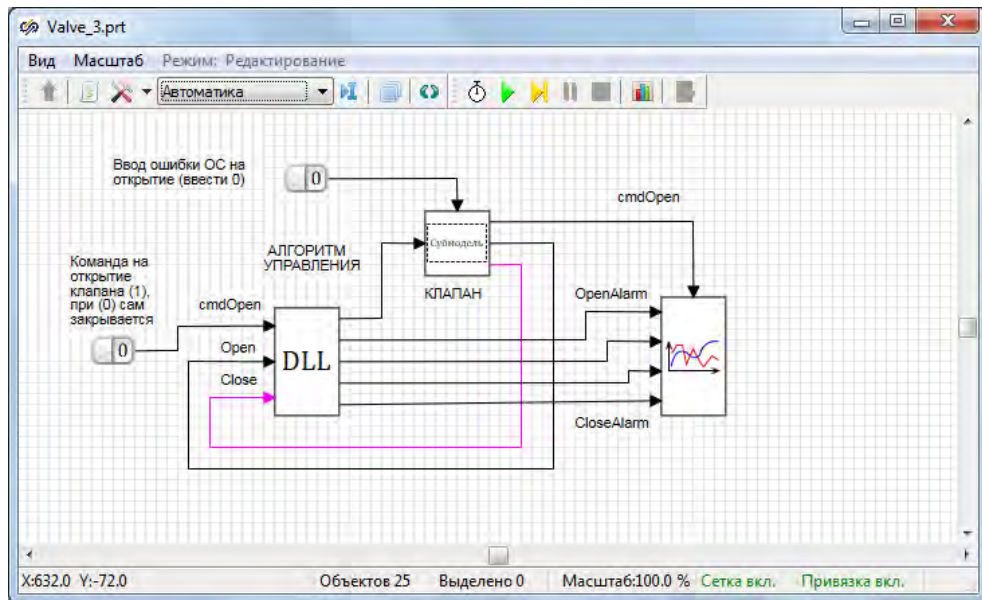


Рис. 13. Новая модель системы управления клапаном

Закончив все эти преобразования, будет получена модель системы управления клапаном, в которой алгоритм управления хранится во внешней dll-библиотеке. Эту модель можно запустить для исследования системы. Как видно из рис. 14, результаты работы системы будут аналогичны ранее полученным, что подтверждает работоспособность сгенерированной на языке Си внешней dll-библиотеки.

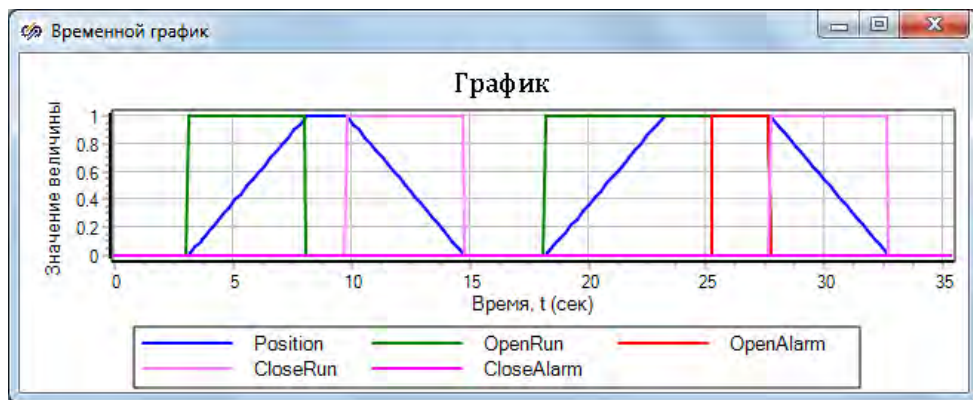


Рис. 14. Результаты работы модели системы

Коротко о кодогенерации. В рассмотренном примере использовался стандартный кодогенератор, для работы которого достаточно было в главном меню выбрать «Инструменты» → «Сгенерировать программу». В результате работы данного генератора в директории с исходным проектом появляются новые файлы, содержащие исходные коды на Си, созданные на основании схемы модели:

<имя проекта>.h; <имя проекта>.inc; <имя проекта>.log;

<имя проекта>_init.inc; <имя проекта>_state.inc;

и файл с динамической библиотекой:

<имя проекта>.dll

Если обратиться к настройкам среды SimInTech, и в главном меню выбрать «Кодогенератор → Кодогенератор Си», а затем в диалоговом окне открыть закладку «Настройки», то в строке «Директории шаблона кода» можно увидеть текст:

```
%codetemplates%MinGW_DLL\
```

Данный шаблон позволяет генерировать код Си и динамическую библиотеку для расчетов с использованием плавающей точки. Вместе с тем можно использовать и другие, встроенные в SimInTech шаблоны (рис. 15).

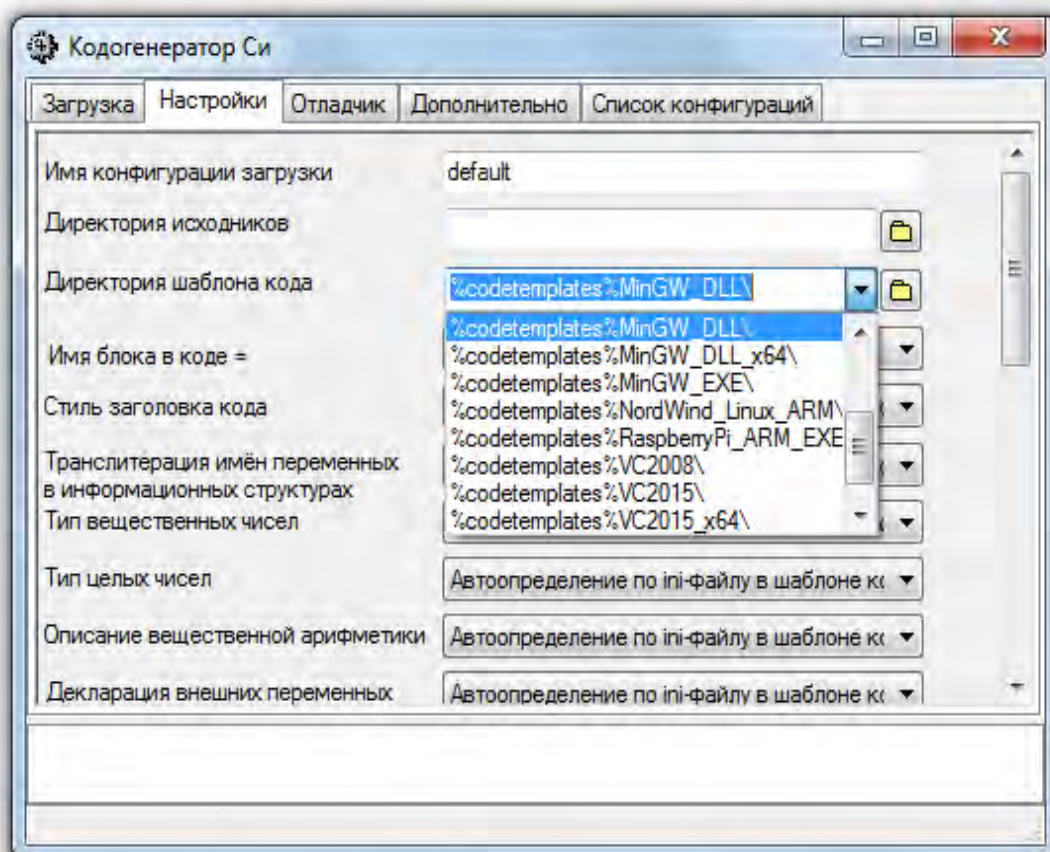


Рис. 15. Выбор шаблона кодогенерации Си

В частности, для получения кода, который выполнял бы вычисления с фиксированной точкой, необходимо в настройках для директории шаблона кода выбрать в выпадающем меню строку:

```
%codetemplates%FixPoint_16_16_MinGW_DLL\
```

Следует отметить, что SimInTech аналогично позволяет выполнять генерацию кода и на языке ST, который используют контроллеры Шнайдер Электрик. Для этого случая также предусмотрены несколько вариантов реализации проектов на языке ST.

Библиографический список

1. SimInTech — Среда динамического моделирования. [Электронный ресурс] — Режим доступа: открытый ресурс, постоянный адрес в Интернете: — <http://simintech.ru/> — Загл. с экрана.

2. Хабаров С. П., Шилкина М. Л. Распределенная обработка данных. Построение распределенных моделей в системе SimInTech. // С. П. Хабаров. – СПб.: СПбГЛТУ, 2018.– 124 с.
3. Амбросовский В.М., Коренев А.С., Хабаров С.П. Распределение упоров в задаче позиционирования подвижных объектов. Известия СПбГЭТУ ЛЭТИ. 2013. № 7. С. 63-69.
4. Амбросовский В.М., Хабаров С.П., Коренев А.С. Адаптивный фильтр калмана к внешним возмущениям. В сборнике: информационные системы и технологии: теория и практика. Сборник научных трудов. отв. редактор А. М. Заяц. – СПб.:СПбГЛТУ, 2015, №7, с. 5-10.
5. Gurov S.V., Habarov S.P., Utkin L.V. Safety analysis of a multi-phased control system. Microelectronics reliability. 1997. Т. 37. № 2. С. 243-254.
6. Хабаров С. П., Шилкина М. Л. Вычислительные машины, системы и сети: Учебное пособие // С. П. Хабаров. - СПб.: СПбГЛТУ, 2017. - 240 с.
7. Хабаров С. П., Жук Ю. А. Сетевые технологии взаимодействия Ubuntu и Windows платформ // С. П. Хабаров. - СПб.: Наука и техника, 2013. - 369 с.
8. Хабаров С.П., Заяц А.М. Использование технологии WebSocket в клиент-серверных экспертных системах. // Леса России: политика, промышленность, наука, образование / материалы научно-технической конференции. Том 1 – СПб.:СПбГЛТУ, 2017. – с.278-280
9. Заяц А.М., Хабаров С.П. Организация доступа к беспроводным AD HOC сетям информационных систем мониторинга лесных территорий из среды Windows 10. // Известия Санкт-Петербургской лесотехнической академии. – СПб, 2018, Вып. 223. - с. 285-299.
10. С.П.Хабаров. Организация гетерогенных ЛВС с терминальным доступом между ее узлами. //Известия СПбЛТА-СПб., 2016, № 216, с. 267-280
11. Вагизов М.Р. Разработка интерактивных геоинформационных систем: принципы построения и конструирования системы. // Сборник научных трудов "Информационные системы и технологии: теория и практика" — СПб.:СПбГЛТУ, 2017. № 9. — с.94-108.

М.Л. Шилкина кандидат технических наук, доцент
Кафедра информационных систем и технологий
СПб ГЛТУ им. С.М.Кирова
mchernobay@inbox.ru

ТЕХНИЧЕСКАЯ ОПТИМИЗАЦИЯ ВЕБ-ПРИЛОЖЕНИЯ ДЛЯ УЛУЧШЕНИЯ ПОЗИЦИЙ В ПОИСКОВОЙ ВЫДАЧЕ

Изменения, которые вносятся на сайт, влияют на оценку сайта поисковыми системами. Но поисковые системы реагируют на изменения не сразу.

Через определенные периоды времени в поисковых системах можно наблюдать динамический процесс ранжирования сайтов и изменения позиций веб-приложения. При этом очень важно не допускать негативной оценки поисковыми системами сайта. Такую оценку сайт может получить, например, из-за несвоевременной оплаты хостинга, размещения неуникальных текстов, нерабочих форм заказов или e-mail адресов. Большое значение для продвижения имеет техническая оптимизация сайта, которая позволяет выявить проблемы, связанные с индексацией сайта поисковыми системами, избавиться от дублей страниц, определить корректное зеркало сайта и др. Исправление технических ошибок не только улучшает индексацию сайта, но и способствует повышению позиций сайта в поисковых системах.

Техническая оптимизация сайта - это комплекс мер, направленных на корректировку технических аспектов сайта, с целью улучшить его взаимодействие с роботами поисковых систем. Основная задача технической оптимизации заключается в обеспечении максимально полной и быстрой индексации страниц сайта поисковыми роботами, потому что такой сайт не выдает ошибки и сбои в работе, в нем присутствуют название и описание страниц для роботов, корректно работают все модули и сервисы. Продвигаемый сайт не получит высокую оценку со стороны поисковых систем в случае наличия дублей страниц и других нарушений технических требований Яндекса [1], так как при оценке ресурса, в первую очередь, поисковый робот индексирует исходный код страниц сайта. Рассмотрим необходимый минимум работ по технической оптимизации веб-приложения на примере сайта, созданного с помощью популярной CMS Joomla:

1. 301-редирект со всех страниц неглавного зеркала сайта на соответствующие страницы главного, то есть заменить адреса всех внутренних ссылок с дубля главной страницы на главную страницу.

Например, в многоязычных сайтах в адресной строке после имени сайта указывается префикс языка типа /ru/, в том числе и по умолчанию в большинстве шаблонов Joomla. С другой стороны, количество многоязычных сайтов ничтожно мало, поэтому в большинстве случаев возникает проблема, как убрать добавляющийся префикс /ru/ к ссылкам на сайте Joomla. Для этого сначала в административной панели надо установить язык по умолчанию Русский: перейти в меню "Расширения -> Менеджер языков" и убедиться, что напротив русского языка стоит желтая звезда, в противном случае кликнуть на звезду напротив русского языка - это сделает русский язык языком по умолчанию. Далее надо перейти в меню "Расширения -> Менеджер плагинов -> Система - Фильтр языка" и переключить опцию "Удалять префикс языка по умолчанию" в состояние "Да".

Также надо избавиться от index.php в конце адресной строки. В Joomla это делается из административной панели Общие настройки -> Сайт -> Настройки SEO -> Перенаправление URL -> Да.

2. Все имеющиеся дубли страниц необходимо исключить из индекса. Для этого используется склейка 301 редиректа. Примеры самых распространенных дублей: страницы для печати, страницы результатов поиска по

сайту, специфические дубли CMS и др. Чтобы поисковый робот не индексировал подобные дубли страниц, надо скорректировать файл robots.txt, находящийся в корне сайта, добавить в него директивы:

```
User-agent: *  
Disallow: /*openstat*  
Disallow: /*?utm*  
Disallow: /*&utm*  
Disallow: /*?from*  
Disallow: /*&from*  
Disallow: /*gclid*  
Disallow: /*yclid*  
Disallow: /*ymclid*  
Disallow: /*?tid*  
Disallow: /*&tid*  
Disallow: /*.swf  
Disallow: /*search  
Disallow: /*cart  
Disallow: /*user/  
Disallow: /*dirDesc  
Disallow: /*by, *  
Disallow: /administrator/  
Disallow: /bin/  
Disallow: /cache/  
Disallow: /cli/  
Disallow: /components/  
Disallow: /includes/  
Disallow: /installation/  
Disallow: /language/  
Disallow: /layouts/  
Disallow: /libraries/  
Disallow: /logs/  
Disallow: /modules/  
Disallow: /plugins/  
Disallow: /tmp/
```

```
User-agent: Googlebot  
Disallow:
```

Эти директивы закрывают все служебные папки CMS, находящиеся в корне сайта, а, следовательно, и все вложенные в них папки, для User-agent * и User-agent Googlebot. Дополнительно надо закрыть все служебные файлы CMS, если они есть в корне сайта для User-agent * и User-agent Googlebot, то есть добавить строки Disallow: <имя служебного файла CMS> в robots.txt. Кроме служебных папок CMS, перечисленные директивы закрывают от индексации поисковыми роботами страницы с запросами на поиск информации по сайту, с содержимым корзины покупателя, с флеш-

анимацией, страницы счетчика посещаемости OpenStat, с сортировками по цене, наименованию и т.д. (dirDesc), а также страницы, автоматически помеченные с помощью инструмента Google Analytics, который называется GCLID (это уникальный параметр, который присваивается при каждом клике по объявлению из рекламной кампании сайта) и помеченные вручную метками UTM. Использование меток GCLID и UTM позволяет точно отслеживать эффективность ключевых слов, определить по какому слову пользователь перешел на сайт, откуда этот пользователь, из какой рекламной кампании это слово и какова стоимость этого клика. Опираясь на эти показатели можно вычислить рентабельность своих инвестиций в определенную кампанию оптимизировать ее, но учитывать в индексе поисковиков страницы с метками GCLID и UTM не надо, так как они перегружают индекс дублиями и затрудняют поиск. Аналогично со страницами, помеченными метками umclid и uclid Яндекс Директом. Если на сайте дополнительно установлен компонент магазина электронной торговли, например, Virtuemart, то в robots.txt также надо закрыть от индексации следующие дубли:

```
Disallow: /*?do*
Disallow: /*?categorylayout*
Disallow: /*?virtuemart_manufacturer_id*
Disallow: /*?page*
Disallow: /*&category_id*
Disallow: /*&option*
Disallow: /*&itemid*
Disallow: /*?tmpl*
Disallow: /*&dir*
Disallow: /*&showcategory*
Disallow: /*&showproducts*
Disallow: /*&productsublayout*
Disallow: /*notify
Disallow: /*?start*
Disallow: /*results,*
```

Эти директивы закроют для индексации дубли страниц с нечеловекопонятными URL карточек товаров, страниц категорий, производителей и результатов поиска по магазину.

Также рекомендуется добавить в раздел <head> всех веб-страниц сайта, закрытых в файле robots.txt для User-agent *, кроме служебных, мета-тег <meta name="googlebot" content="noindex">. То есть код файлов index.html, находящихся в папках, закрытых для индексирования в robots.txt, должен иметь вид:

```
<!DOCTYPE html>
<html><head>
<title></title>
<meta name="googlebot" content="noindex" />
</head>
<body></body>
</html>
```

Кроме того, рекомендуется закрыть от индексации все исходящие ссылки (если, например, на сайте есть ссылки на сторонние ресурсы). Иначе часть ссылочного веса страниц будет постепенно переходить на сторонние ресурсы. Для этого ссылки в статьях должны оформляться с атрибутом "nofollow":

```
<a href="http://www.адрес.ru/" rel="nofollow"> ... </a>
```

3. Прописать уникальные метатеги description, keywords и <title> для всех страниц сайта.

При продвижении сайта любого типа очень важно правильно составлять метатеги title, description, keywords. Если это сделать грамотно, можно в несколько раз увеличить количество просмотров сайта, потому что метатеги - это текстовые сниппеты, описывающие содержимое страницы сайта, и при запросах пользователей на поиск какой-либо интересующей их информации, чем удачнее были составлены метатеги, тем больше посетителей получат сайт в выдачах.

Description - это мета-тег, содержимое которого является описанием страницы. Этот информационный блок видит человек под названием сайта в результатах выдачи поисковиков. Поэтому чтобы заинтересовать пользователя, описание должно быть ёмким (около 150 символов с пробелами), содержать акцент на теме страницы, связанной с заголовком <title>, хорошо добавить несколько эпитетов, чтобы эмоционально заинтриговать пользователя, также полезно кратко упомянуть сильные стороны ресурса, его конкурентные преимущества и добавить призыв к действию - позвонить, перейти и т.д. В простейшем случае для электронного магазина метатег description должен содержать название товара и номер телефона, куда позвонить для заказа:

```
<meta name="description" content="Заголовок страницы в браузере. Наш телефон в Санкт-Петербурге: +7 (812) 111-11-11"/>
```

В CMS Joomla чтобы присвоить странице со статьей уникальный мета-тег, нужно в административной панели открыть Материалы->Менеджер материалов, найти и открыть нужную статью, затем перейти на вкладку Публикация и заполнить поле "Мета-тег Description", например, "Продажа сухих лесоматериалов в Санкт-Петербурге. Тел. +7(812)111-11-11". Рядом находится поле "Мета-тег Keywords", которое тоже рекомендуется заполнять для увеличения количества выдач в поисковых системах. Также уникальный метатег можно присвоить любому пункту меню, например, с блогом статей, относящихся к определенной категории. Для этого надо открыть Меню->Менеджер меню, затем выбрать и открыть нужное меню, а в нем выбрать и открыть нужный пункт меню, перейти на вкладку Метаданные и заполнить те же поля "Мета-тег Description" и "Мета-тег Keywords". Кроме того, если здесь же перейти на вкладку Параметры страницы, то в ней находится поле "Заголовок страницы в браузере" – содержимое этого поля будет отображаться теге <title> кода страницы и использоваться в поисковиками в качестве основной информации, связанной с данной страницей при поиске. Поэтому к заполнению этого поля надо подходить вдумчиво, чтобы максимально

поднять порядковое место страницы в выдачах по запросам, относящимся к содержанию страницы. Наконец, в магазине Virtuemart CMS Joomla можно присвоить уникальные метатеги каждой странице с карточкой товара. Для этого надо открыть веб-приложение в файловом менеджере, найти файл по пути

`public_html/templates/ваш_шаблон/html/com_virtuemart/productdetails/default.php`, открыть его и найти строку (или добавить ее в начало после `<?php`):

```
defined('_JEXEC') or die('Restricted access');
```

После нее надо добавить строки следующего содержания

```
$doc = JFactory::getDocument();
$doc->setTitle($this->product->product_name.' - Компания «Мир
древесины»');
$doc->setDescription($this->product->product_name.'. Наш телефон
в Санкт-Петербурге: +7(812)111-11-11');
```

Эти команды установят требуемое значение метатега `description` и тега `<title>` в карточке товара.

Аналогично можно установить значение метатега `description` и `<title>` странице со списком продуктов определенной категории. Для этого надо открыть файл `categories.php` (или `vertical.php` – смотря по тому, какой шаблон для вывода категории товаров используется), который находится по пути `public_html/templates/ваш_шаблон/html/com_virtuemart/category/`, и добавить строки

```
$document = JFactory::getDocument ();
$desc=$document->getDescription ();
if (empty($desc)) {
$document->setTitle($this->category->category_name.' -
Компания «Мир древесины»');
$document->setDescription($this->category-
>category_name.'. Наш телефон в Санкт-Петербурге:
+7(812)111-11-11');
}
```

Так же для страниц с товарами определенного производителя можно задать значения метатега `description` и `<title>`, если изменить файл `public_html/templates/ваш_шаблон/html/com_virtuemart/manufacturer/default.php`, добавив в него строки

```
$doc->setTitle('Каталог'.$this->manufacturer-
>manufacturer_name.' - Компания «Мир древесины»');
$doc->setDescription($this->manufacturer-
>manufacturer_name.'. Наш телефон в Санкт-Петербурге:
+7(812)111-11-11');
```

Правильный выбор тега <title> имеет определяющее значение для позиционирования ресурса в выдаче. При этом в строке значения <title> первыми должны идти ключевые слова, потому что поисковые системы придают больший вес словам, находящимся вначале. Поэтому, основную тему статьи (если это блог) или название товара (если это интернет-магазин) надо ставить в начало заголовка. Еще рекомендуется включать в заголовок как популярные, так и менее популярные запросы, потому что по популярным запросам конкуренция очень высокая, из-за чего быстро попасть на первые позиции в поиске не получится. Но по менее популярным запросам можно сразу начать получать первых посетителей. Например, по запросу "сухие строительные материалы" поисковики выдают названия крупных компаний типа "Петрович", а по запросу "сухие стройматериалы купить дешево спб" будут выданы уже совсем другие компании. Узнать популярность запроса можно с помощью сервиса <https://wordstat.yandex.ru/>. Чем более цепляющий и интригующий <title> - тем больше людей будут приходить на сайт. Видя заинтересованность со стороны пользователей, поисковые системы начнут поднимать его позиции. Чем выше сайт в выдаче - тем он успешнее. По статистике первые три сайта из органической выдачи получают 61.5% всех переходов, поэтому надо стараться попасть не в десятку, а в первую тройку сайтов, выданных по определенному запросу. Для этого содержание <title> должно вызывать эмоции, содержать эпитеты и числа (например, "10 советов выбрать качественный долговечный пиломатериал для строительства. Пошагово и просто."), длина заголовка <title> не должна превышать 60 символов, остальные все равно будут игнорироваться поисковыми роботами.

После тега <title> и метатега description на результирующую позицию в выдаче поисковика также влияет метатег keywords. Раньше keywords играл большую роль в ранжировании сайта. Достаточно было перечислить в нем все ключевые слова, как сайт без труда выходил на первые строчки в поиске. Но сейчас такое излишнее перечисление ключевых слов может быть расценено поисковыми системами как попытка манипуляции, что в свою очередь может привести к санкциям. С другой стороны, нет ничего страшного если указать несколько ключевых запросов. Они могут учитываться при определении соответствия страницы поисковым запросам.

4. Настроить обработку 404 ошибки.

Ошибка «404 Not Found» может появиться в следующих случаях: страница была перемещена (адрес страницы поменялся), допущены ошибки в написании адреса страницы или страница была удалена. Чтобы снизить вероятность ухода посетителей с сайта, необходимо настроить и оформить страницу 404 ошибки в стиле дизайна сайта, а также дать ссылку, по которой посетители смогут перейти на существующую страницу сайта (например, на главную страницу). В противном случае посетители могут уйти с сайта. Чтобы настроить обработку 404 ошибки в веб-приложении, созданном в CMS Joomla, надо:

- создать материал под названием, например, 'Error 404' в категории 'Uncategorized', включить в него немного текста, например, *Извините, но мы*

не смогли найти запрошенную Вами веб-страницу ... Вы можете [перейти на главную страницу сайта](http://www.site.ru/), на вкладке Публикация в поле Мета-тег Robots выбрать No index, no follow, чтобы исключить созданную страницу из поисковых двигателей;

- создать ссылку на этот новый материал с названием '404': в меню, для которого нет ни одного модуля, создать пункт меню (с заголовком, например, "Ошибка 404 страница не найдена"), который ведет на этот новый материал 'Error 404', в поле Алиас надо написать error-404 и щелкнуть по кнопке 'Сохранить и закрыть';

- в файловом менеджере открыть нужный ресурс и найти файл public_html/templates/ваш_шаблон/error.php, удалить содержимое и вставить

```
<?php
defined('_JEXEC') or die;
if (!isset($this->error))
{
    $this->error = JFactory::raiseWarning(404,
    JText::_ ('JERROR_ALERTNOAUTHOR'));
    $this->debug = false;
}
$doc = JFactory::getDocument();
$app = JFactory::getApplication();
$this->language = $doc->language;
$this->direction = $doc->direction;
?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"
xml:lang="<?php echo $this->language; ?>" lang="<?php
echo $this->language; ?>" dir="<?php echo $this-
>direction; ?>">
<?php
if (($this->error->getCode()) == '404') {
header('Location: /error-404');
exit;
} ?>
</html>
```

5. Создать карту сайта. html, включив в нее ссылки на основные разделы сайта (все страницы, кроме карточек товаров, новостей, статей).

Карта сайта – это полный каталог всех разделов сайта, с кратким описанием каждого раздела. Картой удобно пользоваться для быстрого поиска интересующей информации и перехода по ссылкам. Полная карта сайта необходима для достижения высоких позиций в поисковой выдаче. Поисковые

системы высоко оценивают сайты, где есть навигационные карты. Поэтому важно создать карту сайта .xml, по возможности не применяя параметры <lastmod>, <changefreq>, <priority>, и исключив страницы закрытые в файле robots.txt, затем разместить карту сайта .xml в корневой директории сайта и, наконец, добавить ссылку на карту сайта .html в футере всех страниц сайта.

Карты сайта могут быть в формате xml и в формате Html, причем формат html представляет из себя обычный список ссылок на все материалы web ресурса, который доступен любому желающему по нажатию на соответствующий пункт меню, это способствует ускорению и улучшению индексации ресурса поисковыми системами. Но основным инструментом, предназначенным для прямого указания поисковику тех страниц ресурса, которые нужно будет обязательно проиндексировать, является файл под названием Sitemap.xml (это его наиболее частое название, но можно присвоить и другое имя), который не будет виден посетителям web проекта. Он составляется с учетом специального синтаксиса, понятного поисковым машинам, где будут перечислены все страницы, подлежащие индексации с указанием степени их важности, даты последнего обновления и примерной частоты обновления, это ускорит индексацию роботами ресурса, иначе робот начнет бессистемно просматривать файловые ресурсы веб-приложения, а потом прекратит вследствие того, что у ботов есть определенные лимиты на время и количество просмотренных документов, и содержание ресурса останется непроиндексированным в течение долгого времени. В целом, по статистике [2] (см. рис.1) на первом месте среди причин, затрудняющих продвижение сайта, стоит не-уникальное содержание веб-ресурса; на втором – отсутствие карты сайта в формате xml или ее несоответствие признанному формату создания; на третьем – отсутствие или неправильное заполнение файла robots.txt, который указывает роботам поисковиков, на индексацию каких файлов не стоит тратить время (объекты движка, например), а также с помощью него можно закрыть от индексации и часть страниц для устранения эффекта дублирования страниц, который присущ многим CMS.

В отличие от robots.txt, который создается вручную, файл карты сайта в формате xml, рекомендуется создать каким-либо автоматическим способом, особенно если проект большой и часто обновляемый. Практически для каждой CMS есть расширение, которое позволит создать, а при появлении новых материалов и автоматически пересоздать файл карта сайта. Как альтернатива, можно воспользоваться онлайн генератором карты сайта в виде готового решения (<https://www.xml-sitemaps.com/>), или написать вручную для небольшого проекта. В Joomla для автоматической генерации кор-ректной карты сайта надо из административной панели установить бес-платное расширение OSMaP Free (<https://www.joomlashack.com/docs/osmap/start/>).

После того, как карта сайта создана, ее надо добавить в Яндекс и Google вебмастер – это можно сделать двумя способами. Первый способ – добавить в конце файла robots.txt строку с директивой Sitemap, например:

```
Sitemap: https://домен_ресурса/sitemap.xml
```

TOP-10 технических ошибок пользователей

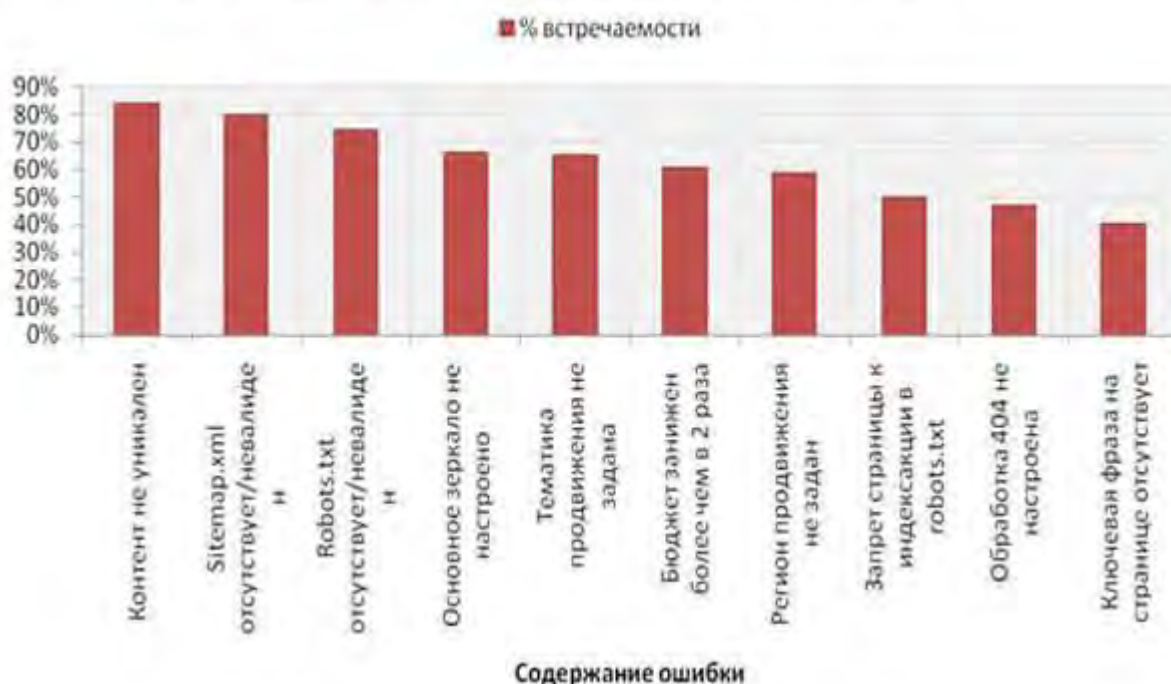


Рис.1. Причины низких позиций ресурса в выдачах поисковых систем

Обязательно нужно прописывать полный абсолютный путь до карты сайта, не опуская имя протокола и доменное имя, хотя во всех остальных директивах robots.txt пишутся относительные пути. Если используется карта сайта, написанная не вручную, а генерируемая компонентом системы, например, OSMap Joomla, то директива выглядит следующим образом:
Sitemap: https://домен_ресурса/component/osmap/?view=xml&id=1&lang=ru

Чтобы получить строку полного пути до xml-карты сайта, надо из административной панели выбрать Компоненты->OSMap Free, затем возле имени созданной карты сайта щелкнуть по ссылке XML – откроется xml-карта сайта, ее адрес надо скопировать и вставить после директивы Sitemap: в файле robots.txt. Поисковые роботы при очередном заходе на web проект просмотрят содержимое robots.txt и загрузят для изучения карту сайта.

Второй способ передать информацию о местоположении карты сайта поисковикам – напрямую, без посредничества robots.txt. Делается это через интерфейс Яндекс Вебмастера и панели инструментов Google. Сначала надо зайти в Яндекс аккаунт, затем открыть <https://webmaster.yandex.ru/>, добавить сайт, после чего вебмастер выдаст код верификации, который надо вставить внутрь тега <head> главной страницы сайта. Для ресурса в CMS Joomla для этого надо открыть этот ресурс в файловом менеджере и найти файл public_html/templates/ваш_шаблон/index.php, открыть его и добавить внутрь тега <head> </head> строку с мета-тегом яндекс-верификации <meta name="yandex-verification" content="полученный код верификации">, сюда же добавляется метатег верификации Google. Далее в Яндекс Вебмастере на странице, откуда копировался код верификации, надо нажать кнопку Проверить

ритель. После этого, если верификация прошла успешно, в панели Яндекс Вебмастера Настройки->Права доступа появится строка с именем пользователя ресурса и рядом проверенный код верификации. Теперь надо перейти на страницу Яндекс Вебмастера Индексирование->Файлы Sitemap и добавить сгенерированную карту сайта, вписав полный абсолютный путь к карте сайта на хостинге. Затем можно нажать кнопку "Отправить файл Sitemap на переобход" для скорейшей индексации ресурса в соответствии с картой, и через некоторое время, если индексация прошла успешно, в поле Статус появится значение ОК.

Аналогично можно получить код верификации Google – зайти в свой аккаунт, затем в <https://search.google.com/search-console/settings> добавить ресурс, затем в Google Search Console зайти в Настройки, скопировать из поля Тег HTML строку кода вида

```
<meta name="google-site-verification" content="код верификации" />
```

и добавить ее в тег <head> главной страницы сайта, далее в Google Search Console Настройки нажать Проверить. Если проверка прошла успешно, появится надпись "Вы являетесь подтвержденным владельцем". После этого надо в Google Search Console перейти в Файлы Sitemap и добавить абсолютный путь к карте сайта, если карта корректна, в поле Статус появится слово Успешно.

Теперь, когда сайт добавлен в Google Search Console, можно анализировать динамику количества посетителей, количества кликов, вида устройств, с которых посетители заходят, относительный состав источников просмотра сайта (прямые заходы, поисковые системы, ссылки, соцсети, внутренние переходы), а также вид запросов пользователей и количество просмотров ресурса по каждому из этих запросов. Все это можно видеть на вкладке Google Search Console Эффективность.

Через панель Яндекс вебмастера проводится анализ сайта для исключения из индекса «технического спама», а также корректной настройки индексации и работы ресурса в целом, и формируется перечень необходимых работ по технической оптимизации ресурса.

6. Настроить доступы для анализа посещаемости ресурса в сервисах статистики Яндекс.Метрика и Google Analytics.

Сервисы Яндекс Метрика и Google Analytics позволяют отследить количество посетителей, которые пришли на сайт, анализировать их поведенческую деятельность и проводить анализ продуктивности рекламных фирм. Сервисы являются эффективным инструментом для оценки посещаемости. Яндекс.Метрика отличается от простых счетчиков, тем, что позволяет следить за поведением посетителей сайта на базе модулей «срезов» и «целей». Такое решение может сделать разделение посетителей на группы, принимая во внимание целый ряд параметров и сведений о поведении. Яндекс Метрику можно эксплуатировать в качестве обычного счетчика, например, для определения посещаемости ресурса и для общей оценки продуктивности рекламных действий в Google Adwords или Яндекс Директ, а также для поисковой раскрутки сайта.

Чтобы статистика собиралась на серверах Яндекс и Google, пользователь должен разместить Java-script код на страницах своего сайта.

Чтобы создать счетчик Яндекс Метрики, надо зайти в аккаунт Яндекс, который является подтвержденным владельцем веб-приложения (сайта), перейти на <https://metrika.yandex.ru/list/> и добавить счетчик для ресурса. Кликнуть по добавленному счетчику и выбрать в левой панели Настройка. Внизу открывшейся страницы располагается код счетчика Яндекс Метрики, который надо скопировать и добавить в код каждой страницы сайта. В Joomla можно, например, в административной панели в меню Расширения->Модули добавить модуль типа HTML-код, выбрать для него какую-либо позицию активного шаблона, указать на вкладке "Привязка к пунктам меню" значение "На всех страницах", а скопированный код вставить в поле редактирования вкладки Модуль и сохранить.

Для добавления счетчика статистики Google Analytics производятся аналогичные действия: открывается google-аккаунт подтвержденного в Google владельца сайта, затем надо перейти на <https://analytics.google.com/analytics/web/>, далее в левой панели щелкнуть на ссылке Администратор, на открывшейся странице в центре выбрать ресурс, затем щелчком развернуть пункт списка Отслеживание->Код отслеживания. Далее нужно скопировать появившийся код и вставить его в код каждой страницы сайта аналогично тому, как это было сделано для Яндекс Метрики.

7. Переход с HTTP протокола на HTTPS улучшит оценку сайта поисковиками.

Сообщения, переданные с помощью HTTP-протокола являются обычной текстовой информацией, что делает её уязвимой. В первую очередь это критично для ресурсов, оперирующих персональной информацией пользователей, например, номерами кредитных карт, логинами-паролями и прочее. Для защиты от атак, основанных на прослушивании и подмене содержимого сайта, был создан защищенный HTTPS-протокол, это обычный HTTP, но работающий через шифрованные транспортные механизмы SSL. Но, даже если веб-приложение не является магазином с платежной системой, все равно использование http протокола ухудшит ранжирование сайта поисковыми системами. Связано это с тем, что все большее количество пользователей используют бесплатные точки доступа Wi-Fi в кафе, транспорте, гостиницах, которые на самом деле используются как точки для интернет-рекламы. Так как в случае, если текст передается через провайдера в прямом виде (http), то нетрудно модифицировать его, например, встроив скрипт или фрейм, добавляющие дополнительный функционал, а именно рекламу. Таким образом, сайты на http, просматриваемые пользователями с бесплатного Wi-Fi, встраивают рекламу в сайты без ведома владельцев этих сайтов. Реклама на сайте уменьшает время, проведенное пользователем на сайте (dwell-time), что снижает, с одной стороны, авторитет сайта, с другой стороны портит поведенческие факторы (действия пользователей на сайте). И то и другое плохо для SEO, так как dwell-time используется поисковиками в различных ранжирующих и антиспам-алгоритмах. Чем больший процент аудитории какого-

либо проекта использует бесплатный Wi-Fi, тем хуже для его ранжирования, если этот проект использует незащищенный HTTP-протокол.

Для сайта, использующего CMS Joomla, при переходе с http протокола на https:

- необходимо заменить ссылки для подключения внешних скриптов на подключение через https, а также ссылки на внутренние страницы сайта в 301-редиректах файла .htaccess тоже поменять на https. Все ссылки на внутренние страницы в материалах сайта должны быть относительными;

- из панели управления хостера надо установить SSL сертификат, платных и бесплатных видов сертификатов SSL много, обычно достаточно бесплатного сертификата Let's Encrypt, предлагаемого большинством хостингов;

- в панели управления хостинга установить флажок "Переадресовывать запросы с http на https";

- в административной панели Joomla выбрать Система->Общие настройки->Сервер и в разделе "Настройки сервера" в поле "Включить SSL" выбрать "Весь сайт";

- в панели Яндекс Вебмастер для аккаунта владельца сайта, переходящего на https, в левой панели выбрать Индексирование->Переезд сайта, затем отметить галочкой пункт "Добавить HTTPS", далее перейти в раздел "Файлы Sitemap" и добавить путь с https к карте сайта;

- аналогично в Google Search Console надо добавить проект с адресом через https, подтвердить на него права нажатием кнопки "Проверить" и исправить путь к карте сайта.

8. Преобразования URL на сайте к виду «ЧПУ» («человекопонятные урлы») эффективны и для посетителей, и для поисковых систем.

Например (источник – Яндекс), http://download.yandex.ru/company/experience/Baitin_Korrekcija%20gramotnosti.pdf - данный URL дает поисковому роботу множество информации о документе: его можно скачать; формат, скорее всего, PDF; документ, вероятно, соответствует запросу «коррекция грамотности».

Преимущества для посетителей:

- подобные адреса легко запоминаются;
- понятный путь, прописанный в адресной строке браузера, позволяет пользователю интуитивно определить, в каком разделе сайта он находится, и при желании перейти в раздел выше, удалив ненужные символы [3];

- удобство для восприятия URL на слух;
- в результатах поиска могут быть выделены названия страниц, если происходит совпадение с поисковым запросом, что способствует тому, что именно этот сайт будет привлекать пользовательское внимание.

Преимущества для поисковых систем:

- чересчур усложненный URL, в особенности включающий несколько параметров, вполне может затруднить сканерам работу», а также «имеется

вероятность того, что сам сканер не сможет провести сканирование сайта в полной мере» (источник – Google);

- вид URL может давать представление о том, что содержится на соответствующей странице. Использование транслитерации в адресах страниц также позволит поисковому роботу определить, чему посвящена страница;

- «ЧПУ» оказывает влияние на ранжирование: соответствие URL запросу может добавить сайту больший вес в глазах поисковых систем.

Таким образом, грамотное изменение структуры сайта, приведение всех URL-страниц к виду «ЧПУ» положительно влияет на следующие факторы:

- улучшение ранжирования сайта;
- увеличение трафика по низкочастотным запросам;
- удобство для восприятия пользователем адресов страниц сайта.

Поэтому рекомендуется генерировать на сайте адресные строки, состоящие только из соответствующих ключевых слов и отражающие структуру ресурса.

Например, в Joomla по умолчанию в конце адресной строки страницы с материалом к алиасу добавляется ID материала, также и для страницы с блогом категории статей в адресной строке в конце по умолчанию добавляется ID категории материалов. Это некрасиво, неинформативно и не может быть позитивно оценено ни пользователями, ни поисковиками. В CMS Joomla, начиная с версии 3.8.2, есть решение проблемы генерации красивых ссылок: из административной панели надо зайти в Материалы -> Настройки -> Интеграция -> Маршрутизация URL-> Экспериментальная (выскочит подпункт) -> Удалить идентификаторы из URL -> Да.

9. Дополнительная модификация содержимого сайта с целью увеличения пользовательской аудитории. Для еще более успешного продвижения сайта можно проанализировать статистику пользовательских запросов, связанных с тематикой сайта. Узнать популярность запроса можно с помощью сервиса <https://wordstat.yandex.ru/>. Выявив 15-20 проблем, которыми наиболее часто интересуются пользователи, рекомендуется по возможности опубликовать на сайте авторские материалы с заголовками страниц в браузере, совпадающими с топ-20 пользовательских запросов по рассматриваемой теме. После этого надо не забыть актуализировать карту сайта, чтобы новые статьи были, как можно скорее проиндексированы поисковиками. Если карта сайта генерируется с помощью стандартного расширения (например, OSMap), то она актуализируется автоматически при изменении состава тех меню, которые в административной панели при редактировании карты сайта выбираются в качестве содержимого карты.

В качестве иллюстрации вышесказанного можно проанализировать положительную динамику изменений показателей эффективности в диаграммах счетчика Яндекс Метрики для сайта станции технического обслуживания автомобилей за квартал. В середине квартала сайт временно не работал в связи с технической оптимизацией в соответствии с вышеизложенными этапами.

Затем, после включения, положительные показатели практически удвоились (см. рис.2).

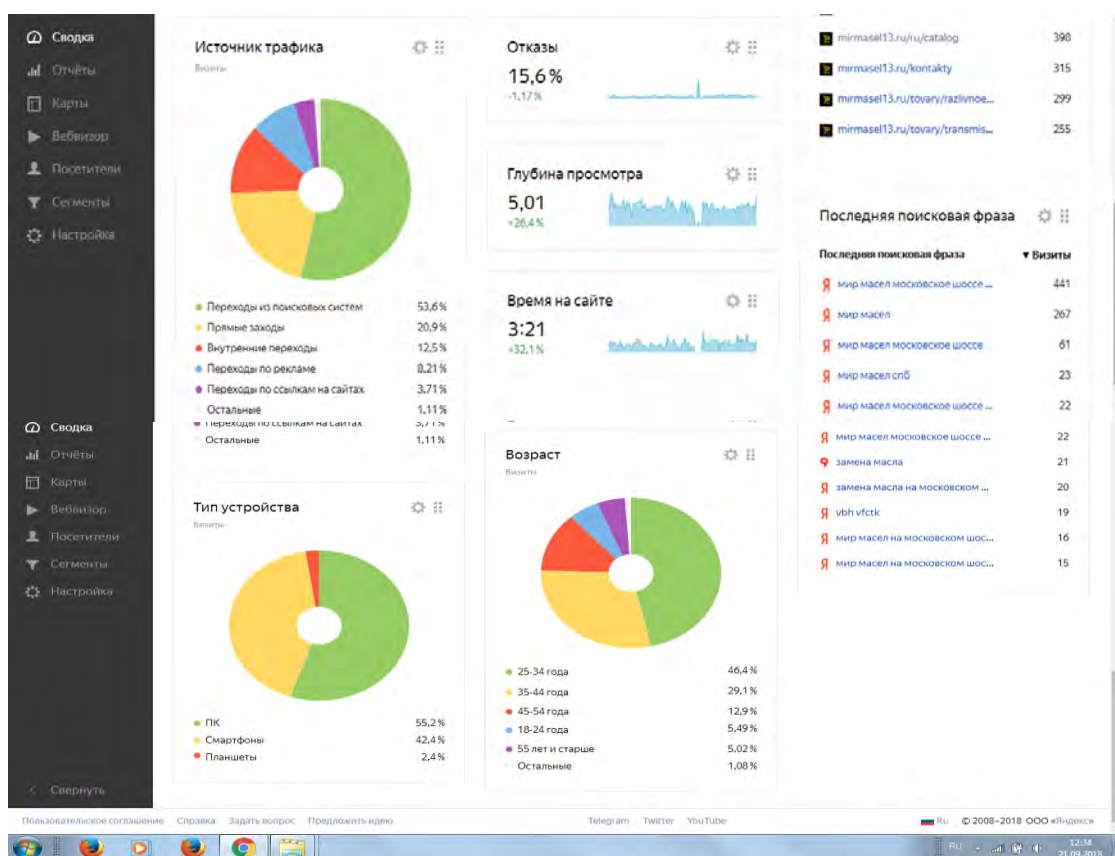
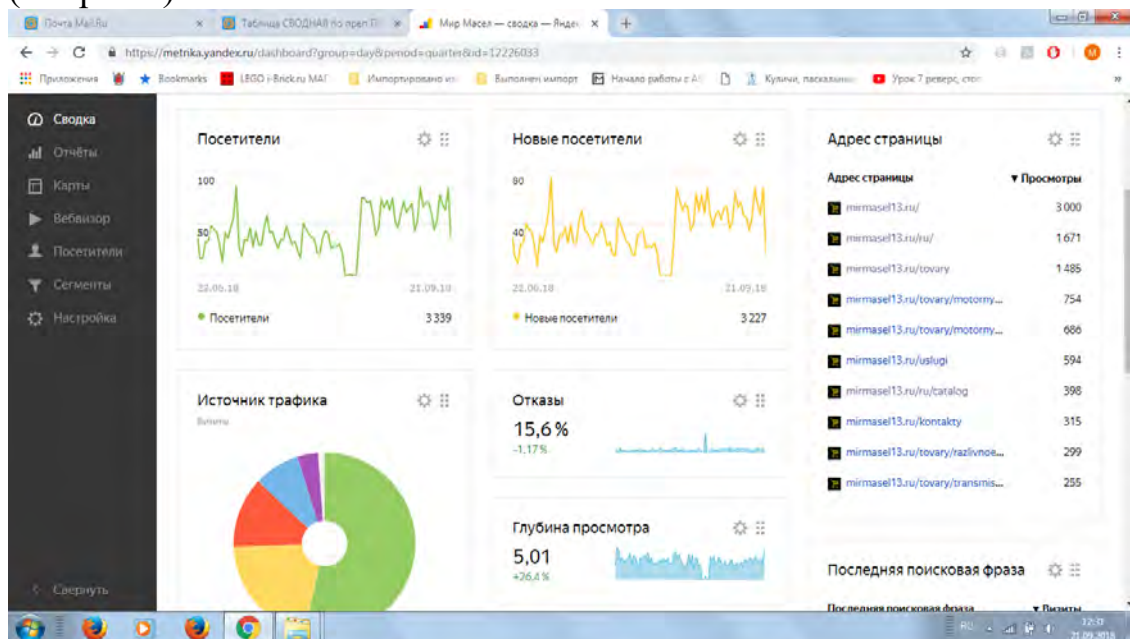


Рис.2. Сводные показатели эффективности сайта за квартал, в середине которого была проведена техническая оптимизация сайта

Библиографический список

1. <https://yandex.ru/support/webmaster/recommendations/sitestructure.html>
https://www.rookee.ru/post/2011/05/12/user-epic-fail.aspx?code=grabr_SL_statia
<https://www.artlebedev.ru/kovodstvo/sections/48>

ОРГАНИЗАЦИЯ ХРАНЕНИЯ И ДОСТУПА К ДИСТАНЦИОННЫМ ДАННЫМ

Взрывной рост объема данных дистанционного зондирования земной поверхности (ДЗЗ) поставил перед разработчиками систем получения данных и предоставления к ним доступа пользователям целый комплекс трудно решаемых проблем, которые связаны с их интеграцией в единую распределенную организацию получения, передачи, хранения и каталогизации большого объема данных. Особенностью таких данных является их разнообразие, определяемое:

- типом используемой для съемки платформы (космические аппараты, авиационные, беспилотники);
- съемочной аппаратурой;
- диапазоном электромагнитного спектра (ультрафиолетовым, видимым, инфракрасным, радио);
- разрешением;
- стандартами, принятыми в стране, осуществляющей запуск и др.

Беспрецедентный объем, разнообразие и скорость доставки данных требуют изменения традиционных способов хранения и анализа данных наблюдения Земли и обмена ими с пользователями. Значительное количество уже накопленных данных не используется в полной мере ввиду отсутствия адекватных инструментов анализа данных, даже в тех случаях, когда проводилась съемка заданных территорий для решения конкретных задач [4]. В последнем случае приходится сохранять еще и избыточную «лишнюю» информацию. Решение многих задач, в частности, задач природопользования связано с необходимостью комбинировать данные из разных источников, предварительно загружать данные для анализа, отслеживать временные ряды изображений все увеличивающимся количеством пользователей. По данным ESA (европейского космического агентства) количество зарегистрированных пользователей спутниковых изображений, получаемых с помощью различных платформ сбора данных наблюдения Земли, возросло за 10 лет в несколько раз: 2010 год - 5000, 2013 год – 10000, 2017 год – 70000 [14].

Хотя в последние годы значительное внимание уделялось развитию систем надежной записи, увеличению скорости передачи, хранения и извлечения больших данных дистанционного зондирования, сокращению времени анализа данных, разрыв между требуемыми характеристиками хранения и обработки данных тоже увеличился.

Ускорение передачи данных

Для ускорения передачи данных до близкой к реальному времени компания Airbus Defence and Space запустила программу SpaceDataHighway, со-

четающую возможности ультраширокополосной лазерной связи и геостационарных ретрансляционных спутников. Лазерная технология в основе SpaceDataHighway позволяет ежедневно передавать до 40 Тбайт данных со спутников ДЗЗ, авиационных систем или с Международной космической станции со скоростью 1,8 Гбит/сек. Технология нового поколения опробована на данных со спутников ДЗЗ Sentinel европейской системы Copernicus [13]. Первый ретрансляционный спутник EDRS-A для программы SpaceData Highway был запущен 30 января 2016 года. Для обеспечения глобального покрытия Азиатско-Тихоокеанский региона будет запущен спутник EDRS-D, который сможет обеспечить двунаправленную связь нескольких лазерных терминалов, чтобы обслуживать одновременно нескольких клиентов. Программа SpaceDataHighway является результатом государственно-частного партнерства между Европейским космическим агентством и Airbus Defence and Space при спонсорстве Германского центра авиации и космонавтики (дистрибьютер в России – Совзонд).

На этапе размещения данных для хранения и их последующего извлечения необходимо обеспечить не только скорость их перемещения, но и решать проблему «проклятия размерности» (Беллман) – использовать способы хранения, позволяющие достигать приемлемое время доступа и анализа данных.

Основные направления увеличения пространства для размещения Баз Данных в настоящее время – потоковая обработка данных (streaming); технология Хранилищ многомерных данных Data Warehouse (в виде многомерных «кубов»); использование облачных технологий, например, облачных баз данных AWS (Amazon Web service); в том числе использование облачных сетей хранения данных (СХД). По мере снижения расходов на хранение данных и вычислений в облаке, появляется новое поколение приложений, которым требуются базы данных для хранения терабайтов и петабайтов новых типов данных, предоставления доступа к данным с миллисекундной задержкой, обработки миллионов запросов в секунду и масштабирования систем для поддержки миллионов пользователей в разных уголках планеты.

Масштабирование вычислительных ресурсов и хранилищ данных предлагается осуществлять с помощью нескольких вызовов API, зачастую даже без простоев. В случае специализированных баз данных, оптимизированных для конкретной модели данных, поддерживаемые такими БД приложения, могут работать и масштабироваться эффективнее, чем приложения на базе универсальных монолитных БД. Полный контроль безопасности данных в этом случае обеспечивается на множестве уровней, включая сетевую изоляцию с помощью Amazon VPC, шифрование данных при хранении с помощью ключей, создаваемых и управляемых через AWS Key Management Service (KMS), а также шифрование данных при передаче. Несмотря на эти меры использование облачных технологий для хранения ДДЗ тормозится сомнениями владельцев данных в достаточном уровне защиты конфиденциальности данных.

В настоящее время Национальное управление по аэронавтике и исследованию космического пространства США (NASA) планирует интегрировать блокчейн с системами навигации и наблюдения за воздушно-космическим пространством. С 1 января 2020 года в США начнёт работу система автоматического зависимого наблюдения-вещания (АЗН-В), которая позволит АЗН-В сравняться или даже превзойти стандарты конфиденциальности и защищённости радиолокационных сетей национальной системы воздушного пространства [15]. В апреле 2018 г. NASA стала одним из спонсоров проекта по постройке автономного космического аппарата, использующего сеть Ethereum.

Исторически хранилища ДДЗ размещались в пунктах приема, организованных компаниями, запускающими аппараты для съёмки поверхности Земли, чаще всего, ориентированными на решение проблемно-ориентированных задач.

Наземные сегменты операторов наиболее известных современных космических систем ДЗЗ, как правило, включают в себя: наземный комплекс приема данных; комплекс обработки данных; комплекс хранения данных. В зависимости от количества предлагаемых оператором услуг и продукции состав наземного сегмента может включать в себя дополнительные комплексы и системы. Некоторые операторы (например, Infoterra) используют в качестве наземных сегментов элементы инфраструктуры национальных космических агентств. Например, в 2008 году был создан Центр получения, распространения и обработки информации дистанционного зондирования Почвы (ДЗЗ) «Самара» (ЦПОИ «Самара»).

Наиболее мощные поставщики ДЗЗ: Космический аппарат (КА) «Ресурс-ДК1» (оператор — НЦ ОМЗ, Россия,) КА Landsat-5,7 (USGS, США), КА RapidEye (RapidEye AG, Германия), КА TerraSAR-X, TanDEM-X (Infoterra, Германия), КА RADARSAT-2 (MDA, Канада), КА COSMO-SkyMed (ASI — Итальянское космическое агентство), КА Proba, Envisat, ERS-2 (оператор: ESA — Европейское космическое агентство), КА GeoEye-1, IKONOS (GeoEye, США), КА ALOS (JAXA — Японское космическое агентство). Используются сервисы прямого доступа, такие, как Direct Access Program (DigitalGlobe, США) ((WorldView-1 и WorldView-2, QuickBird); Infoterra's Direct Access Services (Infoterra GmbH, Германия) [4].

Для российского сегмента хранилища данных о планетах Солнечной системы и их анализа на основе веб-ГИС-технологий – системы накопления, хранения, управления и доступа к данным на основе картографической визуализации – Научным центром оперативного мониторинга Земли, ЦКН Роскосмоса (НЦ ОМЗ) совместно с Фрязинским филиалом Института радиотехники и электроники РАН был разработан геопортал [2], где были реализованы ее составные части:

- разработана модель планетных данных и их описание (создана структура метаданных);
- разработана структура базы данных и осуществляется ее наполнение информацией;

- разработан удобный пользовательский поиск и доступ к результатам обработки.

- При разработке структуры БД была предусмотрена возможность реализации следующих функций:

- ввод данных ДЗЗ и результатов их обработки;

- архивирование, каталогизация и хранение данных ДЗЗ и результатов их обработки;

- каталогизация и хранение дополнительных данных (описание проектов, миссий, справочная информация и др.);

- предоставление данных для предварительной и тематической обработки через локальный доступ;

- предоставление необходимой информации о наличии данных ДЗЗ и результатов их обработки в архиве посредством удаленного доступа через Web-интерфейс;

- защита данных от несанкционированного доступа.

БД обеспечивает хранение и доступ к пространственным данным, хранение метаданных, а также хранение описания типов загруженной информации. При этом растровые данные (ДЗЗ и результаты их обработки) хранятся в одной или нескольких системах [3].

Реализация подобных проектов сложна, требует времени и других ресурсов.

Наиболее распространенным направлением развития доступа к ДДЗ является развитие интеллектуальных интерфейсов, включающее:

- перенос ресурсоемких приложений в среду облачных вычислений, позволяющий использовать веб-ГИС в качестве интернет-сервисов,

- использование распределенной архитектуры, позволяющей работать с пространственно-атрибутивными данными без необходимости их предварительной загрузки на веб-сервер благодаря сохранению в кэше веб-браузера программного кода, необходимого для обработки пространственно-атрибутивных данных ГИС на клиентской стороне приложения (ПО бизнес-логики веб-ГИС-клиента и ПО визуализации данных веб-ГИС-клиента),

- использование технологии GEOSMIS – построения веб-интерфейсов работы с пространственной информацией в системах дистанционного мониторинга, которые предназначены для работы с большими распределенными многомерными архивами спутниковых данных и результатами их обработки. Технология рассчитана на поиск данных и на создание инструментов анализа и управления данными. Основные элементы архитектуры и схема веб-сервиса показаны на рис. 1 [1, 5].

Политика открытого доступа программ Landsat и Sentinel приводит к необходимости использования понятия «big Earth data» [11]. В оптической области спутники Copernicus Sentinel-2 специально предназначены для получения изображений высокого разрешения (HR) в качестве ключевого источника информации Европейской программы Copernicus, которая в настоящее время со спутников Sentinel-2 производит более 1,7 терабайта (Тб) данных в

день для уровня обработки 1С [12]. Систематическое получение значимой информации из этих данных – сложная задача, связанная с технологическими разработками, инновациями в прикладных областях и междисциплинарными исследованиями в области дистанционного зондирования и человеко-компьютерных взаимодействий. Области применения этой информации позволяют оценивать последствия изменения климата или антропогенного воздействия, выполнять мониторинг природных ресурсов, т.е. являются необходимыми для населения Земли.

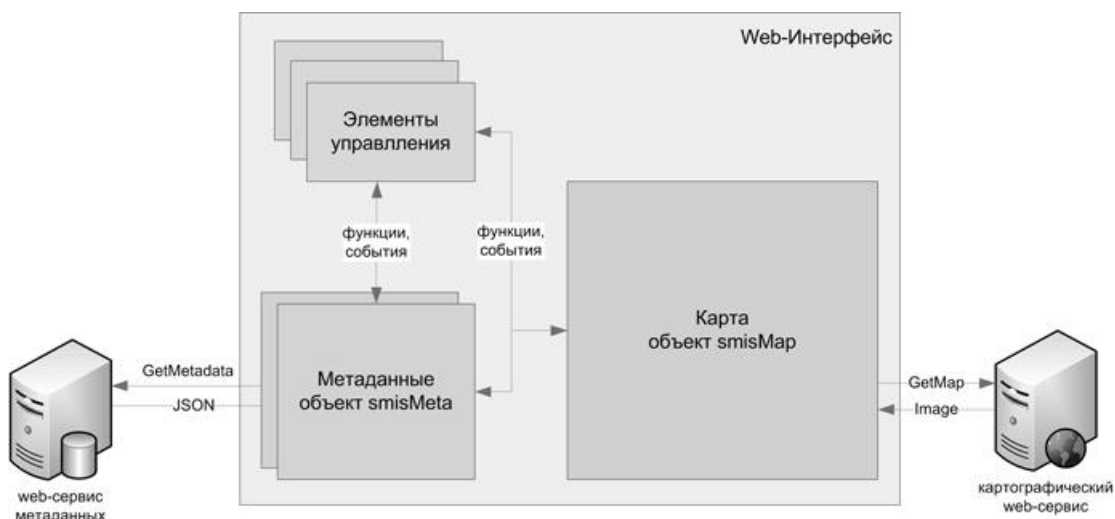


Рис.1. Основные элементы архитектуры веб-интерфейса

Методы анализа данных развиваются в направлении создания платформ, удобных для аналитики. Например, система SiSense (рис.2) [6,11], использует BI (Business Intelligence) – платформу, упрощающую аналитику сложных данных, предлагая полный цикл от подготовки данных до анализа и визуализации в веб с большими возможностями расширяемости и управления (1 поколение - RDBMS / OLAP, 2 поколение - In-Memory, 3 поколение - In- Chip).



Рис. 2. SiSense. Архитектура системы

Особенности SiSense (системы класса «всё в одном»):

- прямая загрузка данных из множества источников данных: SQL Server, Oracle, MySQL, PostgreSQL, MongoDB, Hadoop и многих других;
- своя собственная колоночная база данных Elasticube для сбора и консолидации данных с поддержкой SQL;
- большая скорость обработки данных обеспечивается за счет использования собственной запатентованной технологии In-Chip.
- визуальное проектирование базы данных;
- связь данных из разных источников в простом удобном визуальном интерфейсе;
- встроенный механизм загрузки изменений (инкрементальная загрузка);
- использование SQL запросов для сложных преобразований данных.

Одной из самых больших сложностей в разработке и развертывании решений BI является низкая скорость обработки больших массивов данных.

Решение данной задачи возможно за счет распараллеливания ресурсоемких операций на нескольких машинах (создание кластерных баз данных). Но создание компьютерного кластера часто считается излишеством для бизнес-аналитики и сильно увеличивает нагрузку на ИТ. С появлением 64-разрядных серверов, In-Memory технология стала важным шагом вперед к увеличению производительности по сравнению с традиционными системами управления реляционными базами данных. Но аналитика в оперативной памяти по-прежнему имеет существенные ограничения, когда речь заходит о стоимости серверов.

In-Chip технология использует все плюсы технологии In-Memory (In-Memory Computing – высокопроизводительные распределенные системы, предназначенные для хранения и обработки данных в оперативной памяти), но не требует мощных высокопроизводительных компьютерных кластеров, что достигается за счет эффективного использования (автоматически и в режиме реального времени) современных жестких дисков, оперативной памяти и центрального процессора, обеспечивая при этом такое же или даже лучшее быстродействие, чем In-Memory технология. Данные, хранящиеся на дисках, не пропадают при выключении сервера, в то время как данные, находящиеся в оперативной памяти, мгновенно теряются.

Перечисленные продукты нацелены на аналитику производственной деятельности.

Другой пример – платформа Physical Analytics Integrated Repository and Services (PAIRS, интегрированное хранилище и службы физической аналитики), предложенная Научно-исследовательским центром IBM – IBM TJ Watson Research Center, которая обеспечивает быстрое обнаружение данных путем автоматического обновления, объединения и гомогенизации слоев данных в пространстве и времени. Построенная на основе программного обеспечения с открытым исходным кодом big data PAIRS управляет автоматической загрузкой данных, преобразованием данных и масштабируемым хранилищем, одновременно являясь вычислительной платформой для запуска физических и статистических моделей на курируемых наборах данных [13]. Перед загрузкой данных на платформу, можно выполнять многослойные запросы и

фильтрацию в режиме реального времени. Кроме того, PAIRS предлагает основу для разработки пользовательской аналитики.

На рис.3 [13] показана схема извлечения данных с помощью «облачного» запроса к базе данных Hbase, размещенной в PAIRS для прогноза погоды и решения гидрологических задач. Hadoop — проект фонда Apache Software Foundation – свободно распространяемый набор утилит, библиотек и программный каркас для разработки и выполнения распределённых программ, работающих на кластерах из сотен и тысяч узлов. Используется для реализации поисковых и контекстных механизмов многих высоконагруженных веб-сайтов, Hbase - это распределенная, колоночно-ориентированная, мультиверсионная база типа «ключ-значение», разработанная в рамках проекта Hadoop.

Еще одно направление ускорения анализа данных – использование GPU (*graphics processing units*) – параллельных вычислений на графическом процессоре. Компания MapD Technologies объявила об открытии исходных текстов СУБД MapD Core, обеспечивающей создание хранилища в оперативной памяти (IMDB - *in-memory database*) и поддерживающая SQL и оптимизирована для решения задач по анализу и визуализации данных. Интерфейсы MapD позволяют пользователям добавлять или удалять слои данных [16,17].

В настоящее время в качестве основы современной аналитики данных big Earth используются кубы данных [8, 9] наряду с технологиями машинного обучения.

Недавно компания Planet объявила о своем намерении индексировать физические изменения на Земле и сделать эту информацию общедоступной с помощью бета-версии Planet Analytics, которая использует машинное обучение для преобразования ежедневных спутниковых снимков в информационные каналы, способные обнаруживать и классифицировать объекты, идентифицировать географические особенности и контролировать изменения на протяжении длительного промежутка времени [14].



Рис.3. Схема использования платформы PAIRS

Эти интеллектуальные информационные каналы интегрируются в существующие рабочие процессы и предоставляют клиентам информацию об интересующих их регионах.

Развитие аппаратуры для аэросъемки привело к появлению **беспилотных летательных аппаратов (БПЛА)**. Использование БПЛА для получения ДЗЗ предъявляет дополнительные требования к размещению, хранению и анализу данных. Одной из специфических задач является слияние измеряемых величин с картой в процессе съемки (геотегинг).

Поскольку съемка выполняется в крупных масштабах, объем получаемых данных также огромен. Например, БПЛА «Орлан» за один полет (8 часов) накапливает более 1 Тб данных (это объем суточной информации спутника MODIS).

Между тем, целью использования этой аппаратуры часто является фиксация координат и изображений только аномалий определенного типа (редких животных, чрезвычайных ситуаций и др.), не подразумевает длительного хранения других данных, и разработка «быстрых» алгоритмов распознавания могла бы существенно сократить потребности в хранении больших объемов информации.

Известны стандартные подходы к сокращению размерности для ускорения при анализе данных:

- отбор существенных признаков (с помощью алгоритма вычисления оценок);
- использование метода главных компонент,
- использование функции правдоподобия дифференцируемых классов;
- метод «ближайшего соседа» и др.

Перечисленные алгоритмы эффективны для конкретных частных задач, однако не существует общего подхода к решению указанных проблем. Для решения задачи ускорения анализа данных необходимы алгоритмы параллельного распознавания объектов по образцам для создания квазикластерной базы данных объектов. «Дешевый» способ получения информации приводит к увеличению количества неквалифицированных пользователей, заинтересованных в решении специфических задач, что требует разработки индивидуальных программ для анализа данных из-за недостатка стандартных инструментов.

Программы, применяемые в настоящее время для обработки данных с БПЛА, включают:

- программное обеспечение от компании Trimble (Германия) для фотограмметрической обработки данных, полученных с любых современных беспилотных летательных аппаратов (БПЛА): как с летательных аппаратов с неподвижным крылом, так и с БПЛА вертолетного типа, а также с аэростатов и других типов БПЛА;

- программное обеспечение от компании Harris Geospatial Solutions (бывшая Exelis VIS), подразделение Harris Corporation (США), для фотограмметрической обработки данных с БПЛА, в том числе мультиспектральных, и создания геопривязанных 2D и 3D продуктов;

- программные продукты Golden Software – мощные многофункциональные графические и картографические пакеты для моделирования и анализа данных, создания и визуализации трехмерных карт и моделей, графиков и диаграмм, трехмерного моделирования многокомпонентных данных и др. [19].

В лесном хозяйстве БПЛА успешно применяются для решения задач поиска и предотвращения мест незаконной вырубki леса, борьбы с пожарами, а также осуществления контроля и организации лесотехнических мероприятий [20].

Холдинг «Российские космические системы» (РКС, входит в Госкорпорацию «Роскосмос») продемонстрировал новейшие геоинформационные сервисы, созданные на основе данных с космических аппаратов орбитальной группировки дистанционного зондирования Земли (ДЗЗ), для решения проблем экологии, выявления незаконных вырубок, анализа состояния лесов и контроля лесопользования. Новые геоинформационные сервисы основаны на использовании данных ДЗЗ, продуктах их тематической обработки, а также специализированных приложениях для обеспечения отраслевой информацией лесохозяйственных организаций: контроля лесопользования и выявления изменений в лесном фонде — вырубок, пожаров, ветровалов, погибших насаждений.

Несмотря на решение, принятое совместно с «Роскосмосом» о 100 % охвате космическим мониторингом лесного фонда страны, в лесной отрасли все шире используются беспилотники. Так, WWF России передал дроны департаменту лесного хозяйства Приморского края и управлению лесами правительства Хабаровского края и организовал обучение специалистов лесной отрасли по управлению беспилотными летательными аппаратами; применению их для выявления незаконных рубок и проверки сигналов системы космического мониторинга "КЕДР". По сообщению регионального министерства природных ресурсов и экологии леса Калининградской области будут мониторить беспилотные летательные дроны; группой «Кронштадт» совместно с партнерами Segezha Group разработан проект по созданию геоинформационной системы на основе данных с беспилотников, которая позволит специалистам лесного сектора получать достоверную и актуальную информацию о сырьевых ресурсах с помощью авиа- фото- и космической съемки, а также определять перспективные направления строительства лесных дорог и поновому управлять лесозаготовительной техникой. Segezha Group оценивает новейшую разработку «Кронштадта» как масштабируемый облачный сервис с готовыми desktop-решениями для картографов и геопортальными решениями. На геопортал будут поступать данные с GPS-навигаторов и датчиков, установленных на лесовозах, форвардерах и харвестерах.

Следует отметить, что для проектов экономического развития российских территорий спутниковой лабораторией Геоспайдер создаются интернет ресурсы ГЕОПОРТАЛЫ - системы высокоточного геодезического позиционирования и беспилотной аэрофотосъемки с высоким оптическим разрешением, интегрированные с современными картографическими платформами и

WEB решениями. Из восьмидесяти пяти субъектов федерации ГИС-системы разработали и применяют в управлении своими территориями 36 или 42,3 % от общего числа. Но из этой группы только десять имеют более-менее полноценную ГИС-систему [21]. Геодезическая сеть спутниковых станций Геоспайдер является наземным сегментом ГЛОНАСС для формирования и передачи в своей зоне дифференциальных поправок. Сеть развернута и практически функционирует на территории Северо-Западного региона - Ленинградской, Псковской, Мурманской и Новгородской областях, частично Республики Карелия, а также Санкт-Петербурга и его пригородов. На платформе Геоспайдера и БПЛА Орлан получила развитие комплексная технология аэрофотосъемки земной поверхности с высоким оптическим разрешением для создания ГИС и электронных карт ОРЛАН-ГЕОСПАЙДЕР. В интересах управления земельными ресурсами для руководителей агрохозяйств, глав администраций муниципальных поселений и районов разработано специализированное ГИС-решение Геоинформационный мониторинг земельных ресурсов GEOMAP [22].

В 2019 году число проектов с использованием БПЛА продолжит рост, поэтому компании ищут способы автоматизировать и масштабировать свои операции для достижения большей эффективности. Индустрия движется к автономным дронам, удаленным операциям дронов и применения таких технологий, как «drone-in-a-box» (док-станции для дронов или ангары) и возможность подключения через 4G / LTE или 5G. Компания FlytBase ведет разработки в области интеллектуальной технологии автоматизации дронов и программного обеспечения, используя свою технологическую платформу Internet of Drones, удаленное подключение дронов BVLOS через Интернет, что позволяет операторам дистанционно управлять парком дронов и док-станций в режиме реального времени через облако.

В 2017 году были зарегистрированы ряд отечественных программ по созданию информационных и геоинформационных систем:

Геоинформационные и навигационные системы (GIS). Системы сбора, хранения, обработки, анализа, моделирования и визуализации массивов данных. Информационные системы для решения специфических отраслевых задач. Здесь и далее указаны сайты производителя: <http://www.geoleader.ru>.

Географическая информационная платформа «Атлас». Разработка Территориального фонда информации Республики Коми. <https://gis.rkomi.ru>.

Серверное и связующее программное обеспечение. Системы мониторинга и управления. Прикладное программное обеспечение общего назначения. Системы управления процессами организации. Системы сбора, хранения, обработки, анализа, моделирования и визуализации массивов данных, Библиотеки подпрограмм (SDK).

<http://www.newathena.ru/products/module/1800000000001865>

Распределенная блокчейн-платформа для хранения и обмена аккредитационных и KYC данных клиентов. (Средства обеспечения облачных и распределенных вычислений, средства виртуализации и системы хранения данных, серверное и связующее программное обеспечение).

<http://www.solarl.ru/projectb.html>

Средства обеспечения облачных и распределенных вычислений, средства виртуализации и системы хранения данных.

<http://www.raidix.ru/products/rasp/>

Серверное и связующее программное обеспечение. Средства обеспечения облачных и распределенных вычислений, средства виртуализации и системы хранения данных.

http://aerodisk.ru/upload/Aerodisk_A-CORE_Datasheet_v1.pdf

Системы мониторинга и управления. Системы сбора, хранения, обработки, анализа, моделирования и визуализации массивов данных. Системы управления процессами организации. Системы сбора, хранения, обработки, анализа, моделирования и визуализации массивов данных, информационные системы для решения специфических отраслевых задач.

http://www.rakurs.com/programmnoe_obespechenie/ScadaRSP

Системы сбора, хранения, обработки, анализа, моделирования и визуализации массивов данных. Информационные системы для решения специфических отраслевых задач. <http://www.o-code.ru/node/346>

Средства обеспечения облачных и распределенных вычислений, средства виртуализации и системы хранения данных. Информационные системы для решения специфических отраслевых задач. Системы сбора, хранения, обработки, анализа, моделирования и визуализации массивов данных. <http://baltinfocom.ru/BigData#block-6b43a4d0-1f35-4084-971b-6ba7d956fa9f>

Информационные системы для решения специфических отраслевых задач. Геоинформационные и навигационные системы (GIS).

Система управления базой геоданных. Система проверки картографических данных. Геоинформационные и навигационные системы (GIS). Поисковые системы. Системы управления процессами организации. Системы управления базами данных. Системы сбора, хранения, обработки, анализа, моделирования и визуализации массивов данных. Информационные системы для решения специфических отраслевых задач. Системы мониторинга и управления. <http://antereal.com/sw-cgis.html>.

Система экологического мониторинга. Геоинформационные и навигационные системы (GIS), поисковые системы, системы управления процессами организации, системы сбора, хранения, обработки, анализа, моделирования и визуализации массивов данных, информационные системы для решения специфических отраслевых задач, системы мониторинга и управления. <http://antereal.com/sw-sem.html>.

Программа ParmaGIS – программа для учёта лесных ресурсов. Геоинформационные и навигационные системы (GIS), прикладное программное обеспечение общего назначения, системы управления проектами, исследованиями, разработкой, проектированием и внедрением, системы сбора, хранения, обработки, анализа, моделирования и визуализации массивов данных, информационные системы для решения специфических отраслевых задач.

Большинство программ ориентировано на различные задачи, лишь последняя предназначена для нужд лесного хозяйства. Еще одно сообщение 2019 года о том, что научная группа института географии РАН провела в Кировской области тестирование нового метода оценки качества и количества лесных ресурсов, базирующегося на использовании космических снимков и алгоритмов машинного обучения с использованием big data и обширных материалов существующих баз лесотаксации [20].

Таким образом, задача организация хранения и доступа к дистанционным данным, возможности их консолидации, тем более анализа данных, тем более в реальном времени остается актуальной и требует значительных усилий как организационных, так и теоретических, и практических.

Библиографический список

1. Автоматизированная система поиска пространственных данных. <http://order.cgkipd.ru/>.
2. А.Б. Аквилонова и др. Работа информационной системы ИЦ ОМЗ с данными прибора MODIS космического аппарата TERRA. https://www.researchgate.net/publication/267940790_Rabota_informacionnoj_sistemy_NC_OMZ_s_dannymi_pribora_MODIS_kosmiceskogo_apparata_TERRA
3. Геопортал «Роскосмоса». <https://www.gptl.ru>
4. Б.А. Дворкин, М.А. Элердова. Особенности наземных сегментов современных космических систем ДЗЗ. ГЕОМАТИКА №3'2010, с.19-24.
5. И.П. Карачевцева, Е.Н. Матвеев, А.А. Коханов, Н.А. Козлова, А.С. Гаров. Разработка системы хранения планетных данных и организация доступа к ним на основе ГИС-технологий. Журнал ИКИ РАН «Современные проблемы дистанционного зондирования Земли из космоса». Т.10, №4, 2010. С.89-97
6. Современные системы бизнес-анализа SiSense. <http://sisense.com>
7. «Федеральный фонд пространственных данных». <http://calc.cgkipd.ru>
8. Г.Эринджипурат. Пространственно-временная аналитика геопространственных изображений. Форум «Интеграция геопространства». Анталия, Турция, апрель 18, 2018 г. 7 с. <https://medium.com/>.
9. Baumann, P., Mazzetti, P., Ungar, J., Barbera, R., Barboni, D., Beccati, A., and others. (2016). Big data analytics for earth sciences: The EarthServer approach. *International Journal of Digital Earth*, 1–27. <https://doi.org/10.1080/17538947.2014.1003106>.
10. Baumann, P. (2017). The Datacube Manifesto. Retrieved from EarthServer website: <http://earthserver.eu/tech/datacube-manifesto>.
11. C.Corbane et al. Big earth data analytics on Sentinel-1 and Landsat imagery in support to global human settlements mapping. *Big Earth Data*. Volume 1, 2017 - Issue 1-2. Published online: 30 Nov 2017.
12. Gorelick, N., Hancher, M., Dixon, M., Ilyushchenko, S., Thau, D., & Moore, R. (2017). Google Earth Engine: Planetary-scale geospatial analysis for everyone. *Remote Sensing of Environment*. doi:10.1016/j.rse.2017.06.031.

13. L.J. Klein and others. PAIRS: A scalable geo-spatial data analytics platform
14. Will Marshall. 2018:A Look Back and into the Future. January 21, 2019 www.planet.com/pulse/author/will/Marshall.
15. M.Sudmanns, S.Lang and D.Tiede. Big Earth Data From Data to Information. GI_Forum 2018, Issue.1. Page: 184 – 193.
16. (<https://www.mapd.com/blog/2017...>)
17. <https://www.opennet.ru/opennews/art.shtml?num=41333> GPU.
18. https://bespilotnik.org/info/articles/2018/bpla_v_lesnom_khozyaystve
19. <https://e.mail.ru/thread/0:1283766898000000085:6/www.sovzond.ru>
20. <https://www.sovzond.ru>
21. <http://geoportaly.ru/geoportals/geoportal-predsedatelya-komiteta-poupravleniyu-gosudarstvennym-imushchestvom-leningradskoy-oblasti/>
22. <https://www.geospider.ru>

Р.М. Яковлев, кандидат физ.-мат.наук, ведущий научн.сотр.,
СПб отделение Пагуошского движения
brig-av@yandex.ru;

И.А. Обухова, кандидат технических наук, доцент,
Кафедра информационных систем и технологий
СПбГЛТУ им. С.М.Кирова
IObukhova@inbox.ru

СИТУАЦИЯ С НЕРАСПРОСТРАНЕНИЕМ ЯДЕРНОГО ОРУЖИЯ И ПРОБЛЕМЫ РАЗВИТИЯ АТОМНОЙ ЭНЕРГЕТИКИ

Мы обсуждали в статьях настоящего сборника за 2017 и 2018 год оценки перспектив применения альтернативных моделей реакторов в ядерной энергетике и продолжаем оценки перспектив развития энергетике в настоящей статье. На сегодняшний день 187 государств являются участниками Договора о нераспространении ядерного оружия [1] (ДНЯО, 1968 г., вступил в действие в 1970 г.). ДНЯО - самый большой договор по количеству участников. Вне договора осталось только три страны: Израиль, Индия и Пакистан, в 2005 году из Договора вышел Северный Вьетнам. Согласно ДНЯО ядерными державами признаются государства осуществившие взрывы до 1 января 1967 г. Это – СССР (правопреемницей является только Россия), США, Великобритания, Франция, Китай. Чтобы не допустить переключения ядерной энергии с мирного применения на создание ядерного оружия, была создана специальная организация – Международное Агентство по Атомной Энергии - МАГАТЭ. ДНЯО является весьма важным документом. К государствам, нарушающим этот Договор, должны применяться международные санкции. Сейчас значительно расширяется география размещения атомных станций по всему миру, и АЭС строят даже в такой обеспеченной газом и нефтью солнечной стране, как Арабские Эмираты. Россия сейчас строит реакторы в Турции, Египте,

Индии, Вьетнаме. Расширение географии увеличивает опасность создания ядерного оружия и его применения террористами, несмотря на законы, запрещающие это.

В последние годы опасения, касающиеся ядерного терроризма, неоправданно отходят на задний план. Вместе с тем опасность, исходящая от этого рода деятельности, несколько не уменьшилась, а сильно возрастает, и обусловлено это в первую очередь тем, что плутоний в качестве МОКС топлива уже применяется при изготовлении ядерного топлива для различного типа реакторов, а география ядерной энергетики значительно расширяется. Поэтому для террористических организаций возможности приобрести основную составляющую атомного оружия – плутония, в последние годы значительно увеличились.

Мелкомасштабная переработка даже облученного ядерного топлива с целью выделения необходимого для ядерного устройства 8-10 кг реакторного плутония, хотя и не является простой и безопасной, но вполне реалистична и может дать необходимый продукт [2,3].

Несравнимо проще, без тяжелого защитного оборудования и дистанционных манипуляторов, может быть извлечён плутоний из смешанного, необлученного МОКС-топлива [4]. В состав этого топлива может входить как реакторный плутоний, так и оружейный (запасы которого принято сокращать) из демонтированных боеголовок. Реакторный плутоний не соответствует оружейным стандартам, но из него также можно изготовить заряд для ядерных взрывных устройств. Мощность такого заряда может быть меньше, чем у стандартных, но они будут обладать всеми поражающими факторами, характерными для ядерного оружия. Это подтверждается испытанием подобных взрывных устройств в 1962 г. в США и в 1974 г. в Индии, а также последующих в Индии и Пакистане (подземные испытания продолжались до 1998 года).

На фоне сравнительно простого извлечения плутония, пригодного для «грязной» бомбы, обогащение урана до нужной кондиции представляется исключительно сложным делом, требующим огромных средств и крупных масштабов производства. Для наработки 20 кг 90% U-235, необходимого для изготовления ядерного заряда, требуется превратить в газовую фазу (UF_6) 4000 кг природного урана и затем разделить изотопы на центрифугах или методом диффузии или с использованием лазеров [5]. Сделать это подпольно и не завестись совершенно невозможно. Проще, безусловно, добыть уран высокообогащения, предназначенный для исследовательских реакторов. Но контроль таких реакторов взят под строгий контроль и сейчас предпринимаются меры по замене высокообогащенного урана для исследовательских реакторов на уран низкого обогащения. Поэтому далее мы будем говорить только о плутонии.

Сейчас наработано 2500 тонн «грязного» плутония, а 250 тонн уже извлечено и хранится на комбинатах 8 государств (более подробно об этом в [6,7]). Часть извлекаемого плутония используется в качестве смешанного уран-плутониевого топлива (далее, МОКС-топливо) на небольшом количест-

ве реакторов во Франции, Англии, Канаде, Индии. Предполагается широко использовать МОКС-топливо в России, для чего под Красноярском в Железногорске построен и уже запущен большой комплекс для получения МОКС-топлива из уже использованного облученного топлива для реакторов различного типа с разными уровнями содержания в ядерном топливе плутония (4-5% для реакторов на тепловых нейтронах и до 20% для реакторов на быстрых нейтронах). В целевой правительственной программе интенсивного развития ядерной энергетики России [8] с исчерпанием запасов природного урана, в котором делящегося урана-235 всего 0,7%, предусматривается сделать МОКС-топливо основным ядерным топливом реакторов (см. рис.1).

Стратегические этапы развития АЭ, ГВт 2030-2050

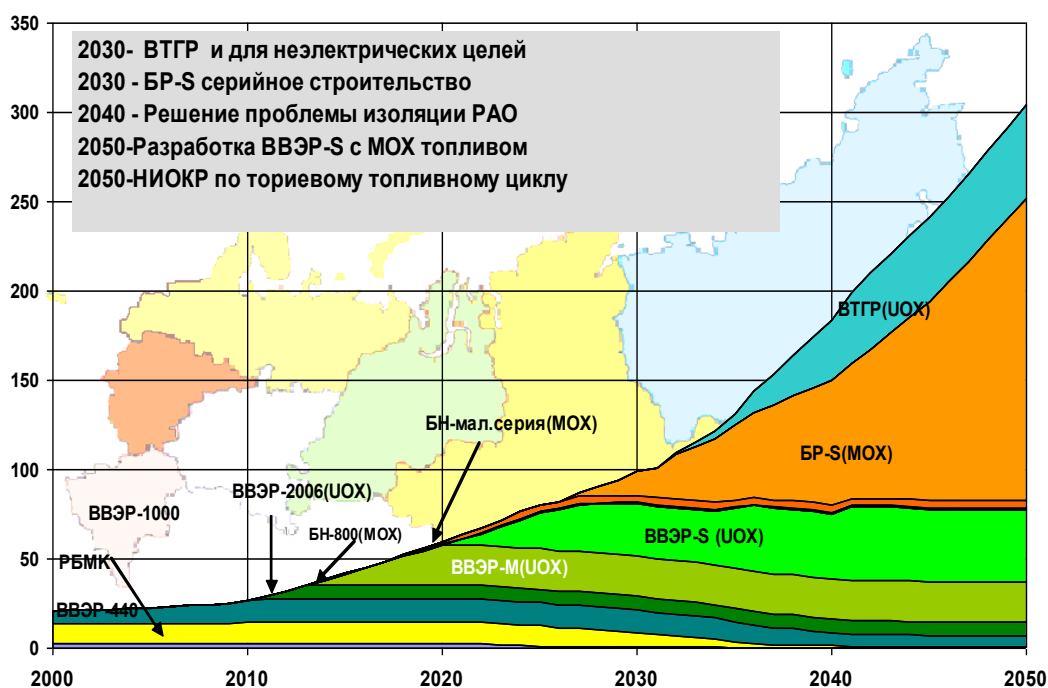


Рис. 1. Базовый сценарий структуры генерирующих мощностей атомной энергетики России до 2050 г

Как видно из рисунка уже после 2030 года появится слишком много объектов, в которых обращается топливо, содержащее сравнительно легко извлекаемый плутоний. Это предприятия, на которых перерабатываются ОЯТ и извлекается из топлива плутоний, заводы на которых изготавливается МОКС-топливо, промежуточные хранилища, сами реакторы, в которых используется МОКС-топливо, транспортные средства для перемещения этого топлива, иногда на очень большие расстояния (транспортировка морем, переработка ОЯТ и изготовление ТВЭЛов из МОКС-топлива для Японии производилось на предприятии БНФЛ в Англии, Селлафилд). **Недоучет, потеря, замена топлива содержащего плутоний возможны на всех этапах перемещения топлива. Следовательно, хищения вполне возможны.**

Тревожная ситуация с дополнительной возможностью распространения складывается в связи с выполнением обязательств по Соглашению о сокращении запасов оружейного плутония. Должно быть утилизировано по 34 тонны оружейного плутония каждой стороной за 20 лет. Россия обязалась перевести все 34 тонны в МОКС-топливо и сжечь его в реакторах. Для США необходимо сжечь в реакторах 25 тонн, а 9 тонн перевести в состояние, из которого очень трудно извлечь плутоний и эту часть захоронить. Предлагаемое дожигание военного плутония на энергетических реакторах приведёт, как и в случае с реакторным плутонием, к распространению теперь уже оружейного плутония. Таким образом, 59 тонн плутония с высоким обогащением как в процессе изготовления МОКС-топлива, так и при дожигании на реакторах, оказываются доступными для хищения в значительно большей степени, чем в процессе иммобилизации, выполняемым в одном месте, что и предполагают сделать в США.

Такая серьезная проблема с реальным распространением основной составляющей ядерного оружия - плутония возникла уже в существующей атомной энергетике с 450 энергетическими реакторами, из которых всего только единственный реактор на быстрых нейтронах БН-800 работает в России, а также с несколькими предприятиями по переработке и изготовлению ядерного топлива. Хорошо, что наиболее крупные из них находятся в Англии, Франции и России, в странах, принадлежащих к ядерному клубу и принимающими условия контроля со стороны МАГАТЭ.

А теперь посмотрим, что нас и потомков ожидало бы при реализации планов наших атомных менеджеров и стратегов в недалёком будущем. В соответствии со стратегическим планом развития атомной энергетики РФ к середине столетия, предполагаемое количество реакторов мощностью 1 ГВт (эл.) будет доведено в России до 300. Как видно из *рисунка 1* к 2050 году, как предполагают наши стратеги безальтернативной атомной энергетики, более половины реакторов будут реакторы на **быстрых нейтронах**, работающие в замкнутом цикле. При замыкании ядерного топливного цикла каждый год из реактора на быстрых нейтронах мощностью 1 ГВт (эл.) за год будет извлекаться 2000 кг плутония, находящегося в 14 тоннах ОЯТ и догружаться переработанное тут же при АЭС топливо с таким же количеством плутония. При работе одного быстрого реактора в нем самом, в зоне выдержки топлива, на заводе его переработки с целью извлечения плутония и изготовления новых ТВЭЛов будет находиться 20 тонн плутония.. **Поскольку МОКС-топливо быстрых реакторов содержит 15-20 процентов плутония, а не 3-4 процента, как в реакторах на тепловых нейтронах, то оно значительно привлекательнее для хищения.**

Но этого ещё недостаточно для энергетики с реакторами на быстрых нейтронах. 20 тонн плутония, необходимые для работы одного реактора, предполагается получить из отработанного топлива реакторов на тепловых нейтронах, переработав 2500 тонн ОЯТ, для чего дополнительно необходим отдельный комбинат для получения этого плутония из ОЯТ тепловых реак-

торов. Это означает следующее: без перевозки топлива содержащего плутоний не обойтись, что увеличивает возможности хищения.

Нужно ещё учесть следующее. Реакторы на быстрых нейтронах были специально созданы для наработки оружейного плутония. Поэтому **в зоне воспроизводства «мирных» быстрых реакторов возможна весьма эффективная наработка плутония оружейного качества (обогащение по изотопу ^{239}Pu 93,5 %) с его отдельным выделением.** Для замыкания ядерного цикла с вовлечением всего урана зона воспроизводства совершенно необходима. Если обходиться без неё, то состав плутония в активной зоне реактора после второй загрузки уже не будет годен, поскольку в нем возникают неделящиеся чётные изотопы плутония. Его надо обогащать или оружейным из запасов, или из зоны воспроизводства, где располагается обедненный уран-238, в котором будет образовываться плутоний-239, который отдельно из нескольких тонн урана-238 следует извлечь, добавляя в топливо из активной зоны. Т.е. без плутония оружейного или близкого ему по составу **плутония** из зоны воспроизводства не обойтись. А потому и контроль за таким плутонием должен быть очень жестким. Чтобы не уходил такой плутоний на изготовление оружия, за всеми реакторами на быстрых нейтронах необходимо установить строгий международный контроль. Но, в действительности, вряд ли удастся добиться желаемого. Пример тому – Индия, правительство которой наотрез отказывается принять контроль МАГАТЭ над построенными с помощью Америки, Франции и Англии «исследовательскими» реакторами [9] и имеет реальную возможность получать на них замечательный оружейный плутоний, объясняя остальным, что ей это не надо.

Нам очень не хотелось бы предполагать возможного обострения отношений с Китаем, но передавать документацию и строить для них реакторы БН-800, на которых может производиться в год 800 кг плутония (часть из которого может быть плутонием высокого обогащения при использовании зоны воспроизводства) крайне неосмотрительно и опасно, хотя может быть и выгодно нашим топ-менеджерам при высокой оплате их услуг.

Перспектива развития атомной энергетики на плутониевом топливе кажется нам весьма удручающей и пугающей. Удручает такая картина потому, что по этому сценарию, который рассматривается сейчас высокими авторитетами в России, как единственно правильный, создающий для человечества энергетическую безопасность на века, вся энергетика станет плутониевой, и Земля под конец века будет покрыта несколькими тысячами плутониевых реакторов на быстрых нейтронах с фабриками переработки на каждом из них больших количеств ядерного топлива, содержащего плутоний. Это безрадостно. А пугает сценарий, прописанный в планах дальнего развития энергетики, тем, что мирное использование плутония гарантируется только циркулярами и инструкциями, которые злоумышленники могут игнорировать. И не только злоумышленники, но и целые государства, если у их правителей воинственный менталитет.

Выход из этой ситуации нам представляется только при отказе от уран-плутониевой энергетики и переходе к торий-урановой в варианте реакторов без

ТВЭЛов [6,7]. Авторами было предложено несколько вариантов уран-ториевых реакторов с расплавами фторидов и с суспензией топлива в расплаве свинца. [10,11] Как первый этап работы нами предлагалось создать «действующую» уран-ториевую реакторную установку на расплаве солей в компьютерном варианте, основанным на «пробирочных» измерениях по образованию различных элементов в жидкосолевой композиции и поведению их в ней, а также на совместимости композиции с различными материалами, используя нерадиоактивные вещества, только меченые радиоактивными атомами. Именно в Радиовом институте с его радиохимиками высокой квалификации эту наиболее ответственную и наиболее важную часть работы предполагалось выполнить за 2-3 года, исследовав также стойкость и процессы разрушения различных конструкционных материалов в солевом расплаве.

Новые реакторные установки без плутония возникнут не сразу. А сейчас, чтобы не было «случайного исчезновения» не только оружейного, но и реакторного плутония, а также материалов, содержащих плутоний, необходимо установить очень строгий международный контроль за всеми предприятиями переработки и хранилищами облученного топлива и извлеченного плутония. Следует также ограничить число стран, в которых может перерабатываться ОЯТ, а МОКС топливо лучше исключить из оборота, учитывая его сверх высокую опасность, на которую мы пытались подробнее донести в наших работах и на всех многочисленных пленарных сообщениях в Доме Ученых, на семинарах проф.М.Б. Игнатьева, руководи-теля С-Пб отделением Пагуошского движения по нераспространению ядерного оружия.

Мы обсудили проблему нераспространения плутония - основной составляющей ядерного оружия, которая неразрешима в атомной энергетике, основанной на производстве и воспроизводстве плутония. АЭС с хранилищами топлива, системой переработки облученного ядерного топлива, в которую входит извлечение и переработка плутония, несёт риск несанкционированного распространения ядерной взрывчатки. Нынешний международный режим гарантий не соответствует проблемам безопасности расширенного развития атомной энергетике, как это предусмотрено сценариями глобального развития.

Кроме этой проблемы в уран-плутониевой энергетике с твердым топливом существуют ещё серьезные проблемы, попробуем кратко обсудить их, чтобы попытаться понять тенденции развития мировой энергетике.

Итак, вот эти проблемы:

-1. Исключительно высокий уровень опасности, обусловленный сосредоточенной в активной зоне потенциальной энергии, которая может в критических обстоятельствах привести к катастрофам типа Чернобыльской или Фукусимской.

Чернобыль для существующих реакторов может повториться. Атомщики всячески замалчивают уровень опасности ядерных реакторов и сильно преуменьшают последствия катастрофы, сводя катастрофу в Чернобыле к гибели 42 человек и лучевой болезни 134. От курения и автомобильных аварий погибает больше. Чтобы всё было в порядке на этих замечательных устройствах, надо правильно нажимать кнопки.

Стараются не вспоминать о 600 тысячах облученных ликвидаторов и о 350 тысячах жителей Белоруссии, Украины и России, которые были эвакуированы из родной Земли, но многие остались на сильно загрязнённых радиоактивностью территориях. На ослабление последствий Чернобыльской катастрофы к 2006 году уже было израсходовано в трёх республиках не менее 500 млрд долларов и расходы продолжаются. Более подробно мы об этом вспоминали в нашей предыдущей работе для этого сборника и в работе, опубликованной в журнале «Биосфера». Исследования последствий Чернобыльской катастрофы для жителей и природы Белоруссии опубликованы в работе под руководством чл.-кор. АН России Яблокова и чл.-кор. АН Белоруссии В.Б. Нестеренко и А.В. Нестеренко [14]. В Чернобыльской катастрофе пострадали сотни тысяч жителей трех республик, при этом, многие из них – дети. А ликвидаторы, получившие высокие дозы в первые смены работы в Чернобыльской Зоне – почти все инвалиды. На службе у Атома очень ученые медики свидетельствуют, что инвалидность и болезни у ликвидаторов – от страха, напуганы разговорами о вреде радиации, болезнь даже такую для облученных инвалидов придумали – *радиофобия*. На это более независимые, а может быть просто честные ученые-биологи и медики, обнаружив существенные повреждения клеток при облучении дозами меньшими, чем у ликвидаторов, задают вопрос первым очень ученым – может быть у клеток тоже радиофобия?

Итак, чтобы не стать зомби, согласившись с атомными топ-менеджерам и оплачиваемыми их «учёными», что **Чернобыльская Катастрофа** – ерунда, а автомобили и сигареты хуже и страшнее потерявшего управление реактора и с тем, что ранние смерти и тяжелые болезни ликвидаторов от их трусости (испугались вреда какой-то радиации) давайте признаем, что это была **Огромная Планетарная Катастрофа**, особенно ужасная для единой тогда для русских, украинцев и белорусов стране нашей. Она стала одной из причин развала большой страны, уж слишком много стало пострадавших, и слишком много вранья и темноты вокруг. Очень велики были и финансовые и материальные потери.

Трагедия великая была для тех, кто покидал навсегда родной дом и край. Чтобы чтить память погибших и помнить о пострадавших, приводим фотографии о случившемся. Это было и всегда есть.



Рис.2.Заброшенные живые блоки Припяти, с уцелевшим деревом



Рис.3.Чернобыльская АЭС в 2003 году с саркофагом

Чтобы правильно понимать и учитывать возможные последствия аварии на атомной станции с разрушением реактора (7-й уровень по междуна-

родной шкале INES) мы приведём здесь суммы тех реальных (у Лукашенко они не могут быть иными) государственных расходов на уменьшение последствий Чернобыльской катастрофы в Белоруссии с 1986 по 2015 год (табл.1).

Т а б л и ц а 1

Sectoral composition of the socio-economic damage incurred by Belarus from the Chernobyl NPP disaster, billions of USD

Sectors of economy	1986-1990	1991-1995	1996-2000	2001-2015	1986-2015
Health care	4,05	16,77	18,13	54,32	93,27
Agriculture	18,3	20,0	15,6	18,1	72,0
Forestry	0,58	0,68	0,70	2,15	4,11
Industry	0,06	0,13	0,11	0,33	0,63
Construction sector	0,15	1,25	0,32	0,96	2,68
Raw materials and water	2,00	0,12	0,15	0,40	2,67
Transportation and communications	0,93	1,20	0,36	0,90	3,39
Social sphere	2,84	5,45	2,96	6,45	17,70
Deactivation of contaminated areas	0,04	4,19	22,48	10,12	36,83
Radioecological monitoring	0,05	0,21	0,19	1,27	1,72
Total	29,00	50,00	61,00	95,00	235,00

National Program on Mitigation of Consequences of the Chernobyl NPP Catastrophe

Отчёт Комитета по проблемам катастрофы на Чернобыльской АЭС при Совете Министров (Comchernobyl), 2015, Минск.

Не меньшими были расходы на уменьшение последствий аварии для Украины и России [15].

После аварий на Тримайл Айленд, в Чернобыле и в Японии на АЭС Фукусима-1 были приняты дополнительные меры по повышению безопасности АЭС во всём мире. Конструкция **современного реактора** позволяет достичь низкого уровня серьёзной аварии **при его нормальной эксплуатации**. В ТВЭЛах при работе реактора накапливается очень высокая радиоактивности и велико их остаточное тепловыделение, поэтому в активной зоне любого реактора при прекращении съёма тепла неминуемо произойдет расплав топлива (2800⁰С) и значительная часть радиоактивных продуктов вылетит, несмотря на жертвенные материалы в ловушке.

Возможность большой аварии на любом из реакторов при разрушении системы теплообмена в активной зоне реактора с неминуемым расплавом топлива отмечает один из ведущих создателей ядерного оружия Л.П.Феоктистов [16]. Кроме этой возможности он отмечает ещё одну воз-

возможность катастрофы: «Ни один из ныне существующих реакторов, работающих по принципу выгорания, нельзя отнести к безусловно безопасным, потому что, если вдруг по случайным причинам регулирующие стержни покинут активную зону, возникнет значительная надкритичность. Цепная реакция в таких условиях будет развиваться настолько быстро, что никакая аварийная защита не поможет. **Подводя краткий итог, рискну предположить, что в освоении ядерной энергии мы не в конце пути, как многим казалось до Чернобыльской трагедии, а где-то в самом начале.»**

Против диверсии или теракта защиты нет, точно также как нет защиты от грубых ошибок или предумышленно неправильных действий персонала. Нет защиты от падения большого самолёта или после начала военного конфликта при ударе по атомной станции или по объекту, где хранится облученное ядерное топливо.

Вероятность большой аварии на АЭС не исчезающе мала, как пытались нас перед самым Чернобылем уверить корифеи атомной науки во главе с Президентом Академии Наук СССР, и сейчас повторяют эти уверения создатели Программы атомного *ренессанса*, а эта вероятность всё время только увеличивается в эпоху развитого терроризма и в связи с расширением географии атомной энергетики, даже в таких странах, где много солнца и нефти (Иран, Саудовская Аравия, Вьетнам). Длинный список аварий и инцидентов приведен на сайте Википедии, с указанием причин происшедших аварий [17]

-2. Отходы. *Весьма важная проблема атомной энергетики. Решение её весьма актуально, поскольку накопилось их слишком много и что с ними делать не ясно.* Геологическое размещение отходов технически возможно, однако, никакой гарантии консервации топлива на десятки тысяч лет и более не существует. Ядерное наследие было создано в СССР в основном при получении ядерного оружия. Эта проблема является очень важной для всех стран и, особенно для России, где их больше всего. При переработке облученного ядерного топлива (ОЯТ) с целью получения плутония получается огромное количество радиоактивных отходов [18]. **Переработка всего одной тонны ОЯТ приводит к образованию дополнительных 2200 т жидких радиоактивных отходов (ЖРО), имеющих суммарную активность около 600 тыс. Ки, в том числе: 45 куб.м высокоактивных ЖРО с активностью до 10 Ки/л, 150 куб.м среднеактивных ЖРО с активностью до 1 Ки/л, 2000 куб.м низкоактивных ЖРО с активностью до 10^{-5} Ки/л.** Проблема ядерных отходов накопившихся в огромных количествах при создании ядерного оружия и проблема ликвидацией ядерного наследия является весьма важной /19/ и для их решения были созданы федеральные целевые программы ФЦП ЯРБ-1 на период с 2008 г до 2015 и затем ФЦП ЯРБ-2. Согласно [20], по ФЦП ЯРБ-1 было переработано 83,9 т ОЯТ, а по ФЦП ЯРБ-2 предполагается переработать 3 081,4 т, всего получается 3 165,3 т. При переработке этого количества ОЯТ получится при выполнении программ:

- 142 тыс. куб.м высокоактивных ЖРО с суммарной активностью до 1420 млн. Ки;

- 475 тыс. куб.м среднеактивных ЖРО с суммарной активностью до 475 млн. Ки;

- 6,3 млн. куб.м низкоактивных ЖРО с суммарной активностью до 6,3 тыс. Ки.

Образующиеся РАО относятся к «удаляемым» отходам и должны быть захоронены в соответствии принятыми требованиями. Включение затрат на переработку ОЯТ в ФЦП ЯРБ-1 и ЯРБ-2 приводит к тому, **что при выполнении этих программ образуется новое радиоактивное наследие в виде удаляемых радиоактивных отходов.**

Остеклованные высокоактивные ЖРО и цементированные среднеактивные ЖРО подлежат захоронению, что создает реальную опасность для будущих поколений. Дело в том, что в России принята такая программа по захоронению радиоактивных отходов (РАО), которая, можно сказать, прямо направлена на создание радиационной опасности для будущих поколений.

В США пришли к осознанию, что невозможно гарантировать безопасность пунктов захоронения РАО на десятки и сотни тысяч лет. Можно только выполнить прогнозную оценку их безопасности и по результатам оценки принимать решения. Оценка безопасности, можно сказать, превратилась в отдельную отрасль науки и стала краеугольным камнем в обеспечении безопасности будущих поколений при захоронении РАО. Именно на широкомасштабные экспериментальные и теоретические исследования методов и параметров оценки безопасности в США были истрочены многие миллиарды долларов. Полученные в США результаты и методические подходы приняли все страны и международные организации, кроме России [21].

Вот такие проблемы уже возникли при работе в России всего 23 реакторов. А как они увеличатся при замыкании ядерного цикла с ОЯТ от 300 реакторов после 2050 года, которые мечтаются установить в соответствии с целевой программой развития энергетики в России? Слава Богу, (здесь и коррупция тоже неожиданно помогает) сроки запуска запланированных блоков год за годом отодвигаются, и стоимость их создания становится непомерно высокой. Таким образом, целевая безальтернативная ядерная программа спасения России и всего человечества от придуманного потепления и всемирного потопа срывается. Несостоятельной оказалась теория, оправдывающая сооружение новых и новых АЭС, – это теория потепления от производимого при сжигании углеводородов избытка углекислого газа. С глубокой и обоснованной критикой придуманной теории потепления давно выступают наши известные ученые [22-24]. Мы живем в эпоху, сравнительно короткую, временного потепления не за счёт избытка углекислого газа, а меняющейся в определенные периоды активности Солнца и расположения от него Земли.

В мировом Океане CO_2 в 100 раз больше, чем в атмосфере и увеличение углекислого газа в воздухе это результат временного потепления океанской воды с уменьшением растворенного в ней углекислого газа. Временное потепление должно закончиться через 30-40 лет и начнется для Земли новый ледниковый период. Из приведенного нами из работы академика РАН Р.И. Нигматуллина [22] рис.3 видно, что Земля находится сейчас перед эпохой гло-

бального и долгого похолодания, как 110 тысяч лет назад, и не Потопа Великого следует ожидать, а понижения уровня мирового океана в связи с наступлением очередного ледникового периода.

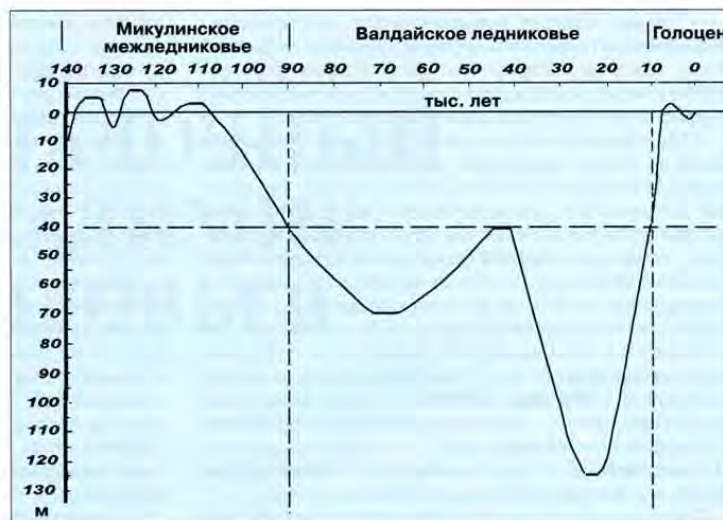


Рис.3. Изменение уровня Мирового океана за 140 тыс. лет

Если всё так, и определяет климат на Земле, как и в древние времена Солнце, которое через некоторое время сокращает свою активность, то Всемирного Потопа не следует ожидать при сжигании людьми чего и сколько угодно. Тогда и Киотский протокол, и очень-очень много атомных станций становятся ненужными. Тогда не пострадает человечество от избытка плутония и непомерного количества ядерных отходов! И тогда, может быть, как уже почти во всех странах, и в России возьмутся по-настоящему использовать самые большие на Земле ресурсы ВОЗОБНОВЛЯЕМОЙ ЭНЕРГИИ?

(Посмотрите работы известных учёных по предстоящему глобальному похолоданию в представленном списке работ [22-23], они есть в интернете)

Есть еще две очень важные проблемы, которые являются наследием прошлого, но которые необходимо решить, а для этого выделить значительное финансирование.

4. Вывод из эксплуатации АЭС. В 2005 году Федеральным агентством по атомной энергии была создана концепция вывода [24]. Для этого нужны значительные средства. Для разных стран они составили от 0.3 до 1 млрд. евро на 1 ГВт Эл выводимой из эксплуатации станции. Для вывода из эксплуатации РБМК в Литве и перемещения ОЯТ в сухое хранилище Евросоюз выделил 1.5 млрд. евро.

В ближайшие 15 лет предстоит масштабная задача по подготовке и выводу из эксплуатации энергоблоков АЭС 1-го поколения [25,26]. В период с 2016 по 2020 гг. должны быть выведены из эксплуатации 8 энергоблоков (3 и 4 блоки Нововоронежской АЭС, первые и вторые блоки Кольской, Билибинской и Ленинградской станций). В России (СССР) достаточное распространение получили энергетические уран-графитовые реакторы, разработанные на базе промышленных реакторов для наработки оружейного плутония. В настоящее

время действуют 12, в том числе 3 энергоблока 1-го поколения, введенные в эксплуатацию в 1973-1976 гг., Масса графитовой кладки реактора РБМК-1000 составляет 1700 т. Активность графитовой кладки реактора определяется долгоживущим изотопом ^{14}C с периодом полураспада 5400 лет (95 % всей активности графита). В настоящее время отсутствуют технические решения и промышленные технологии по кондиционированию радиоактивно загрязненного графита до стадии захоронения. В связи с этим в качестве базового варианта вывода из эксплуатации реактора РБМК-1000 принят вариант длительного (после выгрузки ОЯТ) хранения.

Сооружение хранилищ и могильников РАО разной активности, контейнеров для перевозки, технических средств для кондиционирования РАО, разделения образующихся отходов – без решения этих вопросов невозможно создать технологию работ, а следовательно, и целостную концепцию. [24,27]. Количество РАО при выводе АЭС из эксплуатации значительно возрастает и самым серьезным образом отразится на общей ситуации с РАО. Создание единой эффективной системы обращения с РАО является основной задачей при выводе АЭС из эксплуатации. Наиболее сложно обстоит ситуация с твердыми радиоактивными отходами при выводе из эксплуатации реакторов РБМК-1000 [25]. При демонтаже одного реактора РБМК-1000 количество образующихся отходов, подлежащих захоронению, составляет около 100 тыс. т бетона и 10 тыс. т стали суммарной активностью 10^5 ТБк (2,8 млн. Ки). Помимо металлических ТРО и отходов строительных конструкций, необходимо утилизировать 1700 т радиоактивного графита, технологии переработки которого в мире не существует.

Стоимость вывода из эксплуатации поэтому одного блока РБМК, вероятно, будет не ниже блока в Литве. Мы остановились на снятии блока одного типа – РБМК-1000, поскольку на Ленинградской АЭС предстоит вывести из эксплуатации 4 блока. Стоимость сухого хранилища для 4 блоков тоже будет не малой. Более подробно с проблемой вывода из эксплуатации атомных станций можно ознакомиться в работах [24-27].

3. Реабилитация территорий (ликвидация последствий производства оружейного плутония). Этому уделяется большое внимание на Западе, и выделяются для реабилитации огромные средства. В Англии выделено 64 млрд. фунтов, в США уже затрачено 30 млрд. долларов и дополнительно выделено ещё 320 млрд. долл. В России ситуация весьма сложная. Загрязненными являются огромные территории в Сибири, на Урале, Новой Земле, в Архангельской обл., на Кольском полуострове. Средств для ликвидаций последствий производства ядерного оружия требуется не меньше, чем запланировано в США, где уже произведены первые затраты. С выполненными в этом направлении работами можно ознакомиться из представленного списка [25,26].

5. Стоимость энергии, произведённой на новых атомных станциях. На свободном рынке стоимость, **неконкурентоспособна по сравнению с энергией, произведённой из угля и природного газа**, если добавляются расходы на обеспечение дополнительной безопасности (ловушки расплавленного топлива), а также расходы на снятие АЭС с эксплуатации и утилизацию топлива.

Очень велика также стоимость ежегодной замены трети ТВЭЛов в реакторе (25 тонн топлива). За 40 лет – 1000 тонн ТВЭЛов, и это стоит дорого. Реакторы же на быстрых нейтронах с заводами по переработке ОЯТ при них будут стоить значительно дороже существующих.

Приводим ниже таблицу себестоимостей получения электроэнергии на разных носителях для Англии [28]:

Т а б л и ц а 2

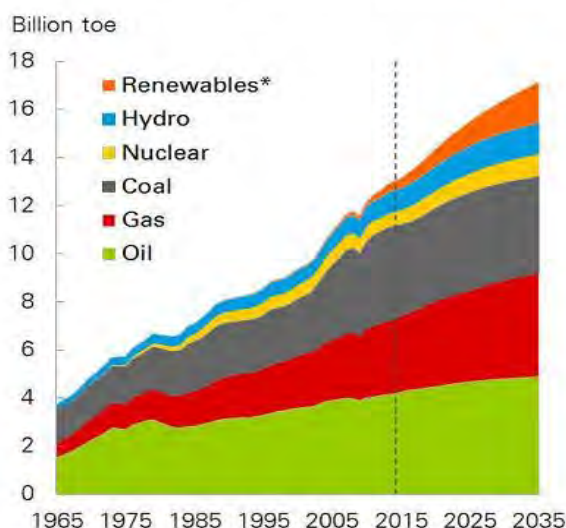
Данные себестоимости электроэнергии в Англии, 2015 г

Вид энергетики	Доля в Англии, средние по стоимости, фунт/МВт час
Ядерная	93
Солнечная	80
Ветровая	62-102
Газ	66
Биомасса	87

В таблице для АЭС приводятся стоимости без учёта утилизации радиоактивных отходов и вывода АЭС из эксплуатации и последующей реабилитации территории. С учётом этих позиций производства электроэнергии следует увеличить, по крайней мере, на 30 процентов.

Солнечных батарей, гидроэлектростанций и усовершенствованные, более экологически чистые, с фильтрами, угольные ТЭС являются приоритетными сейчас в Китае, США, Норвегии и почти везде, кроме России. И это неправильно. Доля возобновляемых источников энергии во всем мире неуклонно возрастает, как показано на следующем рис.4, а доля атомной энергетики не растёт.

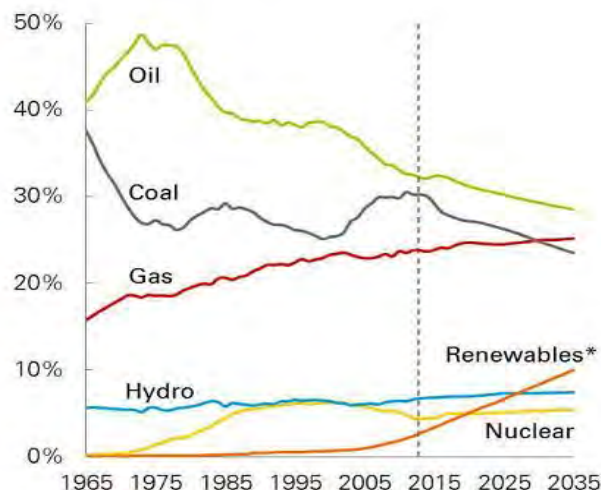
Primary energy consumption by fuel



*Renewables includes wind, solar, geothermal, biomass, and biofuels.

2017 Energy Outlook

Shares of primary energy



© BP p.l.c. 2017

Рис.4. Прогнозы развития энергетики и доля различных источников энергии

Как видно из рисунка доля атомной энергетики не увеличивается, но возрастает доля возобновляемых источников энергии. Эта оценка близка к прогнозу, представленному в работе [31]. В нашей предыдущей статье, опуб-

ликованной в сборнике трудов по информатике в 2018 году, мы привели оценку проф. Б.И.Нигматуллина [29] перспектив развития различных источников электроэнергии до 2050 г. Приведенная им оценка прогноза развития энергетики практически совпадает с оценкой из Китая [30]. В базисном варианте, как следует из этой статьи, мощность всех АЭС во всём мире увеличится от 383 в 2015 году до 456 ГВт в 2050 г., и составит 5% в 2040 году и 6,6% в 2050 году от всей энергетики, при этом, как это следует из анализа китайских ученых [31], количество станций Китая увеличится от 41 до 243. Отсюда следует, что если исключить Китай, во всем остальном мире мощность АЭС в 2015г. была $383-41=342$ ГВт, а в 2050 она будет всего $456-246=210$ ГВт, т.е. **мощность всех АЭС без Китая уменьшится на 38%** (это с учетом закрытия выработавших свой 60 летний срок АЭС). **Вот такой «ренессанс» атомной энергетики в мире происходит!**

А в России несмотря на:

1. Проблему нераспространения ядерного оружия,
 2. Дороговизну (больше 7 млрд долл за ГВт установленной мощности АЭС),
 3. Исключительно высокий уровень опасности при крупной аварии с разрушением активной зоны реактора,
 4. Проблему с накоплением огромного количества весьма опасных радиоактивных продуктов и ещё двух перечисленных выше проблем
- СОГЛАСНО СТРАТЕГИЧЕСКОМУ БЕЗЪАЛЬТЕРНАТИВНОМУ ПЛАНУ (см. рис.1) МОЩНОСТИ АТОМНЫХ СТАНЦИЙ РОССИИ ВОЗРАСТУТ ОТ 22 ДО 300 МВт- почти в 15 раз!!! То есть почти всю энергетику предполагается сделать ядерной –несмотря ни на что. Это тихий ужас, а в случае повтора Чернобыля – конец стране.

Об альтернативных источниках энергии в России

Российские политики, которые хотят быть в мировом тренде и любят говорить о развитии ВИЭ в нашей стране, лукавят, говоря о 17-20 % альтернативной энергетики в общем объеме энергетики России, включая в ВИЭ всю систему гидроэлектростанций. Даже сам Аркадий Дворкович, говорит о 40% нетопливной энергетики в стране, прибавляя к гидроэнергетике атомные электростанции. На деле уровень ВИЭ в России в последние минимум 10 лет топчется на уровне от 0,5 до 1 %.

Совокупная установленная мощность в России по всем видам генерации в 2016 году составила 225 ГВт, из них лишь 1% приходится на долю ВИЭ, в том числе 0,6% – биомасса, 0,3% – малые ГЭС, 0,1% – ветряная, солнечная электроэнергетика и геотермальные источники. Распоряжением Правительства РФ от 28 мая 2013 г. №861-р предусмотрено, что доля зеленой энергетики на оптовом рынке к 2020 году должна составить 2,5%, или около 6 ГВт. Есть большая вероятность того, что в ближайшее время будет издано еще распоряжение, в котором прогнозируемая доля ВИЭ на 2020 год будет снижена до 2 %.

Библиографический список

1. Договор о нераспространении ядерного оружия (ДНЯО, 1968 г., вступил в действие в 1970 г.)
2. Лавроски. Висмут-фосфатный метод отделения плутония. Кн. Химия ядерного горючего. М.Госхимиздат, 1956 г., с.153. /Доклады иностранных ученых по мирному использованию атомной энергии, Женева, 1955 г./;
3. Хайд Э.К. Лантан-фторидный метод отделения плутония, там же, с. 393.
4. Магилл И., Хэмилтон Д., Лютценкирхен К., Туфан М., Тамборини Г., Вагнер В., Берту В., фон Цвайдорф А. Последствия события радиологического рассеивания с ядерными и радиоактивными источниками. Наука и всеобщая безопасность. 2007;15(2):12-21.
5. Кесслер Г. Ядерная энергетика. М.: Энергоатомиздат; 1986.
6. Яковлев Р.М., акад. РАН Данилевич Я.Б., Игнатъев М.Б., Суглобов Д.Н. Атомная энергетика без плутония и Чернобыля. Мир и Согласие, №2(35), М. 2008.
7. Суглобов ДН, Яковлев Р.М., акад. РАН Мясоедов Б.Ф. Торий-урановый топливный цикл для тепло- и электроэнергетики. Радиохимия, 2007;49(5):385-392..
8. Целевая правительственной программе интенсивного развития ядерной энергетики России.
9. Зиа Миан, Найяр А. Х., Раджараман Р., Рамана М. В. Делящиеся материалы в южной Азии и последствия американо-индийского ядерного соглашения. Наука и всеобщая безопасность, т.14, вып. 3, 2006 г.
10. Кузякин Ю.И., Яковлев Р.М. Транспортная жидкосолевая реакторная установка. В кн.: Сборник докладов научно-технической конференции «Корабельная ядерная энергетика – взгляд в XXI век» (Ч. 1). Нижний Новгород: ОКБМ; 2001.
11. Яковлев Р.М. О реакторах нового поколения. Атомная Стратегия XXI, 2005;(4): 20-21.
12. Яковлев Р.М., Обухова И.А. Оценка перспектив применения альтернативных моделей реакторов в ядерной энергетике. Сб.Инф.сист.и технологии. Вып.10, Ч.2, СПб, СПбГЛТУ, с.3.-13.
13. Яблоков А.В., Нестеренко В.Б., Нестеренко А.В. Чернобыль: последствия катастрофы для человека и природы. СПб.: Наука; 2011.
14. National Program on Mitigation of Consequences of the Chernobyl NPP Catastrophe
15. Отчёт Комитета по проблемам катастрофы на Чернобыльской АЭС при Совете Министров (Comchernobyl),2015, Минск.
16. Тихонов М.Н., Муратов О.Э. Катастрофы как источники потери устойчивости национальных государств. В кн.: Проблемы риска в техногенной и социальных сферах. СПб; Изд-во СПбГПУ; 2007. с. 119-130.
17. Феоктистов Л.П. Оружие, которое себя исчерпало. М.: Воскресенье; 2003.
18. Список ядерных аварий и инцидентов <https://en.wikipedia.org>

19. Адамов Е.О., Балашов Л.А., Ганев И.Х., Зродников И.В., Кузнецов А.К., Лопаткин А.В., Мастепанов А.М., Орлов А.В., Рачков А.В., Смирнов А.С., Солонин М.И., Ужанова В.В., Черноплеков Н.А., Шаталов Г.Г. Белая книга ядерной энергетики. М.: Изд-во ГУП НИКИЭТ; 2001.
20. Серебряков Б.Е. Создание радиоактивного наследия при ликвидации ядерного. Атомная Стратегия 16.05.2017.
21. Ларин В. Самая дорогая программа спасения России от атомного прошлого. Обеспечение ядерной и радиационной безопасности на 2008 год и на период до 2015 года. (ФЦП ЯРБ 2008-2015). Беллона, Ноябрь, 2009.
22. Муратов О.Э. Стратегические задачи обращения с РАО // Атомная стратегия – XXI, 2006, № 3(23), с.4-6.
23. Нигматуллин Р.И. акад. РАН, директор ин-та Океанологии. Во власти океана. Наука в России (РАН). Июль-август 4(2010),с.54-62.
24. Молдаванов Д.,(интервью с А.П.Капица). Глобальное потепление и озоновые дыры-научнообразные мифы. Вестник. N 20(357) 29/09/ 2004. <http://www.vestnik.com/issues/98/1013/win/moldav.htm>.
25. Концепция вывода из эксплуатации промышленных уран-графитовых реакторов, инв. //№ 04-01281. Утв.Федеральным агентством по атомной энергии 04.02.2005г.
26. Кузнецов В. Чему учит чужой опыт? // Мировая энергетика, 2005, № 5, с. 97-98.
27. Кузнецов В. Опасный возраст // Мировая энергетика, 2005, № 7-8, с. 104-105.
28. Муратов О.Э.Снятие АЭС с эксплуатации: проблемы и пути решения ,// Атомная стратегия. [14/01/2008]
29. Доклад Br.Petr. 2017 edition of BP's Energy Outlook. HOUSTON, Jan. 25 01/25/2017 By OGJ editors.
30. CNPC: World and China Energy Outlook 2050, 2016.
31. Нигматуллин Б.И, Прогноз мирового электропроизводства на АЭС на период до 2050 г. Атомная Стратегия XXI 27/03/2017.

ОГЛАВЛЕНИЕ

А.М. Заяц, Т.И. Игнатъева Математическая модель функционирования беспроводной сенсорной сети.....	3
Н.П. Васильев, Н.В. Лушкин Интеграция гибридных приложений cordova с web-серверной обработкой графики	10
К. С. Веретенникова, Ю.А. Жук Использование нейронных сетей для распознавания растений.....	25
С. П. Васильев, Н.Г. Полетаева Применение методов машинного обучения в задачах оптимизации	28
М.Р. Вагизов, С. А. Тепляков Разработка web-картографического сервиса gis-aero.....	40
В.А. Горбачев Проектирование программного модуля «Учет посещаемости студентов».....	44
В.А. Горбачев Проектирование программного модуля импорта и разнесения оплат из бухгалтерской системы «Парус» в базу данных «Деканат».....	51
М.И. Думов, С.П. Хабаров Моделирование работы узлов wi-fi сети с учетом интерференционных помех от соседних узлов.....	57
Г.В. Жижин, И.А. Обухова Информационная модель передачи наследственных признаков в опытах Менделя.....	67
Б. М. Каримов Система электронного документооборота и управления взаимодействием «directum» в СПбГЛТУ	75
К.М. Сусь Websocket как инструмент для удалённого управления.....	77
Е.В. Тетюев, Ю.А. Жук Использование трёхмерной графики на основе WebGL для создания обучающих и демонстрационных материалов.....	80
М.П. Филяев, А.С. Якшин Актуальные вопросы имитационного исследования транспортных систем.....	86
С.П. Хабаров Использование утилиты websocketd в среде ОС ubuntu server.....	95
С.П. Хабаров Использование simintech для анализа систем автоматического управления.....	106
М.Л. Шилкина Техническая оптимизация веб-приложения для улучшения позиций в поисковой выдаче.....	117
М.А. Шубина Организация хранения и доступа к дистанционным данным.....	132
Р.М. Яковлев, И.А. Обухова Ситуация с нераспространением ядерного оружия и проблемы развития атомной энергетики	144

ДЛЯ ЗАМЕТОК

Научное издание

Отв. редактор
Зяц Анатолий Моисеевич

ИНФОРМАЦИОННЫЕ СИСТЕМЫ И ТЕХНОЛОГИИ:
ТЕОРИЯ И ПРАКТИКА

Сборник научных трудов

В ы п у с к 11

В авторской редакции с готового оригинал-макета

Подписано в печать 29.04.19.
Усл.-печ. л. 10,0. Заказ № 34. С 6.

Санкт-Петербургский государственный лесотехнический университет
Издательско-полиграфический отдел СПбГЛТУ
194021, Санкт-Петербург, Институтский пер., 3