Intellectual Technologies on Transport No 3

Интеллектуальные технологии на транспорте № 3

Санкт-Петербург
St. Petersburg
2020

Интеллектуальные технологии на транспорте
№ 3, 2020

ISSN 2413-2527

Сетевой электронный научный журнал, свободно распространяемый через Интернет.
Публикуются статьи на русском и английском языках с результатами исследований
и практических достижений в области интеллектуальных технологий
и сопутствующих им научных исследований.

Журнал основан в 2015 году.

Учредитель и издатель
Федеральное государственное бюджетное образовательное учреждение высшего образования
«Петербургский государственный университет путей сообщения Императора Александра I» (ФГБОУ ВО ПГУПС)

Адрес редакции:
190031, Санкт-Петербург, Московский пр., 9, ауд. 1–210
e-mail: itt-pgups@yandex.ru

Журнал зарегистрирован Федеральной службой по надзору в сфере связи и массовых коммуникаций, свидетельство ЭЛ № ФС77-61707 от 07 мая 2015 г.

Журнал зарегистрирован в Российском индексе научного цитирования (РИНЦ).

Периодичность выхода – 4 номера в год. Выпуски журнала доступны на сайте http://itt-pgups.ru.

Копии архивов с выпусками журнала проходят государственную регистрацию как электронное издание сетевого распространения в НТЦ "Информрегистр".

Информация предназначена для детей старше 12 лет.

Intellectual Technologies on Transport
Issue 3, 2020

ISSN 2413-2527

Network electronic scientific journal, open access. It publishes articles in Russian and English with the results of research and practical achievements in the field of intelligent technologies and associated research.

Founded in 2015.

Founder and Publisher

Federal State Educational Institution of Higher Education
«Emperor Alexander I Petersburg State Transport University»

Editorial adress:

190031, St. Petersburg, Moskovsky ave., 9, aud. 1–210
e-mail: itt-pgups@yandex.ru

The journal is registered by the Federal Service for Supervision of Communications and Mass Media, EL no. FS77-61707 testimony from May 7, 2015.

The journal is registered in the Russian Science Citation Index (RSCI).

Frequency of release - 4 issues per year. Issues of the magazine are available at http://itt-pgups.ru.

Copies of the archives with the issues of the journal are state-registered as an electronic publication of network distribution in the Scientific and Technical Center "Informregister".

The content is for children over the age of 12.

Содержание

Contents

Алгоритм поиска кратчайшего пути между подвижными объектами транспортной сети

А. В. Калюжный, к.т.н. В. Г. Терехов, С. С. Зыкова
Военно-космическая академия имени А. Ф. Можайского
Санкт-Петербург, Россия
aleksei.kalyuzhnyi@yandex.ru

Аннотация. **Рассмотрен способ поиска оптимального маршрута информационного обмена между подвижными объектами транспортной сети. Разработан алгоритм поиска кратчайшего пути от одного подвижного объекта до всех остальных на основе алгоритма Дейкстры. Приведен конкретный пример реализации предлагаемого в статье алгоритма. Представлены результаты имитационного моделирования работы разработанного алгоритма в сравнении с алгоритмом поиска кратчайшего пути в графе Беллмана-Форда при увеличении количества вершин в графе. Применение разработанного алгоритма позволит построить оптимальный маршрут информационной взаимосвязи подвижных объектов, что значительно повысит эффективность управления транспортными сетями.**

Ключевые слова**: транспортная сеть, информационный обмен, подвижные объекты, поиск кратчайшего пути в графе, алгоритм Дейкстры.**

Введение

Активное развитие транспортных сетей (ТС) приводит к росту информационных потоков между участниками данных сетей, а именно подвижными объектами. Динамическое изменение ситуации в ТС и изменение числа объектов ТС приводит к необходимости использования моделей и алгоритмов адаптивной маршрутизации [1–3]. Так как подвижные объекты могут выходить из зоны видимости, терять работоспособность, что вынуждает осуществлять доставку информации от одного подвижного объекта к другому через соседние объекты «по цепочке», необходимо вариативно определять маршруты применительно к текущей ситуации в сети [4]. В связи с этим возникает задача поиска кратчайшего пути между подвижными объектами с целью построения оптимальных маршрутов их взаимосвязи в случае возникновения нештатных ситуаций, решение которых возможно осуществить с помощью применения алгоритма Дейкстры [5]. Суть работы данного алгоритма основана на анализе ориентированного взвешенного графа при условии, что все ребра в графе имеют неотрицательные веса.

В данной статье представлен алгоритм поиска кратчайшего пути между подвижными объектами для построения оптимального маршрута их информационного взаимодействия на основе использования алгоритма Дейкстры.

Постановка задачи

Транспортная сеть, состоящая из подвижных объектов, моделируется графом, в котором каждый подвижный объект является его вершиной, линии связи – ребрами графа, а вес каждого ребра соответствует расстоянию между соответствующими подвижными объектами. Необходимо найти кратчайшие расстояния от начальной вершины графа до всех остальных. Начальной вершине соответствует подвижный объект ТС, который инициирует передачу информации другим подвижным объектам. Общий вид анализируемого графа представлен на рисунке 1.

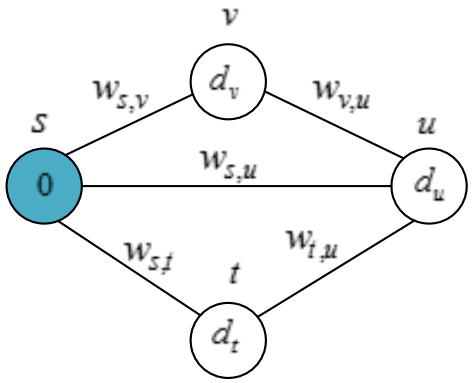

Рис. 1. Общий вид анализируемого графа

Постановку задачи можно представить следующим образом.

Дано:

$G(V, E)$ — граф;

V — множество вершин графа;

E — множество ребер графа;

$w_{i,j}$ — вес (длина) ребра из вершины i в вершину j;

w_{max} — максимальный вес ребра;

$D = \{d_1, d_2, \dots, d_k\}$ — множество расстояний от начальной вершины до всех остальных вершин (k — количество вершин в графе);

$P = \{p_1, p_2, \dots p_k\}$ — множество значений-индикаторов (0 или 1) завершения «прохода» вершин;

s — вершина, инициирующая передачу данных.

Найти:

кратчайшее расстояние от начальной вершины до всех остальных вершин графа с ограничениями: $w_{i,j} < w_{max}$.

С учетом вышеизложенного разработан алгоритм поиска кратчайшего пути между подвижными объектами ТС, схема которого представлена в следующем разделе.

Алгоритм поиска кратчайшего пути между подвижными объектами транспортной сети

Схема алгоритма поиска кратчайшего пути между подвижными объектами ТС изображена на рисунке 2.

Пошаговое описание работы алгоритма представлено в таблице 1.

Кроме того, при работе алгоритма необходимо обратить внимание на следующие особенности:

– на этапе инициализации все подвижные объекты ТС посылают широковещательные запросы всем остальным

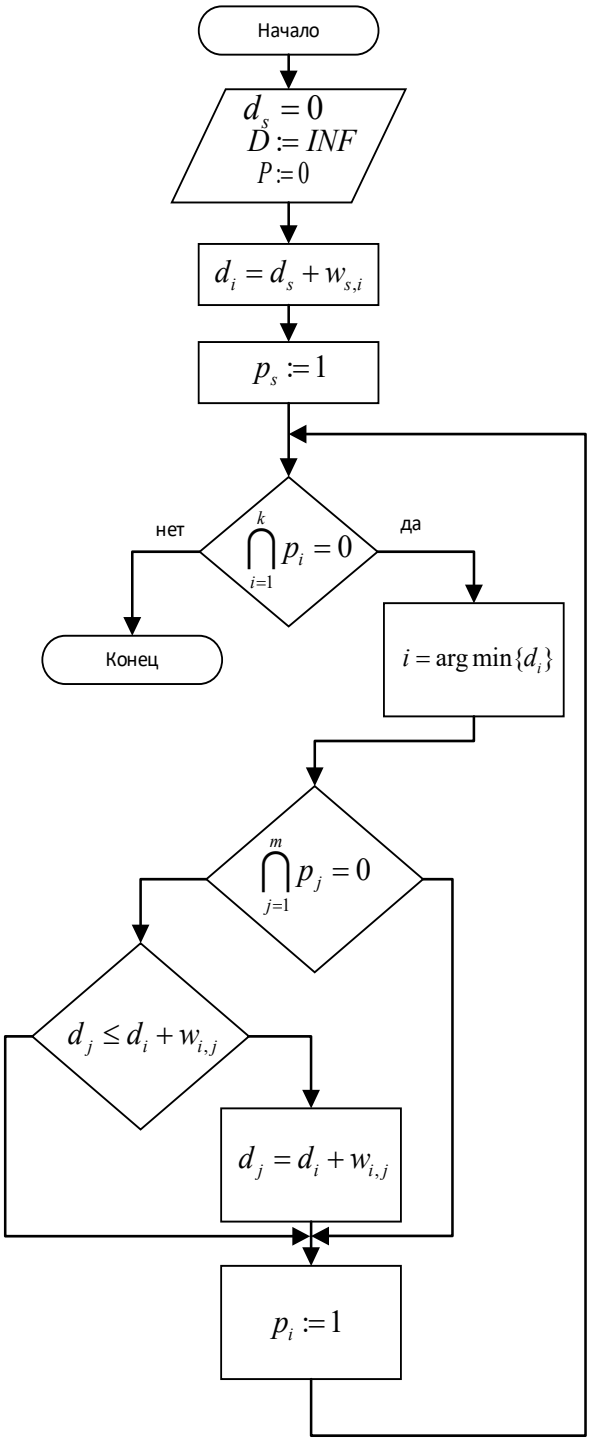

Рис. 2. Схема алгоритма поиска кратчайшего пути между подвижными объектами транспортной сети

подвижным объектам, по ответам которых строится таблица смежности;

– в определенные моменты времени состоянию подвижных объектов ТС соответствует взвешенный граф, в котором весами вершин будет расстояние между подвижными объектами;

– если между вершинами i и j данного графа нет ребра, то его весу присваивается значение INF (не определено, бесконечность, либо значение заведомо большее максимально возможного веса) [6–8].

Таблица 1

Пошаговое описание алгоритма поиска кратчайшего пути между подвижными объектами транспортной сети

Шаги	**Описание**
1.	Начало.
2.	Установить метку 0 исходной вершине ($d_s = 0$), а всем остальным — установить метки INF ($D := INF$). Это означает, что расстояния до остальных вершин пока неизвестны. Все вершины графа помечаются как непройденные ($P := 0$).
3.	Установить во всех i-х вершинах, смежных с вершиной s, значение метки, равное $d_i = d_s + w_{s,i}$.
4.	Пометить вершину s как пройденную ($p_s := 1$).
5.	Проверить, имеются ли в графе не пройденные вершины. Если таковые имеются, то перейти к шагу 6, иначе – к шагу 12.
6.	Поиск номера i-й вершины с минимальной меткой $i = arg\ \underset{i}{min}\{d_i\}$.
7.	Если существуют смежные вершины, не помеченные как пройденные, то выполнить переход на шаг 8, иначе – на шаг 10.
8.	Если $d_j \le d_i + w_{i,j}$, то переход на шаг 10, иначе — на шаг 9.
9.	Установить значение метки равным $d_j = d_i + w_{i,j}$.
10.	Пометить вершину i как пройденную ($p_i := 1$).
11.	Переход к шагу 5.
12.	Конец.

ОПИСАНИЕ ПРИМЕРА РАБОТЫ АЛГОРИТМА

Рассмотрим работу алгоритма на примере, представленном на рисунке 3.

На первом этапе работы алгоритма (рис. 3, *а*) вершина 1 взвешенного графа инициирует передачу данных. При этом метки во всех остальных вершинах не определены, а отмечены символом ∞, и ни одна из вершин не отмечена как пройденная. На рисунке 3, *б* видно, что выставлены метки в вершинах 3 и 4, смежных с вершиной 1, вершина 1 помечена как пройденная, а этап переходит к вершине с минимальной меткой, то есть к вершине 3.

Далее (рис. 3, *в*) осуществляется выставление меток в вершинах, смежных с вершиной 3. На рисунке 3, *г* изображен переход к вершине с минимальной меткой, то есть к вершине 4, и выставление меток смежным для нее вершинам. Значение метки во второй вершине меняется на 30, так как оно меньше значения, записанного там ранее. Вершина 3 отмечается как пройденная.

Процесс улучшения значения меток в смежных вершинах называется релаксацией.

На рисунке 3, *д* имеются две вершины с одинаковыми метками. Согласно принципу работы алгоритма выбирается любая из них. На этом этапе переходим к вершине 2, происходит релаксация значений меток в смежных вершинах, и в седьмую вершину записывается новое значение, равное 50.

На рисунке 3, *е* показан переход к вершине с минимальной меткой, т. е. вершине 5, а вершина 2 отмечается как пройденная. Далее переходим к вершине 6 (рис. 3, *ж*). Так как вершина 5 не имеет ребер, смежных с еще не пройденными вершинами, она отмечается как пройденная. На рисунке 3, *з* вершина 6, также не имеющая смежных непройденных вершин, отмечается как пройденная. Так как все вершины пройдены, то процесс функционирования алгоритма заканчивается.

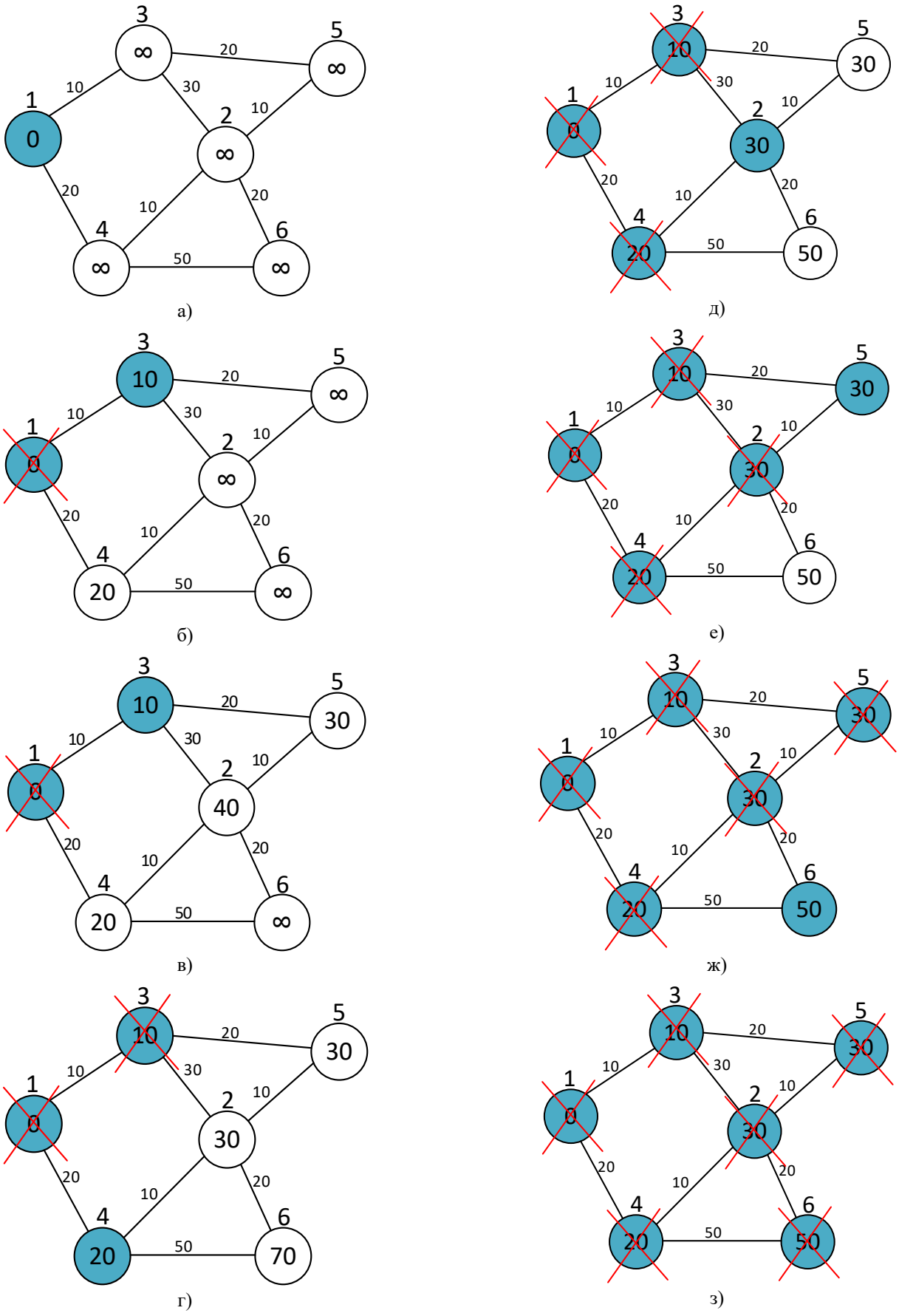

Рис. 3. Пример выполнения алгоритма поиска кратчайшего пути

По завершении всех этапов работы алгоритма составляется таблица маршрутизации с кратчайшими расстояниями от подвижного объекта, инициализирующего передачу данных, до всех остальных.

Таким образом, на основе анализа полученных результатов кратчайших расстояний между подвижными объектами ТС, возможно не только повысить оперативность обмена информацией между подвижными объектами в ТС, но и найти оптимальный путь взаимосвязи между подвижными объектами в случае выхода из строя или потери связи с одним из них.

Сравнение оперативности работы предложенного алгоритма с алгоритмом Беллмана-Форда

Для оценки оперативности работы алгоритма возьмем для сравнения алгоритм поиска кратчайшего пути во взвешенном ориентированном графе Беллмана-Форда. Он, в отличие от алгоритма Дейкстры, может использоваться при наличии отрицательных весов ребер. Однако при проведении имитационного моделирования это свойство не учитывалось, так как было задано ограничение на значения весов ребер в графе, которые не могли принимать отрицательные значения.

При оценке сложности алгоритмов используется специальная величина под названием Big-O «большая О». Она позволяет оценить, насколько время выполнения алгоритма зависит от переданных данных. Разработанный алгоритм, как и алгоритм Дейкстры, имеет сложность $O((V+E)\times log\,V)$, тогда как алгоритм Беллмана-Форда имеет сложность равную $O(V\times E)$ [9]. Сравним влияние увеличения количества вершин в графе на время выполнения алгоритмов.

Все эксперименты проводились при одинаковых условиях, на ПЭВМ с процессором Intel Core i5.

В таблице 2 представлены результаты имитационного моделирования поиска кратчайшего пути в текстовых графах, вершины и ребра в которых генерировались каждый раз случайным образом.

Таблица 2

Время вычисления кратчайших путей из одной вершины

V	E	Разработанный алгоритм поиска кратчайшего пути	Алгоритм Беллмана-Форда
6	8	0,000002	0,000001
66	464	0,000012	0,000005
258	3 872	0,00006	0,000041
1 026	31 808	0,000338	0,000287
2 027	89 190	0,0022	0,00347
3 027	163 460	0,0038	0,00535
4 098	258 176	0,0043	0,0076
6 402	505 760	0,0089	0,0144
8 102	721 080	0,0157	0,0248
10 002	990 200	0,0223	0,0351

Результатом является время (в секундах) вычисления кратчайших путей из одной вершины, каждым алгоритмом, для сгенерированных графов с $|V|=k^2+2$ вершинами и $|E|=k^3-k^2+2k$ ребрами.

На рисунке 4 представлена зависимость значения времени выполнения разработанного алгоритма и алгоритма Беллмана-Форда от числа вершин в графе.

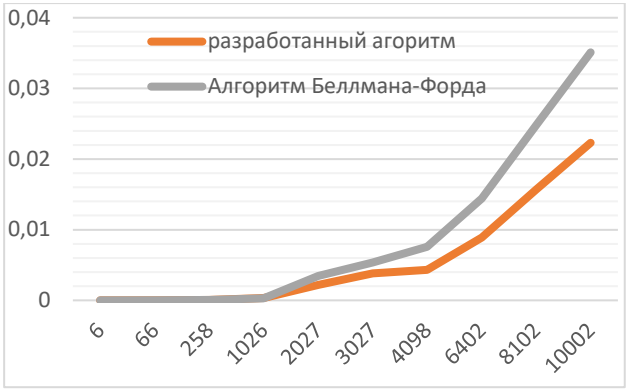

Рис. 4. Зависимость времени выполнения алгоритмов от числа вершин в графе

Как видно из рисунка 4, наибольший выигрыш во времени работы разработанного алгоритма достигается при наличии в графе более 4 100 вершин.

Анализ результатов моделирования показал, что время выполнения алгоритмов, при 8 100 вершинах в графе, уменьшается с 0,0248 с для алгоритма Беллмана-Форда до 0,0157 с для разработанного алгоритма [10].

Применение разработанного алгоритма позволяет повысить оперативность поиска кратчайшего пути в графе примерно в 1,5 раза, при наличии в графе более 4 000 вершин.

Заключение

В статье рассмотрен алгоритм поиска кратчайшего пути между подвижными объектами ТС для построения оптимального маршрута их информационного обмена на основе использования алгоритма Дейкстры. Результаты имитационного моделирования показали, что применение разработанного алгоритма позволяет повысить оперативность поиска кратчайших путей в графе при наличии в графе более 4 000 вершин. Применение разработанного алгоритма позволит построить оптимальный маршрут информационной взаимосвязи подвижных объектов, что значительно повысит эффективность управления ТС за счет более оперативного сбора и обмена информацией.

Литература

1. Дышленко С. Г. Маршрутизация в транспортных сетях // ИТНОУ: Информационные технологии в науке, образовании и управлении. 2018. № 1 (5). С. 15–20.

2. Фокин В. Г. Оптические системы передачи и транспортные сети: Учебное пособие для студентов, обучающихся по направлению «Телекоммуникации». — М.: Эко-Трендз, 2008. — 285 с. — (Инженерная энциклопедия. Технологии электронных коммуникаций).

3. Рафгарден Т. Совершенный алгоритм. Графовые алгоритмы и структуры данных = Algorithms Illuminated. Part 2: Graph Algorithms and Data Structures. — СПб.: Питер, 2019. — 256 с. — (Библиотека программиста).

4. Рудь Д. Е. Технологии топологической оптимизации трафика информационных потоков в телекоммуникационных сетях // Инженерный вестник Дона. 2010. № 2 (12). С. 95–107.

5. Dijkstra E. W. A note on two problems in connexion with graphs // Numerische Mathematik. 1959. Vol. 1. Pp. 269–271. DOI:10.1007/BF01386390.

6. Cormen T. H. Introduction to Algorithms. First Edition / T. H. Cormen, C. E. Leiserson, R. L. Rivest. — MIT Press, 1990. — 1048 p.

7. Назаров А. Н. Модели и методы расчета показателей качества функционирования узлового оборудования и структурно-сетевых параметров сетей связи следующего поколения / А. Н. Назаров, К. И. Сычев. — Красноярск: Поликом, 2010. — 389 с.

8. Лёвин Б. А. Информационное моделирование при управлении транспортом // Перспективы науки и образования. 2017. № 3 (27). С. 50–54.

9. Хайнеман Д. Т. Алгоритмы. Справочник с примерами на C, C++, Java и Python. Второе издание = Algorithms in a Nutshell. Second Edition. / Д. Т. Хайнеман, Г. Поллис, С. Селков; пер. с англ. и ред. И. В. Красикова. — СПб.: Издательство «Диалектика»: ООО «Альфа книга», 2017. — 432 с.

10. Коваленко Т. А. Анализ алгоритмов маршрутизации в вычислительных сетях // Глобальный научный потенциал. 2011. № 9. С. 41–45.

Algorithm for Finding the Shortest Path Between Moving Objects in the Transport Network

A. V. Kalyuzhnyy, PhD V. G. Terekhov, S. S. Zykova
A. F. Mozhaisky Military Space Academy
Saint Petersburg, Russia
aleksei.kalyuzhnyi@yandex.ru

Abstract. **A method for finding the optimal route for information exchange between mobile objects of the transport network is considered. An algorithm for finding the shortest path from one moving object to all the others has been developed based on Dijkstra's algorithm. A specific example of the implementation of the algorithm proposed in the article is given. The results of simulation modeling of the developed algorithm are presented in comparison with the algorithm for finding the shortest path in the Bellman-Ford graph with an increase in the number of vertices in the graph. Application of the developed algorithm will make it possible to build an optimal route for information interconnection of mobile objects, which will significantly increase the efficiency of transport network management.**

Keywords**: transport network, information exchange, mobile objects, search for the shortest path in the graph, Dijkstra algorithm.**

REFERENCES

1. Dyshlenko S. G. Routing in Transport Networks [Marshrutizatsiya v transportnykh setyakh], *Information Technologies in Science, Education and Management [ITNOU: Informatsionnye tekhnologii v nauke, obrazovanii i upravlenii.]*, 2018, No. 1(5), Pp. 15–20.

2. Fokin V. G. Optical transmission systems and transport networks: Study guide [Opticheskie sistemy peredachi i transportnye seti: Uchebnoe posobie], Moscow, Eco-Trends, 2008, 285 p.

3. Roughgarden T. Algorithms Illuminated. Part 2: Graph Algorithms and Data Structures [Sovershennyy algoritm. Grafovye algoritmy i struktury dannykh], St. Petersburg, Piter Publishing House, 2019, 256 p.

4. Rud D. Ye. Technologies for Topological Optimization of Traffic of Information Flows in Telecommunication Networks [Tekhnologii topologicheskoy optimizatsii trafika informatsionnykh potokov v telekommunikatsionnykh setyakh], *Engineering Journal of Don [Inzhenernyy vestnik Dona]*, 2010, No. 2 (12), Pp. 95–107.

5. Dijkstra E. W. A Note on Two Problems in Connexion with Graphs, *Numerische Mathematik*, 1959, Vol. 1, Pp. 269–271. DOI:10.1007/BF01386390.

6. Cormen T. H., Leiserson C. E., Rivest R. L. Introduction to Algorithms. First Edition. MIT Press, 1990, 1048 p.

7. Nazarov A. N., Sychev K. I. The models and the methods of measuring of quality indicators of nodal equipment functioning and network structural parameters of next generation networks [Modeli i metody rascheta pokazateley kachestva funktsionirovaniya uzlovogo oborudovaniya i strukturno-setevykh parametrov setey svyazi sleduyushchego pokoleniya], Krasnoyarsk, Polikom Publishers, 2010, 389 p.

8. Lyovin B. A. Information Modeling in Transport Management [Informatsionnoe modelirovanie pri upravlenii transportom], *Perspectives of Science and Education [Perspektivy nauki i obrazovaniya]*, 2017, No. 3 (27), Pp. 50–54.

9. Heineman G. T., Pollice G., Selkow S. Algorithms in a Nutshell. Second Edition [Algoritmy. Spravochnik s primerami na C, C++, Java i Python. Vtoroe izdanie], St. Petersburg, Dialektika Publishers, Alpha Kniga LLC, 2017, 432 p.

10. Kovalenko T. A. Analysis of Routing Algorithms in Computer Networks [Analiz algoritmov marshrutizatsii v vychislitel'nykh setyakh], *Global Scientific Potential [Global'nyy nauchnyy potentsial]*, 2011, No. 9, Pp. 41–45.

Модель и методика контроля качества бортовых систем космических аппаратов

к.т.н. А. М. Барановский, А. В. Кикоть, к.т.н. Е. Н. Шаповалов
АО «НИИ программных средств»
Санкт-Петербург, Россия
bamvka@mail.ru, a.v.kikot@yandex.ru, henya56@mail.ru

Аннотация. **Предложена модель контроля качества бортовых систем космического аппарата (КА), основанная на иерархии конечных автоматов. В качестве исходных данных для построения модели предлагается модель бортовых систем (БС), основанная на принципах алгебраического агрегирования, отличающаяся расширением образа состояния БС, состояниями подсистем. Разработанная модель построена по модульному принципу, что позволяет создавать и оперативно изменять конфигурацию процесса контроля, варьируя глубину контроля и состав проверяемых систем. Предложена основанная на иерархии конечных автоматов методика контроля качества бортовых систем, применение которой позволит в полной мере использовать преимущества «протолетного» подхода к созданию КА.**

Ключевые слова: **космический аппарат, иерархический конечный автомат, контроль качества, бортовая система, «протолетный» подход.**

Введение

Качество бортовых систем (БС) космических аппаратов (КА) определяется как совокупность свойств, характеризующих их соответствие назначению [1]. Процесс контроля качества БС осуществляется на всех этапах как наземной, так и летной эксплуатации. На этапе наземной отработки БС средствами контроля качества выступают наземные средства контроля и диагностирования (НСКД), входящие в состав автоматизированных испытательных комплексов (АИК), на этапе летной эксплуатации для современных и перспективных КА эти задачи возлагаются на бортовые системы контроля и диагностирования (БСКД). Последние представляют собой многоуровневые распределенные программно-аппаратные комплексы, реализованные на вычислительных средствах БС (как правило, микроконтроллерах) [2].

Сущность процесса контроля качества представляет собой измерение значений свойств БС и отнесение БС к одному из классов состояний, внутри которого объекты обладают одинаковым качеством. При этом к решению о принадлежности БС к определенному классу предъявляются достаточно высокие требования по достоверности и оперативности. Кроме того, при реализации процесса контроля качества средствами БСКД накладываются существенные ограничения по затратам ресурсов (в первую очередь энергетических и вычислительных) [3]. Наибольшую эффективность в данных условиях показал подход, основанный на принципе алгебраического агрегирования [4], который позволяет строить гибкие оптимальные программы контроля для отдельных технических систем. Целью статьи является расширение возможностей данного подхода для его применения к иерархическому объекту контроля, а также разработка модели контроля иерархического объекта, которая может быть реализована как на НСКД, так и на вычислительных средствах БСКД.

Постановка задачи

БС КА представляют собой множество систем, объединенных в иерархию, которые можно представить в следующем виде:

$$BS = \langle E, r \rangle. \quad (1)$$

Здесь E — множество систем КА различного уровня иерархии; r — бинарное отношение, отражающее структуру иерархической системы

$$r = \langle E, R \rangle, R = \{\langle e_i, e_k \rangle | \varepsilon(e_i, e_k) = 1\}, R \subseteq E \times E ,$$

где

$$\varepsilon(e_i, e_k) = \begin{cases} 1, \text{если } e_k \text{ входит в состав } e_i; \\ 0, \text{в противном случае.} \end{cases}$$

Объектом контроля является бортовая система e_i, которую можно представить кортежем

$$e_i = \langle E_i^*, S_i, \Pi_i, \widehat{\Pi}_i, \xi_i, \eta_i, \rho_i \rangle, \quad (2)$$

где E_i^* — множество подсистем системы e_i,

$$E_i^* = \{e_k | \langle e_i, e_k \rangle \in R\};$$

S_i — множество классов состояний системы e_i; Π_i — множество проверок свойств БС; $\widehat{\Pi}_i$ — множество исходов проверок свойств БС; ξ_i — отображение, ставящее в соответствие проверке множество ее исходов,

$$\xi_i : \Pi_i \to \widehat{\Pi}_i ;$$

η_i — отображение, определяющее для каждого j-го состояния $s_i^j \in S_i$ исход проверки $\pi_i^n \in \Pi_i$,

$$\eta_i : S_i \times \Pi_i \to \widehat{\Pi}_i ;$$

ρ_i — отображение, определяющее для каждого j-го состояния $s_i^j \in S_i$ состояние подсистемы $e_k \in E_i^*$,

$$\rho_i : S_i \times E_i^* \to \bigcup_{k | e_k \in E_i^*} S_k .$$

Требуется разработать модель процесса контроля качества БС КА.

Работа выполнена по договору № 2580/НИИПС с Научно-исследовательским институтом космических систем имени А.А. Максимова — филиалом АО «Государственный космический научно-производственный центр имени М. В. Хруничева» на выполнение составной части научно-исследовательской работы «Технология СГ-3.3.3.2» для государственных нужд в рамках программы Союзного государства «Технология-СГ».

Модель процесса контроля качества бортовых систем

Под проверкой в обобщенном смысле понимается эксперимент, связанный с оцениванием свойства БС при подаче входных воздействий, которые могут быть как тестовыми, так и рабочими. Проверка выполняется по определенному алгоритму, который характеризуется затратами времени, ресурсов, а также ошибками первого и второго рода. Исход проверки представляет собой критерий оценивания свойства системы. Ввиду наличия подсистем множество исходов всех проверок расширяется множеством состояний подсистем и составляет изображение состояния БС, т. е.

$$s_i^j = \langle (\pi_i^{j1}, \pi_i^{j2}, \dots), (s_{k_1}^j, s_{k_2}^j, \dots) \rangle \qquad (3)$$

Процесс контроля реализуется путем выполнения ряда проверок по заданной программе. Методы разработки оптимальных программ контроля в настоящее время достаточно хорошо разработаны [5–8]. В основу этих методов положен метод динамического программирования, который достаточно трудоемок, так как требует предварительного определения и хранения в памяти ЭВМ всех возможных реализаций диагностического процесса и выбора наилучшего из них по выбранному критерию. Однако, для реализации иерархической БСКД, структура которой предложена в [2], используются встроенные вычислительные средства бортовых систем и блоков, как правило, представляющие собой микроконтроллеры.

Синтез оптимальных программ контроля представляет собой набор рекуррентных процедур, применение которых зачастую приводит к возникновению ошибок, связанных с переполнением стека. Кроме того, применение рекуррентных процедур с неопределенной степенью вложенности на микроконтроллере может привести к нештатным ситуациям, таким как зависание или перезагрузка последнего. Гарантированно избежать подобных нештатных ситуаций, связанных с работой вычислительных средств БСКД, позволит осуществление синтеза оптимальной программы контроля на наземных средствах с последующей загрузкой в БСКД готовой программы контроля, представленной в виде ориентированного графа, определяющего для каждой i-й бортовой системы (2) конечный автомат Мура [9]

$$a_i = \langle X_i, Y_i, Z_i, \varphi_i, \psi_i, z_i^0 \rangle. \qquad (4)$$

Все автоматы объединены в иерархию (рис. 1)

$$BCDS = \langle A, r \rangle, \qquad (5)$$

изоморфную по отношению к иерархии BS (1) [10].

Множество состояний конечного автомата $Z_i = \{z | z \subseteq S_i\}$ представляет собой σ-алгебру подмножеств множества S_i, элементы которой называются информационными состояниями процесса контроля (физически они представляют собой подмножества «подозреваемых» состояний, в одном из которых находится наблюдаемое состояние системы). Начальное состояние включает в себя все множество состояний $z_i^0 = S_i$.

В каждом состоянии автомат Мура формирует выходной сигнал, который представляет собой команду на выполнение оптимальной в данном состоянии проверки в соответствии с программой контроля. Если в текущем информационном состоянии $z \in Z_i$ программой контроля предусмотрено проведение проверки $\pi \in \Pi_i$, то формируется воздействие на объект управления с целью выполнения проверки. Это может быть запрос показаний датчика (функциональная проверка) или осуществление воздействия на систему с последующим измерением показаний датчиков (тестовая проверка). В данном состоянии автомата входом, изменяющим его состояние, является исход проверки $\hat{\pi} \in \hat{\Pi}_i$. Если в данном информационном состоянии предусмотрено определение состояния подсистемы, на выходе формируется команда на запуск автомата $a_k \in A_i^*$, где $A_i^* = \{a_k | \langle a_i, a_k \rangle \in R\}$.

Таким образом, множество выходных воздействий имеет вид:

$$Y_i = A_i^* \cup \Pi_i, \qquad (6)$$

а функция выходов представляет собой отображение, ставящее в соответствие множеству состояний множество выходов (6)

$$\psi_i : Z_i \to Y_i. \qquad (7)$$

Множество входов конечного автомата представляет собой множество исходов проверок свойств БС в совокупности с множеством состояний подсистем, определяемых работой вложенных автоматов

$$X_i = \hat{\prod}_i \cup \left(\cup_{k | a_k \in A_i^*} Z_k \right) \qquad (8)$$

Функция переходов конечного автомата определяется как отображение, ставящее в соответствие множеству состояний и множеству входных сигналов (8) множество последующих состояний

$$\varphi_i : X_i \times Z_i \to Z_i \qquad (9)$$

Функции переходов (9) и выходов (7) автомата удобно представлять в форме таблицы (табл. 1). Конечный автомат не формирует выхода в случае, если он находится в одном из конечных состояний, т. е. $|z_k| = 1$.

Таблица 1

Пример функции переходов конечного автомата

	Проверки	**Исходы проверок**	z_1	z_2	z_3	z_4	…
Функция переходов (9)							
$\hat{\Pi}_i$	π_{i1}	π_{i1}^+	z_1	z_4	z_3	z_4	…
		π_{i1}^-	z_1	z_5	z_3	z_4	…
	π_{i2}	π_{i2}^+	z_2	z_2	z_3	z_4	…
		π_{i2}^-	z_2	z_2	z_3	z_4	…
	…	…	…	…	…	…	…
	Множества состояний вложенных систем	Технические состояния вложенных систем					
Z_i^*	$S_{k_1} \| a_{k_1} \in A_i^*$	$s_{k_1}^1 \in S_{k_1}$	z_1	z_2	z_3	z_8	…
		$s_{k_1}^2 \in S_{k_1}$	z_1	z_2	z_3	z_9	…
	$S_{k_2} \| a_{k_2} \in A_i^*$	$s_{k_2}^1 \in S_{k_2}$	z_1	z_2	z_6	z_4	…
		$s_{k_2}^2 \in S_{k_2}$	z_1	z_2	z_7	z_4	…
…	…	…	…	…	…	…	…
Функция выходов (7)							
Выход конечного автомата			π_{i2}	π_{i1}	…	s_i^2	…

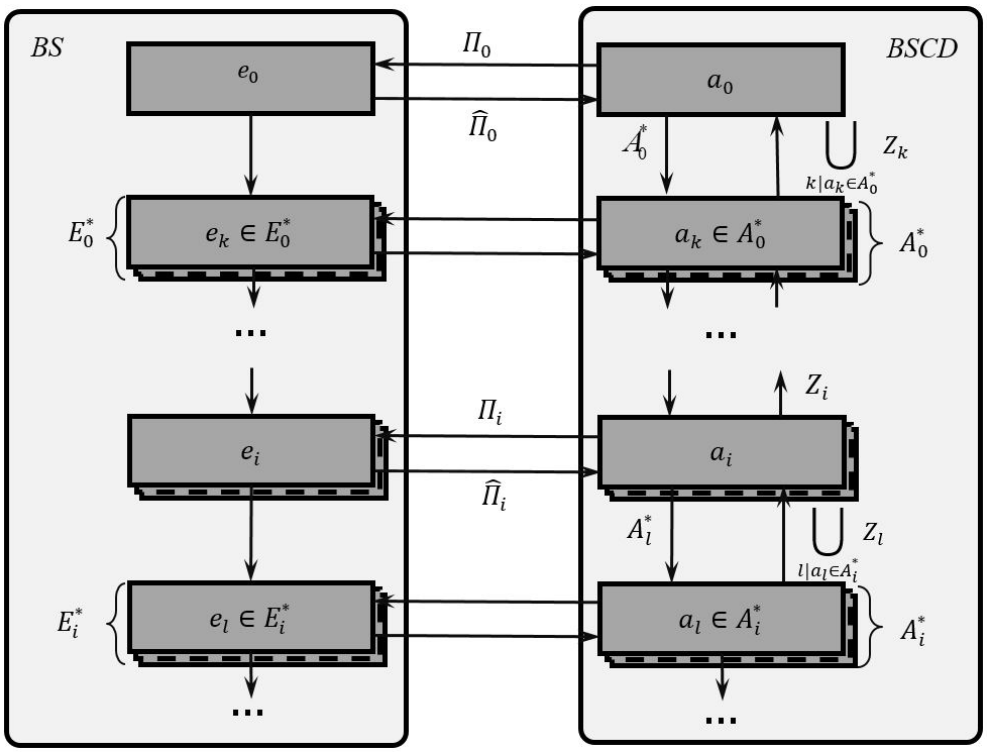

Рис. 1. Структура иерархического автомата

Отличительной особенностью предлагаемой модели является взаимодействие между различными уровнями иерархии (5) конечных автоматов (4) за счет расширения образа технического состояния (3) множеством технических состояний подсистем. При этом иерархия обладает свойством модульности — добавление и удаление модели подсистемы возможно добавлением (удалением) нового автомата и изменением его связей в операторе сопряжения *r*. Модульность системы позволяет реализовать разработанную модель на распределенных вычислительных средствах БСКД следующим образом: автоматы нижнего уровня реализуются на встроенных вычислительных средствах бортовых систем и модулей, автоматы верхнего уровня — в составе программного обеспечения бортовой вычислительной системы.

Модель контроля качества аккумуляторной батареи системы электропитания

Рассмотрим иерархический конечный автомат для управления процессом контроля аккумуляторной батареи, состоящей из 3 последовательно соединенных никель-водородных аккумуляторов (НВА)[1] [11].

$$BS = \langle E, r \rangle.$$

$$E = \{e_0, e_1, e_2, e_3\}.$$

e_0 — аккумуляторная батарея;
e_1 — НВА № 1;
e_2 — НВА № 2;
e_3 — НВА № 3.

$$r = \begin{bmatrix} 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}.$$

Модель НВА № 1 (модели НВА № 2 и НВА № 3 аналогичны) имеет вид (2) Компонентами модели являются:

1. Поскольку НВА не имеет подсистем, то $E_1^* = \emptyset$, следовательно, отношение ρ_1 не существует.

2. $S_1 = \{s_1^0, s_1^1, s_1^2, s_1^3, s_1^4\}$, где s_1^0 — исправное состояние, s_1^1 – необратимая деградация, s_1^2 — разгерметизация, s_1^3 — короткое замыкание, s_1^4 — обрыв цепи.

3. $\Pi_1 = \{\pi_1^1, \pi_1^2, \pi_1^3, \pi_1^4, \pi_1^5, \pi_1^6\}$, где π_1^1 — проверка давления p, π_1^2 — проверка перепада давленияΔp, π_1^3 — проверка напряжения заряда $U_{зар}$, π_1^4 — проверка напряжения разряда $U_{разр}$, π_1^5 — проверка температуры t, π_1^6 — проверка силы тока в цепи I.

4. Отношения ξ_1 и η_1 составляют диагностическую модель НВА, которую можно представить в форме таблицы 2.

Модель аккумуляторной батареи имеет вид (2). Компонентами этой модели являются:

1. $E_0^* = \{e_1,\ e_2,\ e_3\}$.

2. $S_0 = \{s_0^0, s_0^1, s_0^2, s_0^3, s_0^4\}$, где s_0^0 — исправное состояние, s_0^1 — разбаланс степеней заряженности, s_0^2 — неисправность НВА № 1, s_0^3 — неисправность НВА № 2, s_0^4 — неисправность НВА № 3.

3. $\Pi_0 = \{\pi_0^1\}$, где π_0^1 — проверка разности степеней заряженности НВА в модуле (Δ).

4. Отношения ξ_0, ρ_0 и η_0 можно представить в форме таблицы 3.

[1]В никель-водородных аккумуляторных батареях, используемых в системах электропитания современных космических аппаратов более 20 аккумуляторов, однако в рассматриваемом примере в целях обеспечения наглядности размерность задач Z_i уменьшена.

Таблица 2

Диагностическая модель НВА

		$\boldsymbol{\Pi_1}$					
		$\boldsymbol{\pi_1^1}$	$\boldsymbol{\pi_1^2}$	$\boldsymbol{\pi_1^3}$	$\boldsymbol{\pi_1^4}$	$\boldsymbol{\pi_1^5}$	$\boldsymbol{\pi_1^6}$
$\boldsymbol{S_1}$	$\boldsymbol{s_1^0}$	$\hat{\pi}_1^{11}$ $(p \geq p_{min})$	$\hat{\pi}_1^{21}$ $(\Delta p \geq \Delta p_{доп})$	$\hat{\pi}_1^{31}$ $(U_{зар} \in [0.5; 1.8])$	$\hat{\pi}_1^{41}$ $(U_{разр} \geq 0.5)$	$\hat{\pi}_1^{51}$ $(t \leq t_{max})$	$\hat{\pi}_1^{61}$ $(I > 0)$
	$\boldsymbol{s_1^1}$	$\hat{\pi}_1^{11}$ $(p \geq p_{min})$	$\hat{\pi}_1^{21}$ $(\Delta p \geq \Delta p_{доп})$	$\hat{\pi}_1^{32}$ $(U_{зар} \geq 2)$	$\hat{\pi}_1^{42}$ $(U_{разр} \leq 0.5)$	$\hat{\pi}_1^{52}$ $(t \geq t_{max})$	$\hat{\pi}_1^{61}$ $(I > 0)$
	$\boldsymbol{s_1^2}$	$\hat{\pi}_1^{11}$	$\hat{\pi}_1^{22}$ $(\Delta p \leq \Delta p_{доп})$	$\hat{\pi}_1^{31}$ $(U_{зар} \in [0.5; 1.8])$	$\hat{\pi}_1^{42}$ $(U_{разр} \leq 0.5)$	$\hat{\pi}_1^{51}$ $(t \leq t_{max})$	$\hat{\pi}_1^{61}$ $(I > 0)$
	$\boldsymbol{s_1^3}$	$\hat{\pi}_1^{12}$ $(p \leq p_{min})$	$\hat{\pi}_1^{21}$ $(\Delta p \geq \Delta p_{доп})$	$\hat{\pi}_1^{33}$ $(U_{зар} \leq 0.5)$	$\hat{\pi}_1^{42}$ $(U_{разр} \leq 0.5)$	$\hat{\pi}_1^{51}$ $(t \leq t_{max})$	$\hat{\pi}_1^{61}$ $(I > 0)$
	$\boldsymbol{s_1^4}$	$\hat{\pi}_1^{11}$ $(p \geq p_{min})$	$\hat{\pi}_1^{21}$ $(\Delta p \geq \Delta p_{доп})$	$\hat{\pi}_1^{32}$ $(U_{зар} \geq 2)$	$\hat{\pi}_1^{41}$ $(U_{разр} \geq 0.5)$	$\hat{\pi}_1^{51}$ $(t \leq t_{max})$	$\hat{\pi}_1^{62}$ $(I = 0)$

Таблица 3

Диагностическая модель аккумуляторной батареи

		$\boldsymbol{\Pi_1}$	$\boldsymbol{S_i^*}$		
		$\boldsymbol{\pi_1^1}$	$\boldsymbol{S_1}$	$\boldsymbol{S_2}$	$\boldsymbol{S_3}$
$\boldsymbol{S_1}$	$\boldsymbol{s_0^0}$	$\hat{\pi}_0^{11}$ $(\Delta \leq 25\%)$	s_1^0	s_2^0	s_3^0
	$\boldsymbol{s_0^1}$	$\hat{\pi}_0^{12}$ $(\Delta \geq 25\%)$	s_1^0	s_2^0	s_3^0
	$\boldsymbol{s_0^2}$	$\hat{\pi}_0^{11}$ $(\Delta \leq 25\%)$	$s_1^1 \vee s_1^2 \vee s_1^3 \vee s_1^4$	s_2^0	s_3^0
	$\boldsymbol{s_0^3}$	$\hat{\pi}_0^{11}$ $(\Delta \leq 25\%)$	s_1^0	$s_2^1 \vee s_2^2 \vee s_2^3 \vee s_2^4$	s_3^0
	$\boldsymbol{s_0^4}$	$\hat{\pi}_0^{11}$ $(\Delta \leq 25\%)$	s_1^0	s_2^0	$s_3^1 \vee s_3^2 \vee s_3^3 \vee s_3^4$

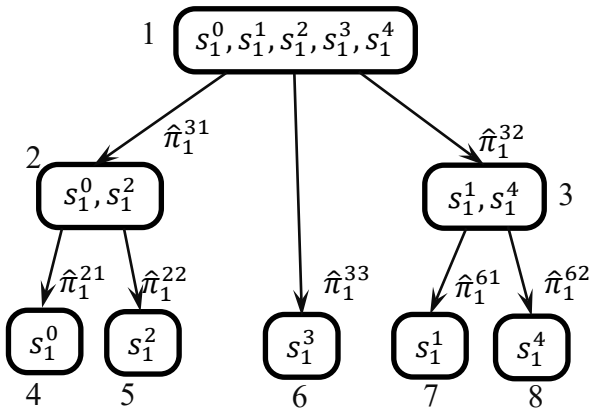

Рис. 2. Граф программы контроля НВА

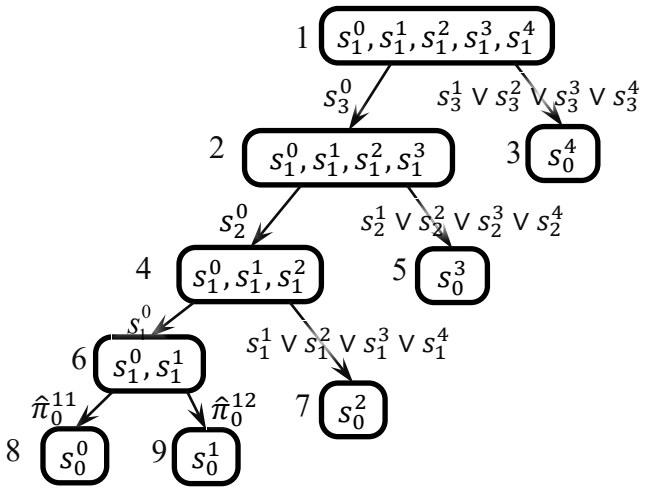

Рис. 3. Граф программы контроля аккумуляторной батареи

Программы контроля НВА и аккумуляторной батареи представлены на рисунках 2 и 3 в виде графов, узлы которых представляют собой фазовые состояния процесса контроля. Состояния пронумерованы для их дальнейшего использования в процессе синтеза конечного автомата.

Процесс контроля качества аккумуляторной батареи описывается совокупностью четырех автоматов в виде (5)

$$A = \{a_0, a_1, a_2, a_3\}.$$

Функции переходов и выходов для автомата a_0 представлены в таблице 4, для автомата a_1 — в таблице 5. Модели автоматов a_2и a_3 аналогичны a_1.

Процесс контроля при этом можно описать следующим образом: конечный автомат аккумуляторной батареи по мере необходимости дает команды на инициацию процесса контроля НВА. Образ состояния НВА нижнего уровня включает в себя только исходы проверок свойств НВА. При завершении контроля НВА (переходе автомата в конечное состояние) его состояние поступает на вход автомата аккумуляторной батареи, переводя его в следующее состояние. Окончание процесса контроля характеризуется нахождением всех автоматов на всех уровнях в их конечных состояниях, т. е. определенными состояниями для каждой бортовой системы.

Таблица 4

Функции переходов и выходов a_0

	Z_0								
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
Функция переходов									
s_3^0	2	–	–	–	–	–	–	–	–
$s_3^1 \vee s_3^2 \vee s_3^3 \vee s_3^4$	3	–	–	–	–	–	–	–	–
s_2^0	–	4	–	–	–	–	–	–	–
$s_2^1 \vee s_2^2 \vee s_2^3 \vee s_2^4$	–	5	–	–	–	–	–	–	–
s_1^0	–	–	–	6	–	–	–	–	–
$s_1^1 \vee s_1^2 \vee s_1^3 \vee s_1^4$	–	–	–	7	–	–	–	–	–
$\hat{\pi}_0^{11}$	–	–	–	–	–	8	–	–	–
$\hat{\pi}_0^{12}$	–	–	–	–	–	9	–	–	–
Функция выходов									
Выход a_0	a_1	a_2	–	a_3	–	π_1^1	–	–	–

Таблица 5

Функции переходов и выходов a_1

	Z_1							
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**
Функция переходов								
$\hat{\pi}_1^{31}$	2	–	–	–	–	–	–	–
$\hat{\pi}_1^{32}$	3	–	–	–	–	–	–	–
$\hat{\pi}_1^{33}$	6	–	–	–	–	–	–	–
$\hat{\pi}_1^{21}$	–	4	–	–	–	–	–	–
$\hat{\pi}_1^{22}$	–	5	–	–	–	–	–	–
$\hat{\pi}_1^{61}$	–	–	7	–	–	–	–	–
$\hat{\pi}_1^{62}$	–	–	8	–	–	–	–	–
Функция выходов								
Выход a_1	π_1^3	π_1^2	π_1^6	–	–	–	–	–

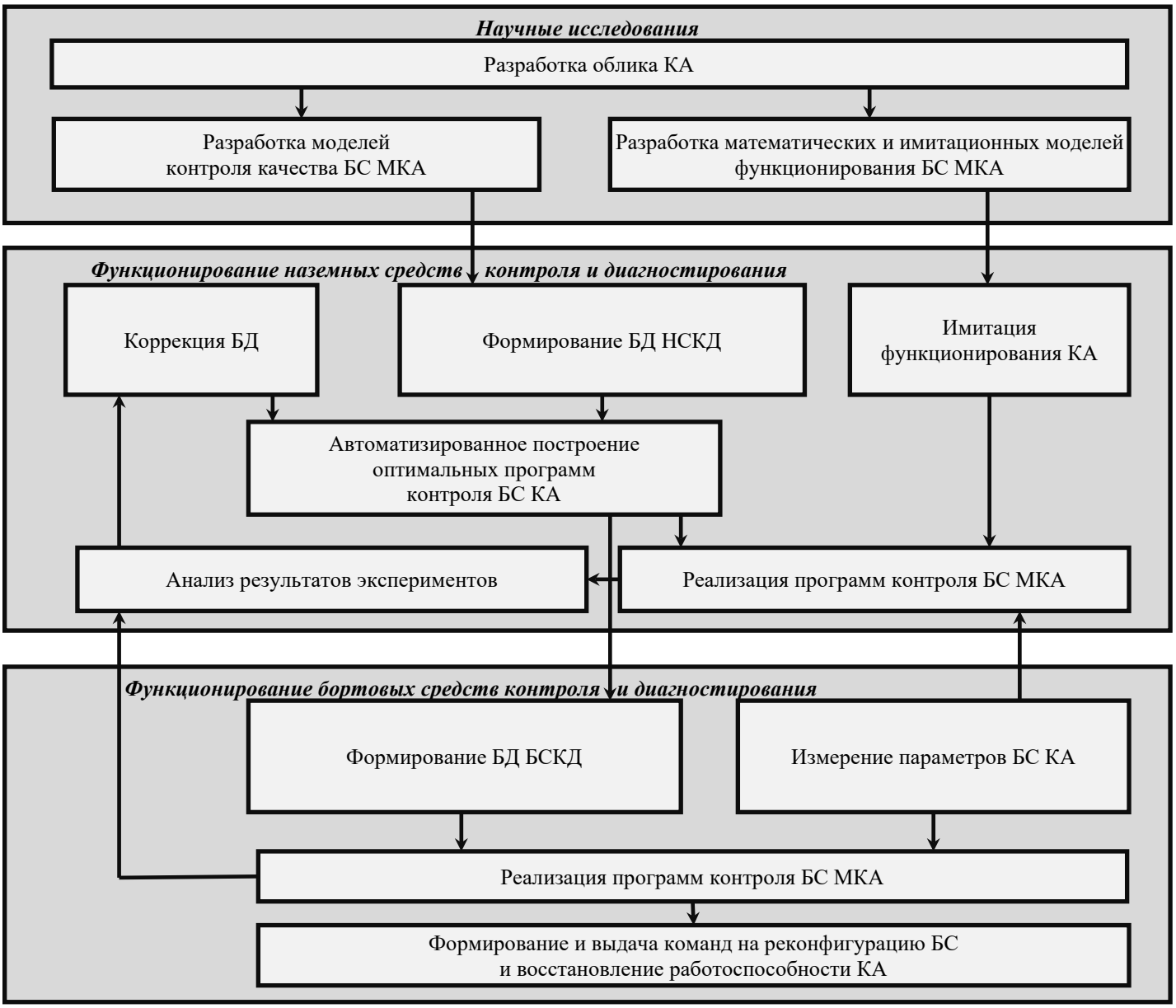

Рис. 4. Методика контроля качества БС КА

Методика контроля качества бортовых систем

Одной из современных тенденций развития технологий создания КА является применение «протолетного» подхода [12, 13]. Опыт его внедрения [14] показал отсутствие повышения риска возникновения отказов КА в полете при условии постоянного контроля и глубокого анализа показателей качества БС КА. В связи с этим в настоящее время проводятся исследования, направленные на совершенствование функций контроля и анализа показателей качества при применении «протолетного» подхода [15]. Реализация данных функций возможна только при параллельном функционировании наземного модуля и БСКД. Методика применения специального программного обеспечения наземного функционального модуля для выполнения задач реализации «протолетного» подхода при создании и эксплуатации КА, а также восстановления работоспособности КА при возникновении аварийных и критических ситуаций в период их эксплуатации представлена на рисунке 4.

На этапе научных исследований после разработки облика КА разрабатываются математические модели функционирования БС КА и КА в целом, а также модели контроля качества e_i БС КА.

На основании математических моделей функционирования БС КА разрабатывается имитатор МКА, который представляет собой имитационную модель КА, предназначенную для решения следующих задач:

– проверки правильности критериев оценивания качества БС;

– проведение статистических испытаний с целью обоснования критериев, обеспечивающих заданный уровень достоверности принятия решения;

– моделирование процесса функционирования КА в процессе полета (метод моделирования параллельно с функционированием системы [16]);

– моделирование нештатных и аварийных ситуаций, возникающих в процессе полета КА, отработка методов их парирования.

На основании моделей контроля качества разрабатывается база данных НСКД, основу которой составляет модель BS. С использованием базы данных формируется оптимальная по заданным критериям программа контроля в виде иерархического конечного автомата $BSCD$, которая реализуется с применением имитатора КА. После обоснования критериев оценивания качества методом статистических испытаний формируется база данных БСКД.

Основу базы данных БСКД составляют программы контроля, представленные в форме конечных автоматов a_i. БСКД реализует программы контроля, используя в качестве входной информации данные системы ТМИ и ИОК. По результатам контроля осуществляется прогноз полного отказа МКА и формируются данные для переключения резервных блоков. Вся информация о работе БСКД отправляется на пункт управления, где сравнивается с результатами работы НСКД. По результатам анализа информации осуществляется коррекция баз данных наземной, а после проведения необходимых испытаний — бортовой системы контроля и диагностирования.

Заключение

Применение иерархии автоматов для управления процессом контроля открывает возможности для унификации алгоритмов управления бортовыми и наземными средствами контроля и диагностирования и применения уже готовых программ контроля в составе последних. Кроме того, расширение агрегированной модели состояниями вложенных систем позволяют создавать необходимую конфигурацию процесса контроля при помощи оператора сопряжения *r*. Изменяя соответствующие связи, можно варьировать глубину контроля, добавлять (удалять) из процесса отдельные блоки или системы, что особенно актуально в условиях автономного функционирования БС КА. Реализация разработанной модели по предлагаемой методике позволит в полной мере использовать преимущества «протолетного» подхода к созданию КА.

Литература

1. Петухов Г. Б. Методологические основы внешнего проектирования целенаправленных процессов и целеустремленных систем / Г. Б. Петухов, В. И. Якунин. — М.: АСТ, 2006. — 504 с.

2. Барановский А. М. Система контроля и диагностирования бортового оборудования малого космического аппарата / А. М. Барановский, А. Е. Привалов // Известия высших учебных заведений. Приборостроение. 2009. Т. 52, № 4. С. 51–56.

3. Привалов А. Е. Модель автономной системы контроля и диагностирования бортового оборудования космического аппарата // Труды Военно-космической академии имени А. Ф. Можайского. 2010. Вып. 626. С. 71–77.

4. Дмитриев А. К. Принципы алгебраического агрегирования в задачах диагностирования сложных технических систем // Известия высших учебных заведений. Приборостроение.1997. Т. 40, № 8. С. 5–13.

5. Алгоритм построения гибкой программы диагностирования технического объекта по критерию ценности получаемой информации / Е. В. Копкин, В. А. Чикуров, В. В. Алейник, О. Г. Лазутин // Труды СПИИРАН. 2015. Вып. 4 (41). С. 106–130.

6. Мануйлов Ю. С. Синтез программы диагностирования технического объекта при оптимальном согласовании достоверности и стоимости получаемой информации / Ю. С. Мануйлов, А. Н. Кравцов // Информация и космос. 2009. № 3. С. 24–29.

7. Дмитриев А. К. Идентификация и техническая диагностика: Учебник для вузов / А. К. Дмитриев, Р. М. Юсупов. — Л.: Министерство обороны СССР, 1987. — 521 с.

8. Дмитриев А. К. Построение информационно-поисковой системы по критерию максимума полезности получаемой информации / А. К. Дмитриев, Е. В. Копкин // Авиакосмическое приборостроение. 2003. № 6. С. 46–51.

9. Кузьмин Е. В. Иерархическая модель автоматных программ // Моделирование и анализ информационных систем. 2006. Т. 13, № 1. С. 27–34.

10. Прангишвили И. В. Системный подход и общесистемные закономерности / И. В. Прангишвили; Российская академия наук; Институт проблем управления имени В. А. Трапезникова. — М.: НПО СИНТЕГ, 2000. — 528 с. — (Серия «Системы и проблемы управления»).

11. Ратушняк А. И. Введение в летную эксплуатацию бортовых систем электроснабжения: Учебное пособие. Часть 2. Никель-водородные батареи в бортовых аккумулирующих подсистемах / А. И. Ратушняк, Г. Б. Стеганов. —

СПб.: Военно-космическая академия имени А. Ф. Можайского, 2004. — 45 с.

12. Куреев В. Д. Перспективы реализации «протолетного» подхода при наземной отработке наноспутников / В. Д. Куреев, С. В. Павлов, Ю. А. Соколов // Известия вузов. Приборостроение. 2016. Т. 59, № 6. С. 477–481.

13. NASA Technical Standard NASA-STD-5001B. Structural Design and Test Factors of Safety for Spaceflight Hardware. Approved Aug 06, 2014. — Washington, DC, 2014. — 25 p. URL: http://standards.nasa.gov/sites/default/files/nasa-std-5001b.pdf (дата обращения 10.06.2020).

14. Coan M. R. Internal NASA Study: NASA's Protoflight Research Initiative (NASA_NTRS_Archive_20150014585) / M. R. Coan, S. R. Hirshorn, R. Moreland // NASA Technical Report Server. Published at January 1, 2015. — 18 p. URL: http://ntrs.nasa.gov/citations/20150014585 2019-08-31T06:49:25 00:00Z (дата обращения 10.06.2020).

15. О научно-технической программе Союзного государства «Разработка комплексных технологий создания материалов, устройств и ключевых элементов космических средств и перспективной продукции других отраслей» («Технология-СГ»): постановление Совета Министров Союзного государства Беларуси и России от 12.05.2016 № 17. URL: http://base.garant.ru/71405416 (дата обращения 01.11.2019).

16. Анализ современного состояния и тенденции развития имитационного моделирования в Российской Федерации (по материалам конференций «Имитационное моделирование. Теория и практика» (ИММОД)) / А. М. Плотников, Ю. И. Рыжиков, Б. В. Соколов, Р. М. Юсупов // Труды СПИИРАН. 2013. Вып. 2 (25). С. 42–112.

Model and Technique of Quality Control of Spacecraft On-Board Systems

PhD A. M. Baranovsky, A.V. Kikot, PhD E. N. Shapovalov
Software Research Institute
St. Petersburg, Russia
bamvka@mail.ru, a.v.kikot@yandex.ru, henya56@mail.ru

Abstract. **The article proposes a model of quality control of spacecraft on-board systems, based on the hierarchy of finite state machines. A model of on-board systems, based on the principles of algebraic aggregation, is proposed as the initial data for the construction of the model. This model distinguished by the expansion of the on-board system state image via states of the subsystems. The developed model is built on the modular principle, which allows you to create and quickly change the configuration of the control process via varying the depth of control or the composition of the tested systems. A technique of quality control of on-board systems based on the hierarchy of finite state machines has been proposed. It will allow to fully use advantage of the «proto-flight» approach to the creation of spacecraft.**

Keywords: **spacecraft, hierarchical of finite state machine, quality control, on-board system.**

REFERENCES

1. Petukhov G. B., Yakunin V. I. Methodological foundations of external design of purposeful processes and purposeful systems [Metodologicheskie osnovy vneshnego proektirovaniya tselenapravlennykh protsessov i tseleustremlennykh sistem], Moscow, AST Publishers, 2006, 504 p.

2. Baranovsky A. M., Privalov A. E. Onboard Monitoring and Diagnostic System of Small Space Vehicles [Sistema kontrolya i diagnostirovaniya bortovogo oborudovaniya malogo kosmicheskogo apparata], *Journal of Instrument Engineering [Izvestiya vysshikh uchebnykh zavedeniy. Priborostroenie]*, 2009, Vol.52, No. 4, Pp. 51–56.

3. Privalov A. E. Model of an Autonomous System for Monitoring and Diagnostics of Spacecraft Onboard Equipment [Model' avtonomnoy sistemy kontrolya i diagnostirovaniya bortovogo oborudovaniya kosmicheskogo apparata], *Proceedings of the Mozhaisky Military Space Academy [Trudy Voenno-kosmicheskoy akademii imeni A. F. Mozhayskogo]*, 2010, Is. 626. Pp. 71–77.

4. Dmitriev A. K. Principles of Algebraic Aggregation in the Problems of Diagnosing Complex Technical Systems [Printsipy algebraicheskogo agregirovaniya v zadachakh diagnostirovaniya slozhnykh tekhnicheskikh sistem], *Journal of Instrument Engineering [Izvestiya vysshikh uchebnykh zavedeniy. Priborostroenie]*, 1997, Vol. 40, No. 8, Pp. 5–13.

5. Kopkin E.V., Chikurov V. A., Aleynik V. V., Lazutin O. G. Algorithm for Constructing a Flexible Program for Technical Object Diagnosing on the Criterion of Received Information Value [Algoritm postroeniya gibkoy programmy diagnostirovaniya tekhnicheskogo ob"ekta po kriteriyu tsennosti poluchaemoy informatsii], *SPIIRAS Proceedings [Trudy SPIIRAN]*, 2015, Is. 4 (41), Pp. 106–130.

6. Manuilov Yu. S., Kravtsov A. N. Synthesis of Diagnostic Program for Technical Object Accompanied by Optimal Coordination of Accommodation Authenticity and Cost of Received Information [Sintez programmy diagnostirovaniya tekhnicheskogo ob"ekta pri optimal'nom soglasovanii dostovernosti i stoimosti poluchaemoy informatsii], *Information and Space [Informatsiya i kosmos]*, 2009, No. 3, Pp. 24–29.

7. Dmitriev A. K., Yusupov R. M. Identification and technical diagnostics: Textbook for universities [Identifikatsiya i tekhnicheskaya diagnostika: Uchebnik dlya vuzov], Leningrad, Ministry of Defense of USSR, 1987, 521 p.

8. Dmitriev A. K., Kopkin E. V. The Construction of an Information Retrieval System According to the Criterion of Maximum Usefulness of the Information Obtained [Postroenie informatsionno-poiskovoy sistemy po kriteriyu maksimuma poleznosti poluchaemoy informatsii], Aerospace Instrument-Making [Aviakosmicheskoe priborostroenie], 2003, No. 6, Pp. 46–51.

9. Kuzmin E. V. A Hierarchical Model of Automaton Programs [Ierarkhicheskaya model' avtomatnykh programm], *Modeling and Analysis of Information Systems [Modelirovanie i analiz informatsionnykh sistem]*, 2006, Vol. 13, No. 1, Pp. 27–34.

10. Prangishvili I. V. System Approach and System-Wide Patterns [Sistemnyy podkhod i obshchesistemnye zakonomernosti], Moscow, NPO SINTEG, 2000, 528 p.

11. Ratushnyak A. I., Steganov G. B. Introduction to flight operation of on-board power supply systems: Study guide. Part 2. Nickel-hydrogen batteries in onboard storage subsystems [Vvedenie v letnuyu ekspluatatsiyu bortovykh sistem elektrosnabzheniya: Uchebnoe posobie. Chast' 2. Nikel'-vodorodnye batarei v bortovykh akkumuliruyushchikh podsistemakh], St. Petersburg, A. F. Mozhaisky Military Space Academy, 2004, 45 p.

12. Kureev V. D., Pavlov S. V., Sokolov Yu. A. Prospects of Realization of Proto-Flight Approach in Nanosatellite Ground Testing [Perspektivy realizatsii «protoletnogo» podkhoda pri nazemnoy otrabotke nanosputnikov], *Journal of Instrument Engineering [Izvestiya vysshikh uchebnykh zavedeniy. Priborostroenie]*, 2016, Vol. 59, No. 6, Pp. 477–481.

13. NASA Technical Standard NASA-STD-5001B. Structural Design and Test Factors of Safety for Spaceflight Hardware. Approved Aug 06, 2014. Washington, DC, 2014, 25 p. Available at: http://standards.nasa.gov/sites/default/files/nasa-std-5001b.pdf (accessed 10 Jun 2020).

14. Coan M. R., Hirshorn S. R., Moreland R. Internal NASA Study: NASA's Protoflight Research Initiative (NASA_NTRS_Archive_20150014585), *NASA Technical Report Server*, 18 p. Published at January 1, 2015.
Available at: http://ntrs.nasa.gov/citations/20150014585 2019-08-31T06:49:25 00:00Z (accessed 10 Jun 2020).

15. About the Scientific and Technical Program of the Union State «Development of Complex Technologies of Creation of Materials, Devices and Key Elements of Space Means and Perspective Products of Other Industries» («Technology-SG»): Resolution of Council of Ministers of the Union State of Belarus and Russia [O nauchno-tekhnicheskoy programme Soyuznogo gosudarstva «Razrabotka kompleksnykh tekhnologiy sozdaniya materialov, ustroystv i klyuchevykh elementov kosmicheskikh sredstv i perspektivnoy produktsii drugikh otrasley» («Tekhnologiya-SG»): postanovlenie Soveta Ministrov Soyuznogo gosudarstva Belarusi i Rossii] from May 12, 2016, No. 17. Available at: http://base.garant.ru/71405416 (accessed 01 Nov 2019).

16. Plotnikov A. M., Ryzhikov Yu. I., Sokolov B. V., Yusupov R. M. The Analysis of Current Status and Development Trends of Simulation in the Russian Federation (on Conference Proceedings «Simulation. Theory and Practice» (IMMOD)) [Analiz sovremennogo sostoyaniya i tendentsii razvitiya imitatsionnogo modelirovaniya v Rossiyskoy Federatsii (po materialam konferentsiy «Imitatsionnoe modelirovanie. Teoriya i praktika» (IMMOD))], *SPIIRAS Proceedings [Trudy SPIIRAN]*, 2013, Is. 2(25), Pp. 42–112.

Использование тестовых информационно-технических воздействий для аудита защищенности информационных систем железнодорожного транспорта

Г. Е. Смирнов
ООО «Корпорация «Интел групп»
Санкт-Петербург, Россия
science.cybersec@yandex.ru

д.т.н. С. И. Макаренко
Санкт-Петербургский Федеральный
исследовательский центр Российской академии наук,
Санкт-Петербургский государственный
электротехнический университет «ЛЭТИ»
имени В. И. Ульянова (Ленина)
Санкт-Петербург, Россия
mak-serg@yandex.ru

Аннотация. **Рассмотрены основные информационные и автоматизированные системы железнодорожного транспорта (ЖТ). Показано, что эти системы являются объектами критической информационной инфраструктуры. В соответствии с законодательством Российской Федерации такие объекты должны быть подключены к центрам Государственной системы обнаружения, предупреждения и ликвидации последствий компьютерных атак (ГосСОПКА), которые осуществляют аудит состояния их информационной безопасности (ИБ). Показано, что существующие центры ГосСОПКА, осуществляющие аудит состояния ИБ информационных систем ЖТ не предусматривают такой функциональности, как оценку защищенности информационных систем тестовыми информационно-техническими воздействиями (ИТВ), аналогичными ИТВ, которые прогнозируются к применению злоумышленниками. Обоснована целесообразность применения такой разновидности аудита, а также предложен вариант совершенствования типовой архитектуры центра ГосСОПКА за счет включения в его состав автоматизированного комплекса тестирования защищенности информационных систем ЖТ. Представлены предложения по составу и порядку функционирования такого автоматизированного комплекса тестирования.**

Ключевые слова: **информационная безопасность, аудит, тестирование, информационно-техническое воздействие, критическая информационная инфраструктура, железнодорожный транспорт.**

Введение

В 2017 году в России принят федеральный закон №187-ФЗ «О безопасности критической информационной инфраструктуры Российской Федерации» [1]. Данный закон устанавливает перечень объектов, относящихся к критической информационной инфраструктуре (КИИ) РФ, а также обязует владельцев объектов КИИ разработать комплекс мер, направленных на обеспечение их информационной безопасности (ИБ). При этом к КИИ отнесен и железнодорожный транспорт (ЖТ), в связи с чем актуальным является формирование новых предложений по повышению полноты аудита ИБ информационных систем (ИС) ЖТ как объекта КИИ.

Целью статьи является обоснование такого перспективного направления аудита ИБ, как тестирование ИС ЖТ тестовыми информационными-техническими воздействиями (ИТВ), которые соответствуют предполагаемым ИТВ злоумышленника. Такое тестирование, по замыслу авторов, дополнит стандартные мероприятия аудита ИС ЖТ и повысит полноту оценки ИБ. При этом отметим, что стандартные мероприятия аудита ИС ЖТ, как правило, не включают практические элементы проверки состояния ИБ и проводятся путем проверки соответствия спецификациям и требованиям руководящих документов по обеспечению ИБ.

Анализ ИС ЖТ как объекта критической информационной инфраструктуры и задач обеспечения ее защищенности

Анализ работ по составу, структуре и функционированию ИС ЖТ [2–4], а также работ по оценке ИБ ИС ЖТ [5–8], показывает следующее. Вопросами обеспечения ИБ ИС ЖТ занимаются следующие отечественные специалисты: Ададуров С. Е., Диасамидзе С. В., Корниенко А. А., Сидак А. А., Котенко И. В., Саенко И. Б., Чечулин А. А., Чернов А. В., Бутакова М. А., а также другие ученые. В их работах рассмотрены нижеуказанные особенности ИС ЖТ, значимые для обеспечения их ИБ как объекта КИИ РФ.

ИС ЖТ относится к классу больших корпоративных систем и предназначена для решения как информационных задач, так и задач управления отраслью. Главная цель применения ИС ЖТ состоит в информационном обеспечении технологических процессов и автоматизации принятия решений в сфере ЖТ в интересах достижения максимальной эффективности его работы в условиях рыночной экономики.

ИС ЖТ представляется в виде двухуровневой структуры. Первый уровень — обеспечивающий, представлен информационной средой и инфраструктурой информатизации, второй уровень — прикладной, реализуется путем использования информационных технологий (ИТ), объединенных в ИТ-комплексы, решающих конкретные задачи управления и автоматизации функций ЖТ.

Исследование выполнено в рамках госбюджетной темы НИР СПИИРАН № 0073-2019-0004.

Информационная среда — информация, реализованная в системе баз данных (БД), которая обеспечивает функционирование объектов, органов управления и отдельных пользователей ЖТ. Информационная среда формирует единое информационное пространство (ЕИП), в котором все абоненты и пользователи ЖТ обеспечены необходимой им информацией.

Инфраструктура информатизации ЖТ включает в себя:

1) главный вычислительный центр (ГВЦ) ЖТ, объединяющий и поддерживающий БД для проведения общесетевой маркетинговой, финансовой и экономической деятельности и управления перевозочным процессом;

2) информационно-вычислительные центры (ИВЦ) ЖТ на дорогах, реализующие комплексы информационных услуг для управлений и отделений дорог;

3) сети связи и телекоммуникаций, устройства автоматического съема информации с подвижного состава, вычислительное оборудование, обеспечивающее выполнение операций формирования, сбора, передачи, хранения, обработки и представления информации.

Отдельные ИТ, обеспечивающие автоматизацию основных функций ЖТ, составляют ИТ-комплексы:

1) управление перевозочным процессом;

2) управление маркетингом, экономикой и финансами;

3) управление инфраструктурой ЖТ;

4) управление непроизводственной сферой.

Рассмотрим эти комплексы более подробно.

1. ИТ-комплекс управления перевозочным процессом обеспечивает информационное сопровождение в области грузовых и пассажирских перевозок. Основными функциями по управлению грузовыми перевозками являются организация поездо- и грузопотоков на сети, диспетчерское управление поездной работой, управление локомотивными и вагонными парками, грузовой и коммерческой работой, обслуживание грузовой клиентуры, разработка графика движения поездов, норм эксплуатационной работы, планирование перевозок и прочее. Основными функциями по управлению пассажирскими перевозками являются организация обслуживания пассажиров и информационно-справочный сервис, планирование пассажирских перевозок в международном и внутридорожном сообщении, управление нормативами, тарифами внутренних и международных перевозок, организация эксплуатации и ремонта парка пассажирских вагонов, управление багажными и почтовыми перевозками, организация билетно-кассовых операций и др. В рамках этого ИТ-комплекса функционируют:

– автоматизированная система оперативного управления перевозками (АСОУП) — основной элемент ИТ-комплекса управления перевозочным процессом;

– система резервирования и продажи билетов («Экспресс-2»);

– единые центры диспетчерского управления (ЕЦДУ);

– система учета, контроля дислокации, анализа использования и регулирования вагонного парка (ДИСПАРК);

– автоматизированная система контроля за использованием и продвижением контейнеров (ДИСКОН);

– автоматизированная система фирменного транспортного обслуживания (АКС ФТО);

– автоматизированные системы управления сортировочными (АСУ СС) и грузовыми (АСУ ГС) станциями и контейнерными пунктами (АСУ КП);

– автоматизированная система централизованной подготовки и оформления перевозочных документов (ЭТРАН);

– сетевая интегрированная Российская информационно-управляющая система (СИРИУС) и др.

2. ИТ-комплекс управления маркетингом, экономикой и финансами охватывает финансовую деятельность, бухгалтерский учет, маркетинговую деятельность и тарифную политику, управление развитием отрасли ЖТ, технической политикой и научно-исследовательскими и опытно-конструкторскими работами, нормативно-правовую работу, управление эксплуатационными расходами и др. ИТ этого комплекса ориентированы на формирование заказов, увеличение доходов, укрепление конъюнктурного положения за счет сохранения и увеличения доли ЖТ на транспортном рынке страны, на стабильное обеспечение денежных и платежных ресурсов, минимизацию затрат, на совершенствование экономической работы и инвестиционной политики. В рамках комплекса функционируют и внедряются ИТ управления финансовой деятельностью, ресурсами, способы расчетов за грузовые перевозки, взаиморасчетов за пользование вагонами и др. Основу этого ИТ-комплекса составляет единый комплекс автоматизированной системы управления финансовой деятельностью (ЕК АСУФР).

3. ИТ-комплекс управления инфраструктурой ЖТ представлен базовыми информационными технологиями, охватывающими управление эксплуатационной работой пассажирского хозяйства, хозяйств пути и сооружений, информатизации и связи, хозяйства энергоснабжения, локомотивного и вагонного хозяйств, управление проектированием и капитальным строительством объектов инфраструктуры, управление ремонтно-восстановительными работами и работами в чрезвычайных условиях, управление промышленностью ЖТ, материально-техническим снабжением и т. д. В составе этого ИТ-комплекса функционируют различные автоматизированные системы управления технологическими процессами (АСУ ТП): управления путевым хозяйством, устройствами энергоснабжения, сигнализации, средствами информатизации и связи.

4. ИТ-комплекс информационных технологий управления непроизводственной сферой железнодорожного транспорта представляет собой совокупность функций, обеспечивающих управление персоналом, учебными заведениями, жилищно-коммунальным хозяйством, рабочим снабжением, здравоохранением.

Основными факторами, актуализирующими значимость вопросов обеспечения ИБ применительно к ИС ЖД, являются следующие [7]:

– интеграция в единые ИТ-комплексы подавляющего числа критических функций, связанных с управлением движением поездов и жизнедеятельности ЖТ;

– постоянное усложнение программного обеспечения (ПО) и оборудования, используемых в ИТ-комплексах управления ЖТ.

– существующая практика удаленной настройки и технического обслуживания элементов ИС ЖТ, осуществле-

ния разработчиками и поставщиками оборудования, входящего в состав элементов информационной инфраструктуры железнодорожного транспорта.

– интенсивное совершенствование потенциальными злоумышленниками средств и способов информационно-технического воздействия (ИТВ), методов социальной инженерии для нанесения ущерба, а также участившиеся попытки их применения в противоправных целях и конкурентной борьбе.

– риск сокрытия попыток или фактов нарушения штатного функционирования ИС ЖТ со стороны эксплуатирующих подразделений.

– временное вынужденное привлечение к созданию элементов ИТ-комплексов ЖТ, в том числе АСО УП и различных АСУ ТП, производителей и поставщиков программно-аппаратных средств обработки, хранения и передачи информации и применение неконтролируемых программно-аппаратных решений.

Помимо вышеуказанных факторов, отметим следующее. ЖТ является одним из ключевых элементов транспортной инфраструктуры РФ, обеспечивая до 88 % грузооборота страны (для сравнения: доля автомобильного транспорта составляет 4 %, а водного – 8 %) [2]. В связи с этим ЖТ является одной из основных целей для профессиональных нарушителей — сил информационных операций («кибервойска») недружественных стран, при ведении информационного противоборства. В связи с этим, при обострении геополитической обстановки в мире, информационная инфраструктура и ИС ЖТ РФ могут оказаться объектом воздействия не только непрофессиональных нарушителей, но и профессиональных действий кибервойск [8].

В рамках формирования нормативно-методической базы обеспечения ИБ ИС ЖТ российскими специалистами в 2015 г. в инициативном порядке были разработаны и представлены на рассмотрение в международную рабочую группу COLPOFER (международная организация, созданная в 1981 году, объединяющая службы безопасности европейских железных дорог и подразделения полиции на ЖТ) два проекта нормативных документов (НМД) [7]:

– НМД-1: «Основные положения защиты информационной инфраструктуры железнодорожного транспорта от компьютерных атак» («Guidelines for protection of railway transport information infrastructure against cyber attacks»);

– НМД-2: «Основные положения порядка использования сил и средств предупреждения и обнаружения компьютерных атак на информационную инфраструктуру железных дорог» («Guidelines for using forces and tools to prevent and detect computer attacks against rail information infrastructure»).

НМД-1 является базовым рекомендательным документом, определяющим направления работ по созданию системы защиты ИС ЖТ от компьютерных атак и общие меры по их предупреждению, обнаружению, анализу и ликвидации последствий. В документе представлены исходные данные и основные направления решения указанных задач [7]:

– дана общая характеристика ИИ ЖТ, идентифицированы основные элементы ИИ ЖТ, подверженные компьютерным атакам, проведена типизация возможных компьютерных атак на элементы ИС ЖТ;

– определены основные принципы защиты ИС ЖТ от компьютерных атак;

– представлен облик системы защиты ИС ЖТ от компьютерных атак;

– определены этапы создания системы защиты ИС ЖТ от компьютерных атак и определены мероприятия по поддержке защищенности на требуемом уровне.

В НМД-2 раскрываются [7]:

– привлекаемые силы и средства, необходимые для защиты ИС ЖТ от компьютерных атак, порядок и основные принципы их использования;

– способы обнаружения и предупреждения компьютерных атак, характерные для элементов ИС ЖТ, с использованием таких компонентов, как ложные информационные системы и ресурсы, а также с использованием «традиционных» систем и средств защиты ИС от компьютерных атак;

– порядок использования дополнительных компонентов для обнаружения и предупреждения компьютерных атак.

Наличие подобных руководящих документов значительно упрощает аудит ИБ ИС ЖТ как объекта КИИ и формирует систему показателей оценки ИБ, объектов мониторинга, а также предполагаемые действия нарушителя.

Дальнейшее развитие НМД в области ИБ ЖТ произошло в 2017 г., когда в России был принят закон № 187-ФЗ «О безопасности критической информационной инфраструктуры Российской Федерации» [1]. Данный закон отнес ЖТ к КИИ и обязал разработать комплекс мер, направленных на обеспечение ИБ ИС ЖТ, а для аудита эффективности этих мер подключить ИС ЖТ к соответствующим центрам Государственной системы обнаружения, предупреждения и ликвидации последствий компьютерных атак (ГосСОПКА).

Анализ системы аудита защищенности ИС ЖТ в рамках системы ГосСОПКА

Для реализации аудита состояния ИБ и защиты КИИ в РФ с начала 2010-х годов были начаты работы по созданию системы ГосСОПКА, основанной на централизованном использовании взаимоувязанных систем обнаружения вторжений IDS (Intrusion Detection System), систем предотвращения вторжений IPS (Intrusion Prevention System), систем предотвращения утечек конфиденциальных данных DLP (Data Leak Prevention), а также систем управления инцидентами информационной безопасности SIEM (Security Information and Event Management).

В составе ГосСОПКА создается система государственных и частных центров, которые обслуживают субъекты КИИ. Такой центр берет на себя часть функций безопасности, необходимых для противодействия ИТВ на ИС субъектов КИИ. К таким функциям, как правило, относятся:

– выявление и анализ уязвимостей, обслуживаемых ИС, координация действий по устранению выявленных уязвимостей;

– анализ событий, регистрируемых компонентами обслуживаемых ИС и средств их защиты, для поиска признаков ИТВ, направленных на эти системы;

– координация действий по реагированию на обнаруженный ИТВ; а если атака привела к инциденту — по ликвидации последствий такого инцидента;

– расследование инцидентов и ретроспективный анализ ИТВ, которые не удалось предотвратить;

– информирование персонала обслуживаемых ИТВ, проведение киберучений.

Для выполнения вышеуказанных функций центры ГосСОПКА тесно интегрируются с защищаемыми ИС — они получают полные инвентаризационные данные ИС, контролируют их защищенность и анализируют события, регистрируемые их средствами защиты. При этом данные центры не заменяют собой собственные системы защиты ИС, так как владельцы объектов КИИ должны обеспечивать их ИБ самостоятельно, а центр ГосСОПКА своей деятельностью лишь компенсирует возможные ошибки.

В работе [9] проведен анализ роли ГосСОПКА в нормативных документах о безопасности КИИ. Показано, что владельцы значимых объектов КИИ обязаны выполнять требования ФСТЭК РФ по обеспечению безопасности этих объектов (ч. 3 ст. 9 ФЗ-187) и создавать системы защиты этих объектов (ст. 10 ФЗ-187). В соответствии с требованиями ФСТЭК (Приказ ФСТЭК России № 239 от 25.12.2017 «Об утверждении Требований по обеспечению безопасности значимых объектов критической информационной инфраструктуры РФ») в системе защиты должны быть реализованы базовые меры, многие из которых непосредственно направлены на противодействие ИТВ злоумышленников [10]:

– инвентаризация компонентов информационных систем и анализ их уязвимостей;

– контроль и анализ сетевого трафика;

– мониторинг безопасности;

– антивирусная защита;

– предотвращение вторжений;

– реагирование на инциденты и т.п.

При этом владелец имеет право самостоятельно решать, как именно будут реализованы эти меры защиты (ч. 1 ст. 9 ФЗ-187). Более того, эти меры защиты являются всего лишь базовыми, то есть необходимыми, но не достаточными для обеспечения безопасности объекта КИИ. В соответствии с существующими процедурами, владелец объекта КИИ должен самостоятельно провести анализ угроз, актуальных для объекта, самостоятельно определить, как должны быть реализованы базовые меры защиты, а если их окажется недостаточно для защиты от угроз — самостоятельно усилить базовые меры защиты или разработать дополнительные [9]. В таких условиях именно аудит ИБ является тем основным инструментом, который позволяет оценить уровень угроз для объекта КИИ и уровень его защищенности.

Ядром центра ГосСОПКА является SIEM-система (рис. 1). Именно на нее возлагаются основные задачи по аудиту ИБ — сбору данных о событиях в ИС ЖТ, их анализу и выявлению инцидентов. Вопросам повышения эффективности SEIM-систем при аудите состояния ИБ посвящены работы [11–16]. В них показано, что основными перспективными направлениями совершенствования SEIM-систем аудита ИБ ИС ЖТ являются:

– повышение полноты и своевременности сбора данных о событиях в элементах и подсистемах ИС ЖТ;

– повышение интеллектуальности обработки данных о событиях в элементах и подсистемах ИС ЖТ, в том числе за счет использования технологий многомерного корреляционного анализа и технологий искусственного интеллекта;

– формирование положительной обратной связи в системе за счет своевременного обнаружения ИТВ и оперативного формирования сценариев защиты от него;

– моделирование действий злоумышленников с автоматической генерацией на основе результатов моделирования как высоковероятных сценариев действий злоумышленников, так и адекватных и эффективных сценариев защиты;

– повышение интеллектуальности человеко-машинного интерфейса системы в части адаптации визуализации представления информации о событиях в системе по отношению к системе зрительного восприятия человека-оператора с целью повышения информативности и эргономичности системы.

Вместе с тем вышеуказанные направления повышения эффективности SIEM-систем в составе центров ГосСОПКА не устраняют один из главных, по мнению авторов, недостатков этих систем — центры ГосСОПКА по своему принципу функционирования ориентированы на сбор данных об уже произошедших инцидентах ИБ, а также на сбор доказательств для оперативного исследования этих инцидентов.

Такая ориентированность центров ГосСОПКА на фиксирование инцидентов постфактум обусловлена общими недостатками существующих подходов к аудиту состояния ИБ ИС.

Проведенный анализ современных теоретических подходов в области аудита ИБ, представленный в работах [17, 18], показал, что задачей аудита является проверка и оценивание ИС на соответствие критериям, которые определяют требования к уровню ИБ. В настоящее время в теории аудита ИБ сложилась ситуация, при которой большинство работ в этой области ориентировано на исследование экспертного аудита и оценки соответствия преимущественно на основе моделей анализа рисков, либо на основе анализа стандартов ИБ. При этом тестирование и, в особенности, тестирование специальными ИТВ, является недостаточно изученной областью аудита. Имеются отдельные работы, например [19–21], которые посвящены такому типу тестирования как «тест на проникновение», однако данные работы носят в большей степени практический, нежели теоретический характер.

Как показано в [17, 18], тестирование является более гибким инструментом аудита чем, например, мероприятия оценки соответствия, так как его проведение не ограничено рамками действующих стандартов и регламентов. Это позволяет выбирать более широкий диапазон средств и способов тестирования, а также быть более избирательным в направлении достижения цели аудита. Например, проводить тестовое исследование ИС КИИ к угрозам и выявлять уязвимости, еще не описанные в базах угроз и уязвимостей. При тестировании ИС КИИ целесообразно сформировать и придерживаться системного подхода к проведению тестирования специальными средствами и способами ИТВ. При этом такое тестирование необходимо рассматривать как основную форму контроля устойчивости объектов КИИ к целенаправленным ИТВ сил информационных операций недружественных стран [8].

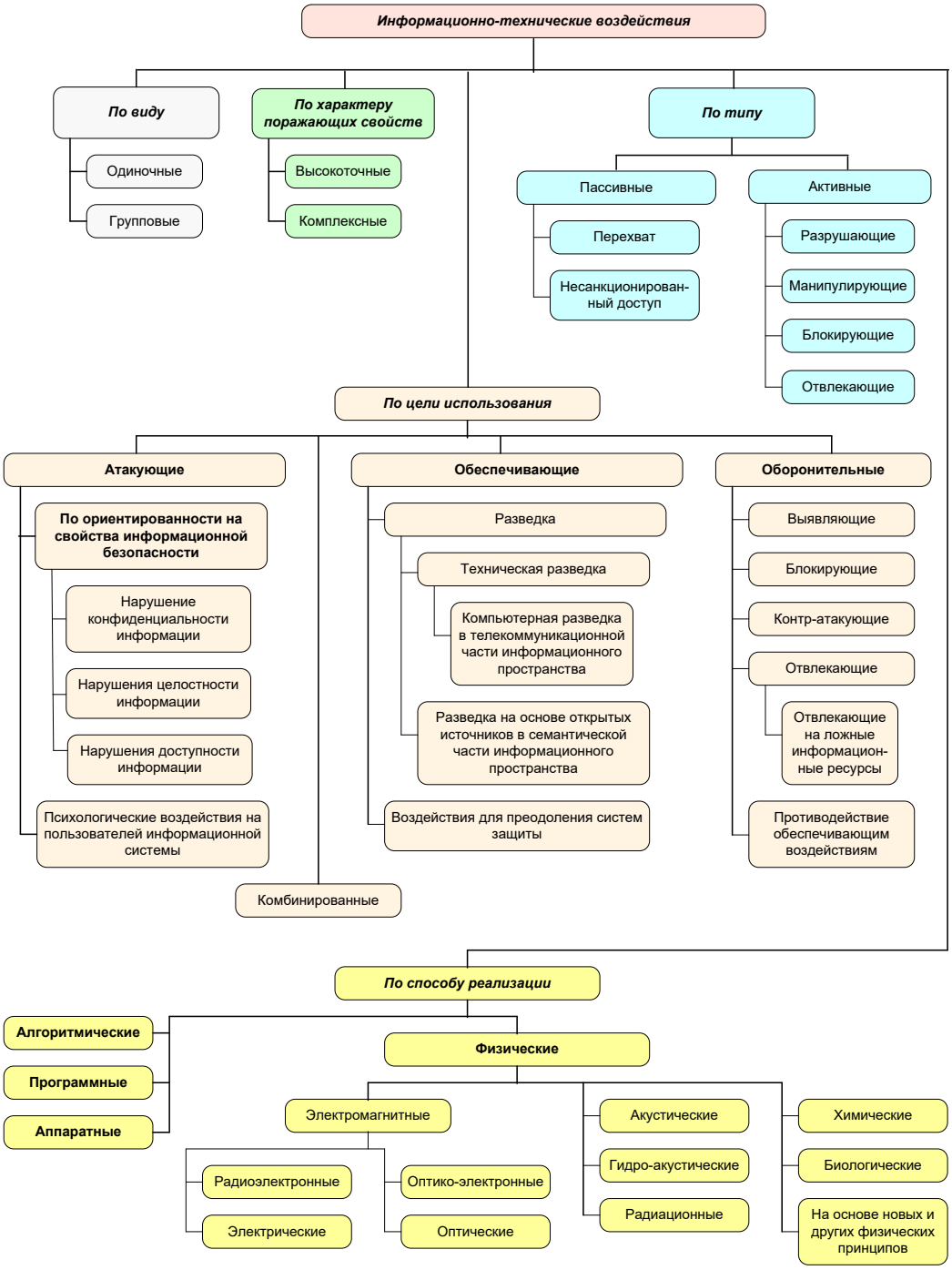

Рис. 1. Классификация ИТВ, предназначенных для тестирования объектов КИИ

Обобщая вышесказанное, можно сделать вывод о том, что функционал существующих центров ГосСОПКА, предназначенных для аудита ИБ ИС ЖТ, не позволяет реализовать превентивный практический аудит состояния ИБ этих ИС к воздействию прогнозируемых ИТВ нарушителей. Способом устранения этого недостатка является, во-первых, разработка теоретических основ тестирования ИС КИИ тестовыми ИТВ с целью практического аудита состояния их ИС, во-вторых, внесение в состав центров ГосСОПКА автоматизированного комплекса оценки тестирования ИС КИИ.

Предложения по использованию тестовых информационно-технических воздействий для аудита защищенности ИС ЖТ

Тестирование — проверка выполнения требований к системе при помощи наблюдения за ее работой в конечном наборе специально выбранных ситуаций [18].

Отдельное мероприятие по исследованию системы или способ изучения процессов ее функционирования называется *тестом* [18].

Тестовое ИТВ — воздействие на информационный ресурс, информационную систему, информационную инфраструктуру, на технические средства или на программы, решающие задачи получения, передачи, обработки, хранения

и воспроизведения информации с целью выявить уязвимости объекта на которое производится воздействие [18].

Общая классификация мероприятий, способов и средств тестирования, используемых при аудите ИБ, представлена на рисунке 1.

В настоящее время сложился подход к тестированию, когда подавляющая часть процессов оценивания безопасности системы основывается на анализе соответствия формальным требованиям по ИБ, а также путем тестирования на основе моделей. Вместе с тем, требования по ИБ, как правило, формулируется по итогам анализа инцидентов, что приводит к тому, что они регулярно отстают от современных возможностей и практики действий нарушителей.

Работы [22–24], посвященные вопросам экспериментального тестирования реальных ИС, рассматривают такие способы и сценарии исключительно как «тестирование на проникновение» или как «инструментальный аудит», при этом проведение такого типа аудита в отечественной практике не регламентируется каким-либо системным или хотя бы общетеоретическим подходом. В некоторых отечественных работах по тестированию на проникновение акцент делается на необходимости выявления наиболее «зрелищных» уязвимостей или тех уязвимостей, устранение которых принесет максимальные экономические выгоды компании, выполняющей аудит.

Вместе с тем прослеживается устойчивая тенденция к наращиванию доли тестов, которые проводятся в форме экспериментальных исследований реального объекта или его прототипа. Особенно это характерно при тестировании ПО. Как правило, для этого используются виртуальные машины, на которых осуществляется контролируемое выполнение тестируемого ПО. Развитие такого подхода к тестированию, что привело к разработке так называемых киберполигонов, которые виртуализируют как аппаратное, так и ПО распределенной ИС и позволяют отработать защиту от различных известных ИТВ. Сейчас это направление активно развивается.

Анализ [25, 26] зарубежных и отечественных методик тестирования на проникновение (OSSTMM, ISSAF, OWASP, PTES, NIST SP 800-115, BSI, PETA, методика от Positive Technology, методика от Digital Security), показывает, что они содержат достаточно развитые методические приемы проведения аудита ИБ, но не содержат исчерпывающего обоснования параметров и критериев выбора тестовых ИТВ, особенно применительно к тестированию объектов КИИ.

Обобщая вышесказанное, можно сделать следующие выводы:

1. Для повышения полноты и адекватности аудита состояния ИБ ИС к прогнозируемым ИТВ злоумышленников, целесообразно включить в состав соответствующих центров ГосСОПКА автоматизированные комплексы тестирования защищенности объекта КИИ, что показано на рисунке 2.

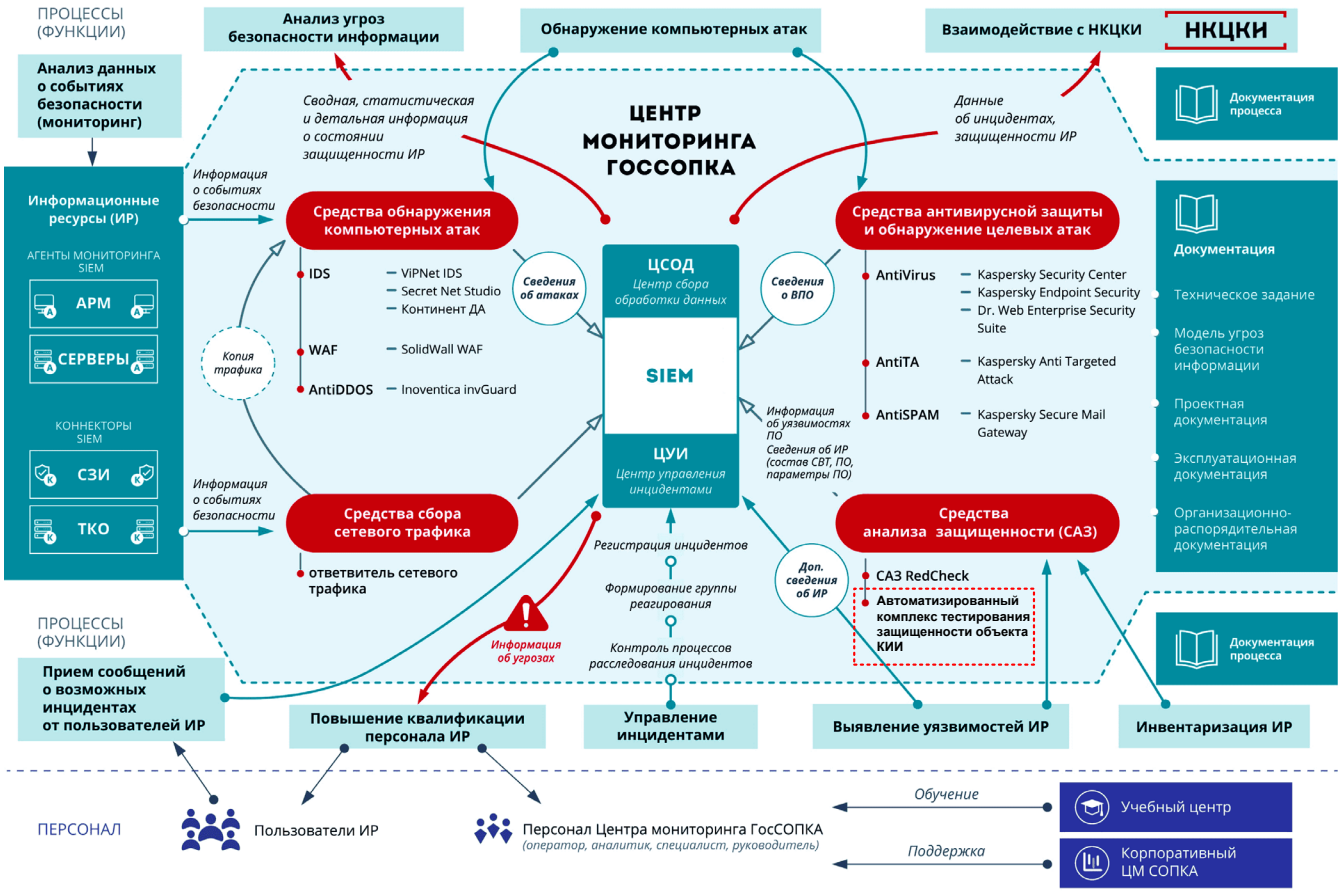

Рис. 2. Предлагаемая модернизация типовой архитектуры центра ГосСОПКА
на основе добавления автоматизированного комплекса тестирования защищенности объекта КИИ

2. Целесообразно провести научно-теоретические исследования в интересах формирования методик выбора тестовых ИТВ с учетом особенностей ИС, подвергающейся тестированию, полноты тестирования, и затрат ресурсов на него.

Внесение в состав центра ГосСОПКА автоматизированного комплекса тестирования защищенности позволит решить одну из основных задач обеспечения защищенности ИС КИИ — задачу превентивного аудита, когда уязвимость ИС к определенному типу ИТВ будет обнаруживаться до того, как это ИТВ будет использовано злоумышленниками.

Автоматизированный комплекс тестирования защищенности ИС КИИ должен включать следующие основные программные модули (рис. 3):

– базу данных ИТВ, которые потенциально могут быть реализованы злоумышленниками против конкретного ИС КИИ;

– базу сценариев проведения конкретных ИТВ;

– программный модуль, обеспечивающий автоматизированное формирование модели оценки защищенности объекта КИИ на основе данных, вводимых операторами центра ГосСОПКА;

– программный модуль, обеспечивающий автоматический расчет и формирование множества тестового набора ИТВ, оптимизированного по критерию «полнота тестирования/стоимость тестирования».

– программный модуль, обеспечивающий автоматизированное формирование множества ИТВ для тестирования ИС КИИ на основе данных, вводимых операторами центра ГосСОПКА.

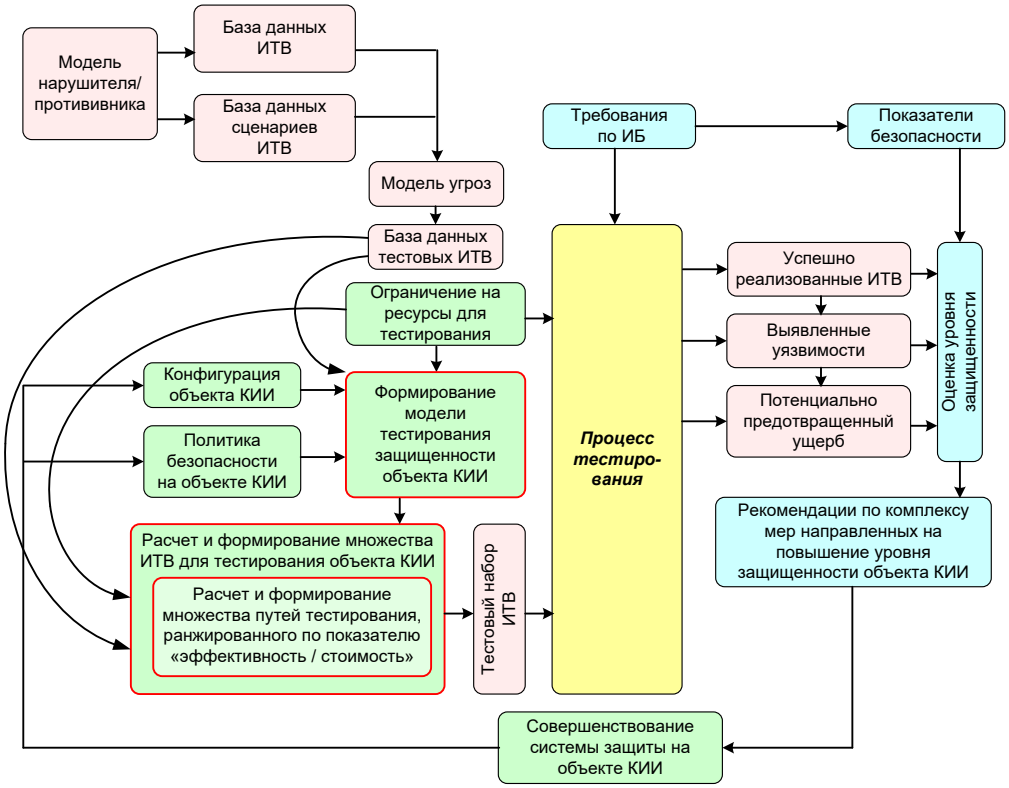

Рис. 3. Схема функционирования автоматизированного комплекса тестирования защищенности ИС КИИ

Вышеуказанные модули на практике позволяют обеспечить превентивный аудит состояния ИБ ИС КИИ центром ГосСОПКА путем его тестирования выбранными ИТВ и заблаговременное принятие мер по повышению уровня защищенности ИС КИИ до того, как злоумышленники смогут причинить ей невосполнимый критический ущерб. Именно такое превентивное обеспечение ИБ ИС и необходимо ЖТ, что также подтверждается выводами других ученых в работах [10, 27] в области обеспечения ИБ ИС ЖТ как объекта КИИ.

Заключение

В статье рассмотрены ИС ЖТ. Обоснована необходимость обеспечения их защиты как объекта КИИ, а также проведен краткий анализ нормативных документов, регламентирующих обеспечение ИБ информационной инфраструктуры ЖТ. Показано, что существующие способы аудита состояния ИБ ИС ЖТ осуществляются соответствующими центрами системы ГосСОПКА. При этом существующая архитектура центров ГосСОПКА не предусматривает такой функциональности, как практический аудит ИБ ИС тестовыми ИТВ, аналогичными ИТВ, которые прогнозируются к применению злоумышленниками. Обоснована целесообразность применения такого аудита, а также предложен вариант совершенствования типовой архитектуры центра ГосСОПКА на основе включения в его состав автоматизированного комплекса тестирования защищенности ИС КИИ. Представлены предложения по составу и порядку функционирования такого автоматизированного комплекса тестирования.

Литература

1. О безопасности критической информационной инфраструктуры Российской Федерации: федеральный закон от 26.07.2017 № 187-ФЗ.

2. Исаков О. А. Вопросы совершенствования АСУ железнодорожного транспорта. — Саарбрюккен: Lambert Academic Publishing, 2012. — 224 с.

3. Санькова Г. В. Информационные технологии в перевозочном процессе: Учебное пособие / Г. В. Санькова, Т. А. Одуденко; Министерство транспорта Российской Федерации; ФГБОУ ВПО ДВГУПС. — Хабаровск: Изд-во ДВГУПС, 2012. — 111 с.

4. Информационные системы железнодорожного транспорта: Рабочая программа и задание на контрольную работу с методическими указаниями для студентов специальности 230101 «Вычислительные машины, комплексы, системы и сети (ЭВМ)» / сост. Г. В. Самме. — М.: РГОТУПС, 2008. — 30 с.

5. Котенко И. В. Анализ защищенности инфраструктуры железнодорожного транспорта на основе аналитического моделирования / И. В. Котенко, А. А. Чечулин, Д. С. Левшун // Защита информации. Инсайд. 2017. № 6 (78). С. 48–57.

6. Определение уровня безопасности значимых объектов критической информационной инфраструктуры железнодорожного транспорта / А. П. Глухов, В. В. Василенко, А. А. Сидак [и др.] // Двойные технологии. 2020. № 1 (90). С. 84–88.

7. Международная кибербезопасность на железнодорожном транспорте: методологические подходы и нормативная методическая база / С. Е. Ададуров, С. В. Диасамидзе, А. А. Корниенко, А. А. Сидак // Вестник научно-исследовательского института железнодорожного транспорта (Вестник ВНИИЖТ). 2015. № 6. С. 9–15.

8. Макаренко С. И. Информационное оружие в технической сфере: терминология, классификация, примеры // Системы управления, связи и безопасности. 2016. № 3. С. 292–376. DOI: 10.24411/2410-9916-2016-10311.

9. Кузнецов Д. ГосСОПКА: что такое, зачем нужна и как устроена. — 02.04.2019 // Anti-Malware.
URL: http://www.anti-malware.ru/analytics/Technology_Analysis/gossopka-what-is-it-how-it-works (дата обращения 25.11.2019).

10. О безопасности критической информационной инфраструктуры / С. Е. Ададуров, А. П. Глухов, А. А. Корниенко, Е. И. Белова // Автоматика, связь, информатика. 2020. № 4. С. 2–4. DOI: 10.34649/AT.2020.4.4.001.

11. Котенко И. В. Об архитектуре многоуровневой интеллектуальной системы обеспечения информационной безопасности автоматизированных систем на железнодорожном транспорте / И. В. Котенко, И. Б. Саенко // Методы и технические средства обеспечения безопасности информации: Материалы 23-й научно-технической конференции (Санкт-Петербург, 30 июня—03 июля 2014 г.). — СПб.: Изд-во Политехнического ун-та, 2014. — С. 97–98.

12. Котенко И. В. Предложения по созданию многоуровневой интеллектуальной системы обеспечения информационной безопасности автоматизированных систем на железнодорожном транспорте / И. В. Котенко, И. Б. Саенко // Вестник Ростовского государственного университета путей сообщения. 2013. № 3 (51). С. 69–79.

13. Построение многоуровневой интеллектуальной системы обеспечения информационной безопасности для автоматизированных систем железнодорожного транспорта / И. В. Котенко, И. Б. Саенко, А. В. Чернов, М. А. Бутакова // Труды СПИИРАН. 2013. № 7 (30). С. 7–25.

14. Kotenko I. Intelligent Security Analysis of Railway Transport Infrastructure Components on the Base of Analytical Modeling / I. Kotenko, A. Chechulin, M. Bulgakov // Proceedings of the Second International Scientific Conference «Intelligent Information Technologies for Industry» (IITI'17) (Varna, 14–16 September 2017), Vol. 2 / A. Abraham, S. Kovalev, et al. (eds) // Advances in Intelligent Systems and Computing, Vol. 680. Pp. 178–188.
DOI: 10.1007/978-3-319-68324-9_20.

15. Чернов А. В. Методы распределенных рассуждений в интеллектуальных системах ситуационной осведомленности об инцидентах в критической информационной инфраструктуре / А. В. Чернов, М. А. Бутакова, В. Д. Вескун // Сборник докладов XXI Международной конференция по мягким вычислениям и измерениям (SCM`2018) (Санкт-Петербург, 23–25 мая 2018 г.). — СПб.: Изд-во СПбГЭТУ «ЛЭТИ», 2018. — Т. 1. — С. 618–621.

16. Управление безопасностью кибер-физических систем на основе оперативного ситуационного информирования об инцидентах / М. А. Бутакова, А. В. Чернов, П. С. Шевчук, С. М. Ковалев // Труды Ростовского государственного университета путей сообщения. 2016. № 5. С. 14–16.

17. Макаренко С. И. Аудит информационной безопасности: основные этапы, концептуальные основы, классификация мероприятий // Системы управления, связи и безопасности. 2018. № 1. С. 1–29.
DOI: 10.24411/2410-9916-2018-10101.

18. Макаренко С. И. Аудит безопасности критической инфраструктуры специальными информационными воздействиями: Монография. — СПб.: Наукоемкие технологии, 2018. — 122 с.

19. Скабцов Н. Аудит безопасности информационных систем. — СПб.: Питер, 2018. — 272 с. — (Библиотека программиста).

20. Penetration Testing. Procedures and Methodologies. — USA: Cource Techology: EC-Council Press, 2010. — 256 p.

21. Metasploit. The Penetration Tester's Guide / D. Kennedy, J. O'Gorman, D. Kearns, M. Aharoni. — San Francisco: No Starch Press, 2011. — 328 p.

22. Cardwell K. Building Virtual Pentesting Labs for Advanced Penetration Testing. Second Revised Edition. — Birmingham: Packt Publishing Ltd., 2016. — 524 p.

23. Makan K. Penetration Testing with the Bash shell. — Birmingham: Packt Publishing Ltd., 2014. — 150 p.

24. Baloch R. Ethical Hacking and Penetration Testing Guide. — CRC Press: Auerbach Publications, 2017. — 531 p.

25. Богораз А. Г. Сравнительный анализ методик оценки межсетевых экранов / А. Г. Богораз, О. Ю. Пескова // Интернет и современное общество: Сборник научных статей. Труды XVI Всероссийской объединенной научной конференции (IMS`2013) (Санкт-Петербург, 9–11 октября 2013 г.). — СПб.: НИУ ИТМО, 2013. — С. 202–209.

26. Klíma T. PETA: Methodology of Information Systems Security Penetration Testing // Acta Informatica Pragensia. 2016. Vol. 5, No. 2. Pp. 98–117. DOI: 10.18267/j.aip.88.

27. Методологические аспекты упреждающего управления информационной безопасностью железнодорожного транспорта / А. П. Глухов, Д. Н. Бирюков, В. В. Василенко [и др.] // Двойные технологии. 2019. № 3 (88). С. 86–92.

The Use of Test Information and Technical Impacts for Security Audit of Information Systems of Railway Transport

G. E. Smirnov
Intel Group Corporation LLC
Saint Petersburg, Russia
science.cybersec@yandex.ru

Grand PhD S. I. Makarenko
Saint Petersburg Federal Research Center of the Russian Academy of Sciences
Saint Petersburg Electrotechnical University 'LETI'
Saint Petersburg, Russia
mak-serg@yandex.ru

Abstract. **The article discusses the main information and automated systems of railway transport (RT). It is shown that these systems are objects of a critical information infrastructure. In accordance with the legislation of the Russian Federation, such facilities must be connected to the centers of the State system for detection, prevention and elimination of the consequences of computer attacks (SSDPECCA), which audit the state of their information security (IS). It is shown that the existing GosSOPKA centers that audit the state of information security of information systems of RT do not provide for such functionality as an assessment of the security of information systems by test information and technical impacts (ITI), similar to ITV, which are predicted for use by intruders. The expediency of using this kind of audit has been substantiated, and a variant of improving the standard architecture of the SSDPECCA center has been proposed by including an automated complex for testing the security of information systems of the RT into its composition. Proposals are presented on the composition and procedure for the operation of such an automated testing complex.**

Keywords: **information security, audit, testing, information technology impact, critical information infrastructure, rail transport.**

REFERENCES

1. On the Security of the Critical Information Infrastructure of the Russian Federation: Federal Law from July 26, 2017 No. 187-FZ [O bezopasnosti kriticheskoy informatsionnoy infrastruktury Rossiyskoy Federatsii: federal'nyy zakon ot 26.07.2017 № 187-FZ].

2. Isakov O. A. Questions of improving the automated control system of railway transport [Voprosy sovershenstvovaniya ASU zheleznodorozhnogo transporta], Saarbrücken, Lambert Academic Publishing, 2012, 224 p.

3. Sankova G. V., Odudenko T. A. Information technologies in the transportation process: Study Guide [Informatsionnyye tekhnologii v perevozochnom protsesse: Uchebnoe posobie], Khabarovsk, Far Eastern State Transport University, 2012, 111 p.

4. Summe G. V. Information systems of railway transport: Work program and test task with methodological instructions [Informatsionnye sistemy zheleznodorozhnogo transporta: Rabochaya programma i zadanie na kontrol'nuyu rabotu s metodicheskimi ukazaniyami], Moscow, Russian State Open Technical University of Railway Transport, 2008, 30 p.

5. Kotenko I. V., Chechulin A. A., Levshun D. S. Security Analysis of Railway Transport Infrastructure on the Base of Analytical Modeling [Analiz zashchishchennosti infrastruktury zheleznodorozhnogo transporta na osnove analiticheskogo modelirovaniya], *Zašita informacii. Inside. [Zashchita informatsii. Insayd]*, 2017, No. 6 (78), Pp. 48–57.

6. Gluhov A. P., Vasilenko V. V., Sidak A. A., et al. Determination of The Security Level of Significative Objects of Critical Information Infrastructure of Railway Transport [Opredelenie urovnya bezopasnosti znachimykh ob"ektov kriticheskoy informatsionnoy infrastruktury zheleznodorozhnogo transporta], *Dual technology [Dvoynyye tekhnologii]*, 2020, No. 1 (90), Pp. 84–88.

7. Adadurov S. E., Diasamidze S. V., Kornienko A. A., Sidak A. A. International Cybersecurity on Railway Transport: Methodological Approaches and Normal Procedural Framework [Mezhdunarodnaya kiberbezopasnost' na zheleznodorozhnom transporte: metodologicheskie podkhody i normativnaya metodicheskaya baza], *VNIIZHT Scientific Journal [Vestnik nauchno-issledovatel'skogo instituta zheleznodorozhnogo transporta (Vestnik VNIIZhT)]*, 2015, No. 6, Pp. 9–15.

8. Makarenko S. I. Information Weapon in Technical Area — Terminology, Classification and Examples [Informatsionnoe oruzhie v tekhnicheskoy sfere: terminologiya, klassifikatsiya, primery], *Systems of Control, Communication and Security [Sistemy upravleniya, svyazi i bezopasnosti]*, 2016, No. 3, Pp. 292–376. DOI: 10.24411/2410-9916-2016-10311.

9. Kuznetsov D. GosSOPKA: what is, why is it needed and how is it arranged [GosSOPKA: chto takoye, zachem nuzhna i kak ustroyena], *Anti-Malware*. Published at April 02, 2019. Available at: http://www.anti-malware.ru/analytics/Technology_Analysis/gossopka-what-is-it-how-it-works (accessed 25 Nov 2019).

10. Adadurov S. E., Glukhov A. P., Kornienko A. A., Belova E. I. Principles of Railway Transport Critical Information Infrastructure Security Supporting [O bezopasnosti kriticheskoy informatsionnoy infrastruktury], *Automation, Communication, and Informatics [Avtomatika, svyaz', informatika]*, 2020, No. 4, Pp. 2–4. DOI: 10.34649/AT.2020.4.4.001.

11. Kotenko I. V., Saenko I. B. On the Architecture of a Multilevel Intelligent Information Security System for Automated Systems in Railway Transport [Ob arkhitekture mnogourovnevoy intellektual'noy sistemy obespecheniya informatsionnoy bezopasnosti avtomatizirovannykh sistem na

zheleznodorozhnom transporte], *Methods and technical means of information security: Proceedings of the XXIII Scientific and Technical Conference [Metody i tekhnicheskie sredstva obespecheniya bezopasnosti informatsii: Materialy XXIII nauchno-tekhnicheskoy konferentsii], St. Petersburg, June 30 — July 03, 2014)*, St. Petersburg, Peter the Great St. Petersburg Polytechnic University, 2014, Pp. 97–98.

12. Kotenko I. V., Saenko I. B. Proposals on Creation of a Multi-Level Intelligent Information Security System of Automated Systems on Railway Transport [Predlozheniya po sozdaniyu mnogourovnevoy intellektual'noy sistemy obespecheniya informatsionnoy bezopasnosti avtomatizirovannykh sistem na zheleznodorozhnom transporte], *Vestnik RGUPS [Vestnik Rostovskogo gosudarstvennogo universiteta putey soobshcheniya]*, 2013, No. 3 (51), Pp. 69–79.

13. Kotenko I. V., Saenko I. B., Chernov A. V., Butakova M. A. The Construction of a Multi-Level Intelligent Information Security System for Automated Systems of Railway Transport [Postroenie mnogourovnevoy intellektual'noy sistemy obespecheniya informatsionnoy bezopasnosti dlya avtomatizirovannykh sistem zheleznodorozhnogo transporta], *SPIIRAS Proceedings [Trudy SPIIRAN]*, 2013, No. 7 (30), Pp. 7–25.

14. Kotenko I., Chechulin A., Bulgakov M. Intelligent Security Analysis of Railway Transport Infrastructure Components on the Base of Analytical Modeling. *In: Abraham A., Kovalev S., et al. (eds) Proceedings of the Second International Scientific Conference «Intelligent Information Technologies for Industry» (IITI'17), Varna, September 14–16, 2017). Vol. 2. Advances in Intelligent Systems and Computing*, Vol. 680. Pp. 178–188. DOI: 10.1007/978-3-319-68324-9_20.

15. Chernov A. V., Butakova M. A., Vereskun V. D. Methods of Distributed Reasoning for Intelligent Systems of Situational Awareness About Incidents in Critical Information Infrastructure [Metody raspredelennykh rassuzhdeniy v intellektual'nykh sistemakh situatsionnoy osvedomlennosti ob intsidentakh v kriticheskoy informatsionnoy infrastrukture], *Collection of Reports of the XXI International Conference on Soft Computing and Measurement (SCM'2018) [Sbornik dokladov XXI Mezhdunarodnoy konferentsiya po myagkim vychisleniyam i izmereniyam (SCM`2018)], St. Petersburg, May 23–25, 2018*, Saint Petersburg, Saint Petersburg Electrotechnical University "LETI", 2018, Vol. 1, Pp. 618–621.

16. Butakova M. A., Chernov A. V., Shevchuk P. S., Kovalev S. M. Cyber-Physical Systems Security Management Based on Operational Situational Incidents Information [Upravlenie bezopasnost'yu kiber-fizicheskikh sistem na osnove operativnogo situatsionnogo informirovaniya ob intsidentakh], *Proceedings of the Rostov State Transport University [Trudy Rostovskogo gosudarstvennogo universiteta putey soobshcheniya]*, 2016, No. 5, Pp. 14–16.

17. Makarenko S. I. Audit of Information Security — the Main Stages, Conceptual Framework, Classification of Types [Audit informatsionnoy bezopasnosti: osnovnye etapy, kontseptual'nye osnovy, klassifikatsiya meropriyatiy], *Systems of Control, Communication and Security [Sistemy upravleniya, svyazi i bezopasnosti]*, 2018, No. 1, Pp. 1–29. DOI: 10.24411/2410-9916-2018-10101.

18. Makarenko S. I. Security audit of critical infrastructure with special information impacts: Monography [Audit bezopasnosti kriticheskoy infrastruktury spetsial'nymi informatsionnymi vozdeystviyami: Monografiya], Saint Petersburg, Naukoemkie tekhnologii Publishers, 2018, 122 p.

19. Skabtsov N. Audit of information systems security [Audit bezopasnosti informatsionnykh sistem], Saint Petersburg, Piter Publishing House, 2018, 272 p.

20. Penetration Testing. Procedures and Methodologies. USA, Cource Techology, EC-Council Press, 2010, 256 p.

21. Kennedy D., O'Gorman J., Kearns D., Aharoni M. Metasploit. The Penetration Tester's Guide. San Francisco, No Starch Press, 2011, 328 p.

22. Cardwell K. Building Virtual Pentesting Labs for Advanced Penetration Testing. Second Revised Edition. Birmingham, Packt Publishing Ltd., 2016, 524 p.

23. Makan K. Penetration Testing with the Bash shell. Birmingham, Packt Publishing Ltd., 2014, 150 p.

24. Baloch R. Ethical Hacking and Penetration Testing Guide. CRC Press, Auerbach Publications, 2017, 531 p.

25. Bogoraz A. G., Peskova O. Yu. Comparative Analysis of Methods for Evaluating Firewalls [Sravnitel'nyy analiz metodik otsenki mezhsetevykh ekranov], *Internet and Modern Society: Proceedings of the XVI All-Russian Joint Scientific Conference (IMS'2013) [Internet i sovremennoe obshchestvo: Sbornik nauchnykh statey. Trudy XVI Vserossiyskoy ob"edinennoy nauchnoy konferentsii (IMS`2013)], Saint Petersburg, October 9–11, 2013*, Saint Petersburg, ITMO University, 2013, Pp. 202-209.

26. Klíma T. PETA: Methodology of Information Systems Security Penetration Testing, *Acta Informatica Pragensia*, 2016, Vol. 5, No. 2, Pp. 98–117. DOI: 10.18267/j.aip.88.

27. Glukhov A. P., Biryukov D. N., Vasilenko V. V., et al. Methodological Aspects of Proactive Management of Railroad Transport Information Security [Metodologicheskie aspekty uprezhdayushchego upravleniya informatsionnoy bezopasnost'yu zheleznodorozhnogo transporta], *Dual technology [Dvoynyye tekhnologii]*, 2019, No. 3 (88), Pp. 86–92.

К оценке эффективности системы обнаружения вторжений на основе матричных игр и нечетких множеств

к.ф.-м.н. В. Б. Вилков
Военная академия материально-технического обеспечения имени генерала армии А. В. Хрулева
Министерства обороны Российской Федерации
Санкт-Петербург, Россия
amirusha@rambler.ru

к.т.н. А. К. Черных
Санкт-Петербургский военный ордена Жукова институт войск национальной гвардии Российской Федерации
Санкт-Петербург, Россия
nataliachernykh@mail.ru

к.воен.н. А. И. Дергачёв,
к.т.н. О. Н. Куранова
Петербургский государственный университет путей сообщения Императора Александра I
Санкт-Петербург, Россия
d_ader@mail.ru
olga_kuranova@mail.ru

Аннотация. **Рассматривается задача, сформулированная как матричная игра, в которой выигрышем должностных лиц, использующих конкретную систему обнаружения вторжений (преступных действий) злоумышленников (игрок 1), является вероятность своевременного обнаружения этих преступных действий (игрок 2). Как правило, однозначно задать вероятность своевременного обнаружения преступных действий не представляется возможным, поэтому для ее оценки предлагается использовать аппарат теории нечетких множеств. Рассмотрены и раскрыты основные понятия теории нечетких множеств, а также приведен пример практического применения данной теории для оценки эффективности использования системы обнаружения преступных действий злоумышленников. Применение теории нечетких множеств в части оценки возможных действий злоумышленника позволяет обнаружить имеющиеся уязвимости в информационной безопасности автоматизированной системы, и далее провести совершенствование систем обнаружения преступных действий злоумышленников (хакеров), предотвращающих возможность нанесения экономического и иного ущерба компании.**

***Ключевые слова*: система обнаружения вторжений злоумышленников, матричная игра, нечеткие множества, решение матричной игры.**

Введение

Оценка эффективности применения систем обнаружения вторжений и преступных действий злоумышленников в современных автоматизированных системах в настоящее время является весьма актуальной задачей.

В настоящей статье рассматривается задача, сформулированная как матричная игра [1], в которой выигрышем должностных лиц, использующих конкретную систему обнаружения вторжений (преступных действий) злоумышленников (игрок 1), является вероятность своевременного обнаружения этих преступных действий (игрок 2). Как правило, однозначно задать вероятность своевременного обнаружения преступных действий не представляется возможным, поэтому для её оценки предлагается использовать аппарат теории нечетких множеств.

Рассмотрены и раскрыты основные понятия теории нечетких множеств, а также приведен пример практического применения данной теории для оценки эффективности использования системы обнаружения преступных действий злоумышленников.

Применение теории нечетких множеств [2–7] в части оценки возможных действий злоумышленника позволяет обнаружить имеющиеся уязвимости в информационной безопасности автоматизированной системы, и далее провести совершенствование систем обнаружения преступных действий злоумышленников (хакеров), предотвращающих возможность нанесения экономического и иного ущерба компании.

Основные понятия и определения

Под матричной игрой будем понимать игру двух участников, в которой каждый из них имеет определенное число вариантов своего действия (стратегий) [1–7]. Игроки одновременно и независимо один от другого выбирают свою стратегию. Такой выбор стратегий однозначно определяет выигрыши игроков, которые в сумме равны нулю.

Предположим, что $a_1, a_2, \dots, a_m$ — стратегии первого игрока, а $b_1, b_2, \dots, b_n$ — стратегии второго игрока. Выбор игроками своих стратегий назовем ситуацией. Под ситуацией (i, j) будем понимать ситуацию, в которой первый игрок выбрал стратегию a_i, второй — b_j. Для игры g определены функции выигрыша $H_g(i, j)$ и $-H_g(i, j)$ соответственно первого и второго игроков, соотносящие каждой возможной в игре ситуации их выигрыши; при этом каждый игрок стремится максимизировать свой выигрыш.

Матрица выигрышей первого игрока (A_g) однозначно задает игру g и имеет вид

$$A_g = \begin{pmatrix} H_g(1,1) & H_g(1,2) & \dots & H_g(1,n) \\ H_g(2,1) & H_g(2,2) & \dots & H_g(2,n) \\ \vdots & \vdots & \vdots & \vdots \\ H_g(m,1) & H_g(m,2) & \dots & H_g(m,n) \end{pmatrix} \quad (1)$$

Игроки могут гарантировать себе выигрыши, равные соответственно $\max\limits_{1\le i\le m} \min\limits_{1\le j\le n} H_g(i,j)$ и $\min\limits_{1\le j\le n} \max\limits_{1\le i\le m} H_g(i,j)$

В матричной игре g ситуация (i_0, j_0) называется равновесной, или седловой, точкой, если

$$H_g(i, j_0) \le H_g(i_0, j_0) \le H_g(i_0, j) \quad (2)$$

при $1 \le i \le m$ и $1 \le j \le n$.

Ценой игры называется выигрыш первого игрока в ситуации равновесия, а стратегии, образующие седловую

точку, называются оптимальными. Следует отметить, что далеко не всякая игра имеет седловую точку, что придает предлагаемому в статье подходу актуальность.

При решении разных задач мы далеко не всегда можем однозначно утверждать, что данный объект полностью обладает (не обладает) свойствами, присущими элементам данного множества. Например, рассмотрим множество возможных значений вероятностей обнаружения злоумышленников «вероятность обнаружения злоумышленника большая». Про вероятность обнаружения злоумышленника 0,8 мы однозначно сказать, является или не является она элементом этого множества, не можем. Это вызвало необходимость создания теории нечетких множеств. Дадим некоторые определения.

Нечеткие множества задаются на некотором обычном множестве U, называемом универсальным. Это может быть множество автоматизированных систем, множество преступных действий злоумышленников, множество матричных игр и т. п. Нечетким множеством $\hat{A}$ на универсальном множестве U называется совокупность пар $(\mu_{\hat{A}}(u), u), u \in U$, где $\mu_{\hat{A}}(u)$ — функция принадлежности, выражающая степень принадлежности элемента $u \in U$ к нечеткому множеству $\hat{A}$. Как правило, предполагается, что функция принадлежности принимает значения из отрезка [0, 1].

Пересечением нечетких множеств $\hat{A}$ и $\hat{B}$ заданных на U, называется нечеткое множество $\hat{C} = \hat{A} \cap \hat{B}$ с функцией принадлежности

$$\mu_{\hat{C}}(u) = min\{\mu_{\hat{A}}(u), \mu_{\hat{B}}(u)\}, u \in U, \quad (3)$$

их объединением — нечеткое множество $\hat{D} = \hat{A} \cup \hat{B}$ с функцией принадлежности

$$\mu_{\hat{D}}(u) = max\{\mu_{\hat{A}}(u), \mu_{\hat{B}}(u)\}, u \in U. \quad (4)$$

Нечеткие множества в случае, когда универсальным множеством является числовая ось и функции, принадлежности которых непрерывны и имеют единственный максимум, называются нечеткими числами. Часто при решении практических задач используются треугольные нечеткие числа. Треугольным нечетким числом $\hat{D}$ называется такая тройка $\langle c, d, f\rangle, c < d < f$ действительных чисел, что

$$\mu_{\hat{D}}(u) = \begin{cases} \frac{u-c}{d-c}, \text{если } u \in [c, d], \\ \frac{f-u}{f-d}, \text{если } u \in [d, f], \\ 0, \text{иначе.} \end{cases} \quad (5)$$

Второе число тройки $\langle c, d, f\rangle$ обычно называют модой, или четким значением, нечеткого треугольного числа $\hat{D}$, $\mu_{\hat{D}}(d) = 1$.

Следуя публикациям [4, 5, 8], введем некоторые понятия нечеткой логики. Степень истинности нечеткого высказывания принимает значения из замкнутого промежутка [0;1].

Степень истинности нечеткого высказывания P обозначим $\mu(P)$.

Конъюнкцией нечетких высказываний P и T называется логическая операция, результатом которой является нечеткое высказывание $P \wedge T$, для которого

$$\mu(P \wedge T) = min\{\mu(P), \mu(T)\}, \quad (6)$$

дизъюнкцией $P \vee T$ — логическая операция, для которой

$$\mu(P \vee T) = max\{\mu(P), \mu(T)\}. \quad (7)$$

Рассмотрим постановку задачи для нечеткой матричной игры.

Постановка задачи для нечеткой матричной игры

Пусть G есть множество всех матричных игр с m стратегиями у первого игрока и n стратегиями у второго игрока. Будем рассматривать G как универсальное множество, на котором заданы нечеткие множества — нечеткие матричные игры $\hat{g}$, то есть игры, в которых выигрыши задаются нечеткими треугольными числами

$$\hat{D}_{ij}(\hat{g}) = \langle c_{ij}(\hat{g}), d_{ij}(\hat{g}), f_{ij}(\hat{g})\rangle.$$

Функцию принадлежности нечеткой игры $\hat{g}$ обозначим $\mu_{\hat{g}}(g), g \in$. В силу того, что матричная игра однозначно определяется матрицей выигрышей, будем считать, что $g = A_g$ и, следовательно, $\mu_{\hat{g}}(g) = \mu_{\hat{g}}(A_g)$.

Пусть $\hat{g}$ — нечеткая игра, заданная на G, и выигрыши первого игрока являются нечеткими числами

$$\hat{D}_{ij}(\hat{g}) = \langle c_{ij}(\hat{g}), d_{ij}(\hat{g}), f_{ij}(\hat{g})\rangle, i = 1,2, \dots, m, j = 1,2, \dots, n$$

с функциями принадлежности $\mu_{ij}^{\hat{g}}$.

В соответствии с определением конъюнкции в нечеткой логике имеем:

$$\mu_{\hat{g}}(A(g)) = \min_{i,j} \mu_{ij}(H_{ij}(g)). \quad (8)$$

Рассмотрим нечеткую матричную игру $\hat{g}$ с нечеткой матрицей выигрышей $\hat{A}_{\hat{g}} = \left\|\hat{D}_{ij}(\hat{g})\right\|_{i,j=1}^{m,n}$ и множество игр, для которых ситуация (i, j) является седловой точкой. Через $F^{ij}(\hat{g})$ обозначим множество матриц выигрышей таких игр.

Пусть $A_0^{ij}(\hat{g}) \in F^{ij}(\hat{g})$ и $\mu_{\hat{g}}\left(A_0^{ij}(\hat{g})\right) = \max_{A \in F^{ij}(\hat{g})} \mu_{\hat{g}}(A)$.

Величину $\mu_{\hat{g}}\left(A_0^{ij}(\hat{g})\right)$ будем рассматривать как степень надежности того, что ситуация (i, j) в игре $\hat{g}$ является седловой точкой. Решением игры $\hat{g}$ будем считать ситуацию (i, j), для которой надежность того, что она является седловой точкой, максимальна.

Задача по отысканию описанного решения игры сводится к решению ряда задач математического программирования. Приведем для содержательного примера графический подход к решению этой задачи.

Пример использования графического подхода к решению игровой задачи

В рамках совершенствования автоматизированной системы управления транспортными процессами ОАО «РЖД», необходимо эффективно использовать систему обнаружения преступных действий злоумышленников в части, касающейся нарушения управления транспортными процессами, предотвращающую четыре варианта преступных действий хакеров. Хакеры имеют четыре стратегии вредоносных действий. В качестве вредоносных действий хакеров, направленных на нарушение информационной безопасности указанной автоматизированной системы, будем рассматривать: анализ сетевого трафика (в дальнейшем — b_1), DDOS атаки (в дальнейшем — b_2), вирусное

заражение данных (в дальнейшем — b_3) и перехват пароля (в дальнейшем — b_4). Должностные лица, использующие систему обнаружения преступных действий хакеров (первый игрок), также имеют четыре стратегии предотвращения этих действий (стратегии защиты), соответственно: шифрование трафика (в дальнейшем — a_1), межсетевое экранирование (в дальнейшем — a_2), антивирусная защита (в дальнейшем — a_3), однократное паролирование (шифрование канала связи) (в дальнейшем — a_4). Под их выигрышем предлагается рассматривать вероятности своевременного обнаружения действий хакеров (второго игрока). Предполагаем, что имеющейся информации об этих вероятностях недостаточно, и она носит нечеткий характер и задается с помощью нечетких чисел $\langle c_{ij}, d_{ij}, f_{ij} \rangle$ (их моды указаны в таблице 1).

Таблица 1

Моды нечетких выигрышей первого игрока

Стратегии защиты	**Стратегии хакеров**			
	b_1	b_2	b_3	b_4
a1	0,95	0,60	0,50	0,65
a2	0,60	0,90	0,60	0,55
a3	0,50	0,65	0,95	0,70
a4	0,50	0,60	0,65	0,95

Отметим существующую особенность решения приведенного примера, заключающуюся в том, что, задавая выигрыши нечетким числами, мы можем получить в качестве ответа ситуацию с нулевым значением функции принадлежности.

В качестве иллюстрации сказанного рассмотрим несколько случаев.

1. В случае, если $c_{ij} = d_{ij} - 0{,}05$ и $f_{ij} = d_{ij} + 0{,}05$, надежность того, что ситуация является седловой точкой, равна нулю, что требует дополнительных пояснений.

2. Если $c_{ij} = max\{d_{ij} - 0{,}2{,}0\}$ и $f_{ij} = min\{d_{ij} + 0{,}3{,}1\}$, то ситуации становятся седловыми точками с максимальными надежностями, указанными в таблице 2.

Таблица 2

Надежности седловых точек (случай 2)

Стратегии защиты	**Стратегии хакеров**			
	b_1	b_2	b_3	b_4
a1	0,1	0,2	0,1	0,1
a2	0,2	**0,3**	0,2	0,2
a3	0,1	0,2	0,1	0,1
a4	0,1	0,2	0,1	0,1

Решением является ситуация (2,2), в этой ситуации с надежностью 0,3 первый игрок выигрывает 0,76 (с надежностью 0,3 вероятность обнаружения преступных действий равна 0,76).

3. Рассмотрим несколько случаев, когда $[c_{ij}, f_{ij}]$ одинаковы и равны $[c, f]$ при любых i и j. В этом случае в качестве решения любой такой игры мы получим некоторую ситуацию с надежностью большей нуля.

3.1. В случае 1 минимальное значение для c_{ij} равно 0,45, максимальное для f_{ij} равно единице. Если в качестве $[c, f]$ использовать интервал [0,45,1] что, на наш взгляд, не лишено смысла, то ситуации становятся седловыми точками с максимальными надежностями, указанными в таблице 3.

Таблица 3

Надежности седловых точек (случай 3.1)

Стратегии защиты	**Стратегии хакеров**			
	b_1	b_2	b_3	b_4
a1	0,55	0,579	0,55	0,55
a2	0,579	**0,611**	0,579	0,579
a3	0,55	0,579	0,55	0,55
a4	0,55	0,579	0,55	0,55

Решением является ситуация (2,2), в ней с надежностью 0,611 первый игрок выигрывает 0,725.

3.2. Если реализовывать схему случая 2, то в качестве $[c, f]$ можно было бы использовать интервал [0,3,1]. Тогда ситуации становятся седловыми точками с максимальными надежностями, указанными в таблице 4.

Таблица 4

Надежности седловых точек (случай 3.2)

Стратегии защиты	**Стратегии хакеров**			
	b_1	b_2	b_3	b_4
a1	0,608	0,636	0,608	0,608
a2	0,636	**0,666**	0,636	0,636
a3	0,608	0,636	0,608	0,608
a4	0,608	0,636	0,608	0,608

Решением является ситуация (2,2), в ней с надежностью 0,666 первый игрок выигрывает 0,70.

4. В качестве выигрышей первого игрока рассматривать не вероятности, а полезности ситуаций для него. Если опираться на использованный Д. фон Нейманом и О. Моргенштерном вероятностный подход к определению полезности, то полезность лежит в интервале [0,1], и у нас тогда, может быть, больше оснований считать, что $[c, f] = [0,1]$.

Представление графического подхода к решению задачи

Представим графический подход решения задачи, рассматривая случай 3.2) и ситуацию (2,2).

Для решения надо, во-первых, для каждой ситуации (i, j), $i = 1, 2, 3, 4$, $j = 1, 2, 3, 4$ найти игру, для которой значение функции принадлежности максимально среди всех таких игр, для которых рассматриваемая ситуация является седловой точкой. Ну и, во-вторых, выбрать ситуацию, для которой найденное значение функции принадлежности максимально.

Обозначим:

$$\mu_{ij}^{l}(u) = \frac{u - c_{ij}}{d_{ij} - c_{ij}}, c_{ij} \le u \le d_{ij}$$

и (9)

$$\mu_{ij}^{r}(u) = \frac{f_{ij} - u}{f_{ij} - d_{ij}}, d_{ij} \le u \le f_{ij}.$$

Напомним, что для того, чтобы ситуация (i_0, j_0) была равновесной, надо, чтобы выигрыш в этой ситуации не превосходил выигрышей в ситуациях $(i_0, j), j = 1, 2, 3, 4$ (по строке) и был бы не меньше выигрышей в ситуациях $(i, j_0), i = 1, 2, 3, 4$ (по столбцу).

Рассмотрим ситуацию (2,2). Чтобы получить из игры с матрицей выигрышей, заданной таблицей 1, игру, в которой ситуация (2,2) является ситуацией равновесия, надо изменить в сторону неуменьшения выигрыши в третьей строке и в сторону неувеличения — выигрыши из четвертого столбца. Выигрыш же в ситуации (2,2), может быть, надо оставить без изменения, может быть, увеличить, а может быть уменьшить.

Построим необходимые графики функций $\mu_{i2}^{l}(u)$, $i = 1,2,3,4$ и $\mu_{2j}^{r}(u), j = 1,2,3,4$ (рис. 1).

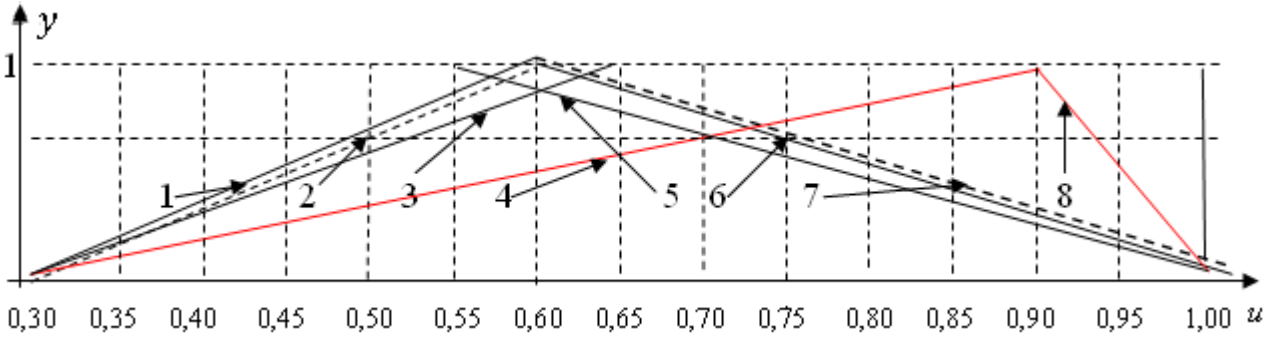

Рис. 1. Функции принадлежности $\mu_{i2}^{l}(u)$ и $\mu_{2j}^{r}(u)$

На рисунке 1 цифрами обозначены следующие графики функций: 1 — $y = \mu_{12}^{l}(u)$; 2 — $y = \mu_{42}^{l}(u)$; 3 — $y = \mu_{32}^{l}(u)$; 4 — $y = \mu_{22}^{l}(u)$; 5 — $y = \mu_{24}^{r}(u)$; 6 — $y = \mu_{21}^{r}(u)$; 7 — $y = \mu_{23}^{r}(u)$; 8 — $y = \mu_{22}^{r}(u)$.

Если ситуация (2,2) является равновесной при выигрыше в ней, равном b, то на оси абсцисс существует точка с абсциссой (выигрышем в ситуации (2,2) при надежности 0), большей выигрышей (при надежности 0) по столбцу и меньшей выигрышей (при надежности 0) по строке. На оси абсцисс находим отрезок, для точек которого выполняются указанные неравенства. На рисунке 1 это точки оси абсцисс от 0,3 до 1.

Над этим отрезком находим точку, являющуюся точкой пересечения возрастающего и убывающего графиков. Если таких точек несколько, выбираем из них точку с минимальной ординатой. Абсцисса этой точки дает искомый выигрыш в рассматриваемой ситуации, а ее ордината равна максимальной надежности того, что эта ситуация является равновесной.

На рисунке искомая точка лежит на пересечении графиков четвертой и пятой функций. Их уравнения соответственно имеют вид:

$$y = \frac{u-0,3}{0,9-0,3}\ ;\ y = \frac{1-u}{1-0,55} \qquad (10)$$

Решая эту систему, находим

$$y \cong 0,67, x \cong 0,70\ ,$$

следовательно, надежность того, что ситуация (2,2) (DDOS атака хакеров — стратегия защиты — межсетевое экранирование) является седловой точкой, равна 0,67, с этой надежностью равновесное значение вероятности обнаружения и предотвращения преступных действий равно 0,70.

Заключение

В качестве заключения отметим, что достоинством предлагаемого подхода является то, что любая игра имеет решение в чистых стратегиях, чего нельзя сказать о классическом подходе.

Литература

1. Петросян Л. А. Теория игр: учеб. пособие для ун-тов / Л. А. Петросян, Н. А. Зенкевич, Е. А. Семина. — М.: Высшая школа: Книжный дом «Университет», 1998. — 304 с.

2. Andreev V. P. Information Security of Automated Working Places in Case of Emergencies / V. P. Andreev, A. I. Dergachev, A. K. Chernykh // Интеллектуальные технологии на транспорте. 2019. № 1 (17). С. 27–32.

3. Гладких В. П. Особенности моделирования системы информационной безопасности в органах военного управления / В. П. Гладких, В. Г. Швед, А. И. Дергачёв // Национальные приоритеты России. Серия 1: Наука и военная безопасность. 2015. № 1 (1). С. 47–49.

4. Кофман А. Введение в теорию нечетких множеств = Introduction a la théorie des sous-ensembles flous. A l`usage des ingénieurs (Fuzzy Sets Theory) / Пер. с фр. В. Б. Кузьмина; под ред. С. И. Травкина. — М.: Радио и связь. Редакция литературы по кибернетике и вычислительной технике, 1982. — 432 с.

5. Штовба С. Д. Введение в теорию нечетких множеств и нечеткую логику. — Винница: УНИВЕРСУМ-Винница, 2001. — 756 с.

6. Заде Л. А. Понятие лингвистической переменной и его применение к принятию приближенных решений = The concept of a linguistic variable and its application to approximate reasoning / Пер. с англ. Н. И. Ринго; под ред. Н. Н. Моисеева и С. А. Орловского. — М.: Мир, 1976. — 165 с. — (Математика. Новое в зарубежной науке. Вып. 3).

7. Орловский С. А. Проблемы принятия решений при нечеткой исходной информации. — М.: Наука. Главная редакция физико-математической литературы, 1981. — 208 с. — (Оптимизация и исследование операций).

8. Нечеткие множества в моделях управления и искусственного интеллекта / А. Н. Аверкин, И. З. Батыршин, А. Ф. Блишун [и др.]; под ред. Д. А. Поспелова. — М.: Наука. Главная редакция физико-математической литературы, 1986. — 312 с. — (Проблемы искусственного интеллекта).

Evaluation of the Effectiveness of an Intrusion Detection System Based on Matrix Games and Fuzzy Sets

PhD V. B. Vilkov
Military Educational Institution
of Logistics named after General
of the Army A. V. Khrulyov
of the Ministry of Defense
of the Russian Federation
St. Petersburg, Russia
amirusha@rambler.ru

PhD A. K. Chernykh
Saint Petersburg Military
Order of Zhukov Institute
of the National Guard Troops
of the Russian Federation
St. Petersburg, Russia
nataliachernykh@mail.ru

PhD A. I. Dergachev,
PhD O. N. Kuranova
Emperor Alexander I
Petersburg State Transport University
St. Petersburg, Russia
d_ader@mail.ru
olga_kuranova@mail.ru

Abstract. **The article considers a problem formulated as a matrix game, in which the gain of officials using a specific system for detecting criminal actions of intruders (player 1) is the probability of timely detection of these criminal actions of intruders (player 2). As a rule, it is not possible to set unambiguously the probability of timely detection of criminal acts, so it is proposed to use the apparatus of fuzzy set theory for its evaluation. The article discusses and reveals the basic concepts of the theory of fuzzy sets, as well as an example of the practical application of this theory to assess the effectiveness of the system of detection of criminal acts of intruders. The use of fuzzy sets theory in the evaluation of possible actions of an attacker will allow to detect existing vulnerabilities in the information security of an automated system, and subsequently to improve the detection systems of criminal actions of intruders (hackers), preventing the possibility of causing economic and other damage to the company.**

Keywords**: system of detection of criminal actions of intruders, matrix game, fuzzy sets, matrix game solution.**

REFERENCES

1. Petrosyan L. A., Zenkevich N. A., Semina E. A. Games theory: Study guide [Teoriya igr: Uchebnoe posobie dlya universitetov], Moscow, 1998, 304 p.

2. Andreev V. P., Dergachev A. I., Chernykh A. K. Information Security of Automated Working Places in Case of Emergencies, *Intellectual Technologies on Transport [Intellektual'nye tekhnologii na transporte]*, 2019. № 1(17). Pp. 27–32.

3. Gladkih V. P., Shved V. G., Dergachev A. I. Features of Modelling of Information Security System in Bodies of Military Administration [Osobennosti modelirovaniya sistemy informatsionnoy bezopasnosti v organakh voennogo upravleniya], *National Priorities of Russia. Series 1: Science and Military Security [Natsional'nye prioritety Rossii. Seriya 1: Nauka i voennaya bezopasnost']*, 2015, No. 1 (1), Pp. 47–49.

4. Kaufmann A. Introduction a la théorie des sous-ensembles flous. A l`usage des ingénieurs (Fuzzy Sets Theory) [Vvedenie v teoriyu nechetkikh mnozhestv], Moscow, Radio and Communication Publishing House, 1982, 432 p.

5. Shtovba S. D. Introduction to fuzzy set theory and fuzzy logic [Vvedenie v teoriyu nechetkikh mnozhestv I nechetkuyu logiku], Vinnytsia, UNIVERSUM-Vinnytsia Publisher, 2001, 756 p.

6. Zadeh L. A. The concept of a linguistic variable and its application to approximate reasoning [Ponyatie lingvistich-eskoy peremennoy i ego primenenie k prinyatiyu priblizhennykh resheniy], Moscow, Mir Publishers, 1976, 165 p.

7. Orlovsky S. A. Decision problems in case of fuzzy input information [Problemy prinyatiya resheniy pri nechetkoy iskhodnoy informatsii], Moscow, Nauka Publihers, 1981, 208 p.

8. Averkin A. N., Batyrshin I. Z., Blishun A. F., et al. Fuzzy sets in management models and artificial intelligence [Nechetkie mnozhestva v modelyakh upravleniya i iskusstvennogo intellekta], Moscow, Nauka Publishers, 1986, 312 p.

Оригинальный английский текст © A. A. Tyugashev, A. P. Dolgintsev, I. A. Molodkin, S. E. Adadurov опубликован в Proceedings of Models and Methods of Information Systems Research Workshop 2019 (MMISR 2019). CEUR Workshop Proceedings. 2020. Vol. 2556. С. 80–84.

К вопросу об адаптивном устойчивом управлении сложными системами в транспортной отрасли

А. А. Тюгашев, А. П. Долгинцев
Самарский государственный университет путей сообщения
Самара, Россия
a.tyugashev@samgups.ru,
dolgintsev@rambler.ru

И. А. Молодкин
Петербургский государственный университет путей сообщения Императора Александра I
Санкт-Петербург, Россия
imolodkin@pgups.ru

С. Е. Ададуров
АО «Научно-исследовательский институт железнодорожного транспорта» (АО «ВНИИЖТ»)
Москва, Россия

Аннотация. **Средства железнодорожного, морского и авиационного транспорта, космические аппараты могут рассматриваться как характерные примеры сложных технических систем, нуждающихся в согласованном и устойчивом управлении в режиме реального времени, часто в условиях непредсказуемо меняющихся внешних условий. Статья посвящена выявлению основных проблем построения соответствующих систем управления, а также возможным подходам к синтезу и верификации средств управления, обладающих необходимыми свойствами. Представлено описание некоторых помогающих в этом программных инструментов.**

Ключевые слова: **STM-решатели, системы управления, логика управления, адаптивное управление.**

Введение

В настоящее время сложные технические комплексы используются в самых разных областях. Мы можем упомянуть в этой связи железнодорожный транспорт, автоматизированные производства, атомные электростанции, космические аппараты [1] и т. д. Подобные системы характеризуются некоторыми важными общими чертами. Например, многоуровневая иерархическая организация: система состоит из подсистем, каждая из которых, в свою очередь, — из десятков приборов, агрегатов, датчиков и других устройств. Другая важнейшая особенность — активное и адаптивное поведение в непредсказуемой внешней среде. Можно отметить, что важнейшей особенностью работы большинства сложных технических систем в настоящее время является режим реального времени — наличие временных ограничений [2].

Важнейшей и непростой проблемой является обеспечение подобного сложного технического комплекса соответствующими средствами управления, гарантирующими выполнение поставленных перед ним целевых задач. Статья посвящена попытке анализа различных сторон проблемы построения подобного адаптивного и устойчивого управления и возможным путям решения данной проблемы.

Каждая техническая система создается для решения каких-либо задач. Транспортные средства должны обеспечивать доставку пассажиров и грузов из пункта отправления в пункт назначения. Электростанция — вырабатывать энергию. Автоматизированная производственная линия нужна для изготовления продукции и т. п. Обычно присутствует главная цель, или задача системы, однако чтобы достичь ее, как правило, предварительно необходимо реализовать набор подготовительных процессов и процедур. Другие операции должны быть выполнены после выполнения основной задачи (например, чистка салона самолета после перелета). Соответственно, мы можем выделить набор целей, которых должна достигать система. При этом нередко с этими целями связаны крайние сроки (заранее заданные моменты времени). Во многих случаях необходимо не просто достигнуть неких целей к заданным моментам времени, но реализовать целую последовательность логически связанных и выстроенных во времени операций, многие из коих должны при этом выполняться параллельно. Некоторые из них имеют ненулевую длительность, представляя собой протяженные во времени процессы, соответственно, мы должны иметь адекватные средства для их моделирования. Фактически, мы можем говорить, что сложная система должна реализовать некий «план», или «расписание», состоящее из взаимоувязанных процессов. В космической отрасли подобные планы принято называть циклограммами.

В ряде случае существуют ограничения физической природы, не позволяющие некоторым процессам накладываться друг на друга, или же задающие требование по синхронизации моментов начала и окончания определенных процессов. Логика управляющих алгоритмов RTCAL, разработанная авторами ранее [3], предоставляет изобразительные средства для описания подобных требований.

Дополнительный аспект сложности управляющей логики связан с возможностью наступления непредсказуемых событий во внешней среде и внутри системы. Данные события

вызывают необходимость соответствующей реакции, включая адаптацию системного плана.

При графическом представлении системного плана факт, что конкретный процесс выполняется не всегда, а лишь в специфичной ситуации (например, при наступлении определенного события), может отражаться тем или иным цветом (рис. 1) [4].

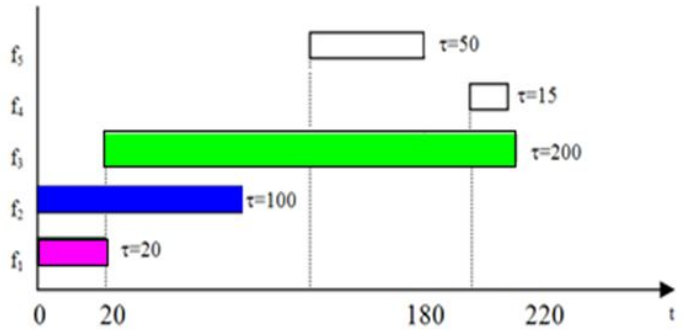

Рис. 1. Циклограмма (системный план)

Естественно, адаптивное устойчивое управление должно быть надежным. Одним из нежелательных, но вполне возможных событий может быть полный или частичный отказ того или иного входящего в состав системы оборудования. Это не должно привести к тяжелым последствиям, человеческим жертвам, авариям и катастрофам, в идеальном случае отказ не должен влиять на успешное выполнение возложенных на систему задач [5].

Это возможно за счет использования избыточности. При проектировании сложной технической системы избыточность в нее закладывают изначально. Наиболее важные устройства и агрегаты могут дублироваться, и, если основное будет повреждено или выйдет из строя, система управления должна вовремя определить аномальную ситуацию и осуществить переключение на резерв. Иными словами, система управления должна поддерживать важнейшее свойство кибернетических систем — реконфигурацию, или адаптацию структуры. Еще один успешно используемый путь повышения надежности [1] — использование функциональной избыточности. Например, некоторая система, или устройство, входящее в состав сложной системы, утрачивает способность полноценно реализовывать свои функции, но при этом часть из них (или полностью, пусть и с потерей качества или точности) может взять на себя другая подсистема или прибор.

Для поддержания возможности использования функциональной избыточности средства управления сложной технической системой должны иметь внутреннее «понимание» функциональности и взаимозаменяемости установленного оборудования и степень востребованности различных видов функций бортовых устройств при выполнении системой ее целевых задач.

Как правило, в современных сложных технических системах управление реализуется на основе встроенных цифровых ЭВМ, зачастую даже с использованием бортовой вычислительной сети. А непосредственно реализация логики управления возлагается на программное обеспечение (ПО). В аэрокосмической отрасли подобное ПО называется «flight control software». Это — особый вид управляющих программ, основные действия которого сводятся к выдаче команд управления бортовым устройствам путем посылок по специальным интерфейсам кодовых последовательностей электрических импульсов. Примерами подобных команд могут быть «Немедленно активировать основную систему торможения» или «Отключить второй гироскоп». Замечательная особенность программного обеспечения, представляющего собой фактически разновидность информации, сохраняемой в памяти управляющей ЭВМ, заключается в возможности внесения в него изменений в ходе эксплуатации. За счет этого мы иногда можем компенсировать даже не предусмотренные в ходе инженерного анализа при проектировании аномальные ситуации, связанные со сложными отказами аппаратуры [6].

Необходимо отметить и еще одно обстоятельство. С одной стороны, система подвергается воздействию внешней среды. С другой — сама техническая система имеет влияние на среду. Влияние носит взаимный характер, и может образовывать цепочки взаимосвязанных воздействий. Таким образом, помимо всего перечисленного, устойчивое гибкое управление должно обеспечивать, чтобы система не причинила вреда окружению (например, вредные выбросы находились в допустимых границах).

Адаптивное устойчивое управление с этих позиций подразумевает не только внутреннюю безопасность или устойчивое сохранение системой своей работоспособности, но и безопасность внешней среды, системного окружения. Если говорить о транспортных средствах, то они по определению несут в себе потенциальную опасность пешеходам, другому транспорту и т. п.

Еще одним контролируемым параметром (набором параметров), за которым должна следить система управления, является расходование доступных системе ресурсов. Каждое устройство, входящее в состав сложной технической системы, потребляет определенные ресурсы, например электроэнергию или топливо. Многие из устройств могут иметь набор различных режимов с разным уровнем потребления ресурсов.

Вышесказанное приводит к необходимости отражения в средствах управления системой внутреннего представления целого ряда аспектов, относящихся к ее функционированию, целям, внешним событиям и т. д. Мы должны иметь представление системного плана (набора планов для различных вариантов достижения целей), набор переменных для отражения текущего состояния системного оборудования и внешней среды, а также иметь представление о времени и пространстве для оценки уже выполненного системой, ее местоположения, того, что предстоит выполнить и куда переместиться.

Это можно считать примером воплощения известного кибернетического закона необходимого разнообразия Эшби: чем более сложным является объект управления и его поведение, тем более сложной и имеющей разнообразное поведение должна быть система управления.

Описание метода

Выделим еще раз важнейшие черты адаптивного устойчивого управления сложными техническими системами в режиме реального времени:

- присутствие внутреннего представления («отражения») внешней среды, текущего состояния самой системы, включая информацию о доступной функциональности различного оборудования — «картины себя», а также знаний о системном плане, подлежащем реализации, включая уже выполненные задачи и еще не выполненные;
- возможность гибкой реконфигурации при изменившихся внешних условиях или в случае аномальной ситуации, вызванной отказом аппаратуры;
- логика управления, включающая правила с учетом функционирования в режиме реального времени.

Дружинин и Конторов [7] приводят в своей книге классификацию кибернетических систем по уровням сложности:

- детерминированные S_1 системы с жестким (постоянным) законом преобразования входа X в выход Y;
- стохастические S_2 системы с существенным влиянием случайных факторов на результат;
- системы класса S_3 без четко сформулированных правил преобразования входа в выход;
- системы класса S_4, активно достигающие заранее поставленные цели;
- Системы S_0, обладающие возможностями самостоятельного выбора целей, изменения собственной структуры и адаптивной реакцией на внешние стимулы.

Используя данную классификацию, мы можем отнести рассматриваемые в настоящей статье сложные технические системы к классу S_0.

Причины этого в следующем. Во-первых, нам нужна гибкая логика управления, учитывающая различные ситуации и корректную их обработку управляющим программным обеспечением. Во-вторых, выше мы заявили наличие требования о самодиагностике и реконфигурации. Наконец, системный план может быть изменен или уточнен в ходе его выполнения.

Итак, нам необходимы средства для описания подобной управляющей логики. Какими путями мы можем это сделать?

Когда речь заходит о «логике», обычно подразумевают наличие набора аксиом (исходных посылок), правил вывода и средств записи формул. Используя правила вывода, задействуется потенциал механизма рассуждений. Что понимать под правилами специфической логики — логики управления сложной системой в реальном режиме времени?

Правила часто записывают в форме «ЕСЛИ {набор посылок-антецедентов} ТО {набор выводов}». При этом и антецеденты, и заключения равноправны и выражают некоторые утверждения. В нашем случае логики управления более уместным видится следующее представление:

$$\alpha_1(t_{u1}) \wedge \neg\, \alpha_2(t_{u2}) \wedge \ldots\ \alpha_M(t_{uM}) \longrightarrow A_1(t_{a1}) \wedge A_2(t_{a2}) \wedge \ldots A_N(t_{aN}) \quad (1)$$

В левой части правила мы видим набор логических переменных α_j с подразумеваемым по умолчанию значением ИСТИННО или явно заданным отрицанием, а в правой части мы видим набор действий A_i, причем с привязкой ко времени. Поскольку некоторые из осуществляемых действий влияют на истинность α_j (на самом деле, в системе управления есть и действие, непосредственно связанное с «установкой» или «сбросом» некоего признака или «флага»), возможность проведения пошаговых заключений сохраняется.

Возможность наличия нескольких правил с совпадающей правой частью можно считать эквивалентом операции логическое «ИЛИ». В левой части правил — конъюнкция вхождений логических переменных с наличием или отсутствием логического отрицания, подобным образом в силу известного универсализма ДНФ возможно представление произвольных условий в качестве посылок.

Проблема построения правил логики управления с сохранением свойств адаптивности и устойчивости заключается в построении следующих переходов. Во-первых, от системных целей (задач) к набору системных планов для разных вариантов развития событий и сценариев управления с привязкой ко времени. Далее — от плана выполнения задач к графику необходимой функциональности. Например, транспортное средство нуждается в навигации, движителе, возможностях в области связи. Навигация может осуществляться с помощью инерциальных приборов, с задействованием спутниковых систем GPS/ГЛОНАСС и пр. Это дает возможность запланировать использование различных видов оборудования и аппаратуры на различных временных отрезках в ходе эксплуатации системы.

На данном этапе производится переход от необходимой функциональности к циклограмме работы конкретных приборов и устройств.

Из циклограммы в силу ее семантики можно извлечь правила управляющей логики в виде (1).

На следующем шаге, зная потребление приборами ресурсов в разных режимах, мы осуществляем переход к графику расходования ресурсов.

Правила управляющей логики реального времени затем реализуются тем или иным способом (например, либо ручным программированием, либо автоматической генерацией исходных текстов управляющих программ [8]) в средствах управления сложной системой.

Существует и важная обратная задача, или проблема обратного проектирования. Она заключается в том, действительно ли управляющая логика, уже реализованная в управляющем программном комплексе, соответствует всем вышеприведенным требованиям. Для решения обратной задачи нам необходим переход от существующих программных модулей к правилам управляющей логики. Мы можем использовать специальные анализаторы, позволяющие распознавать управляющие конструкции, такие как выдача команды управления на бортовую аппаратуру, запуск другой программы на выполнение или установку/сброс признака, и «извлекать» тем самым правила логики управления.

Устойчивость логики управления может рассматриваться в нескольких аспектах:

- соответствие необходимым требованиям синхронизации, например в терминах [2] $f_1 << f_2$ — необходимость

предшествования начала выполнения процесса f_1 началу f_2, f_3 *CH* f_5 — необходимость совмещения процессов f_3 и f_5 по началу, $f_1 \rightarrow f_5 \rightarrow f_7$ — требование обеспечить непосредственное следование выполнения процессов f_1, f_5, f_7 один по окончании другого, запрещение наложения процессов вида $f_{11} <> f_8$ (запрет наложения может быть вызван, например, физическими причинами — выдвинутая антенна может мешать оптическим приборам и пр.);

- соблюдение требований по уровням доступных ресурсов и допустимому влиянию на окружающую среду;
- наличие в нужные моменты времени всех видов требуемой функциональности для достижения поставленных перед системой целей;
- надлежащий уровень надежности, или обеспечения выполнения системных задач, несмотря на возможные отказы бортовых систем и непредвиденные ситуации.

В соответствии с изложенным, модель сложной технической системы может быть представлена как

$$\{BA, G, FS, CL, RS, EM, CA, SC\}, \qquad (2)$$

где *BA* — набор входящих в систему устройств (бортовая аппаратура), каждое из которых имеет набор допустимых режимов работы;
G — набор системных целей, с установленными временами их достижения;
FS — виды присутствующей в системе функциональности;
CL — управляющая логика в виде набора правил вида (1);
RS — набор ресурсов, доступных системе и оказывающих влияние на достижение ею поставленных целей, с заданными предельными доступными уровнями;
EM — набор воздействий системы на среду, также с заданными предельными границами;
CA — комплекс управляющего программного обеспечения;
SC — набор временных ограничений на функционирование системы.

Модель бортовой аппаратуры может строиться в виде алгебраической системы, как показано в работе [2], с отношением вхождения прибора или устройства в состав той или иной системы, а систем — в общий состав оборудования транспортного средства; отношением соответствия между устройством и его режимами работы; отношением между режимами работы и предоставляемой прибором функциональностью; отношением между режимами работы и потребляемыми ресурсами/воздействием на окружающую среду в данном режиме.

Для решения задачи проверки средств управления на выполнение необходимых требований, приведенных в настоящей статье, на этапе проектирования, а также для анализа и оценки уже существующих систем управления, мы можем использовать моделирование на ЭВМ с использованием различных средств — как самостоятельно «с нуля» разработанных программных инструментов, так и с привлечением возможностей существующих пакетов моделирования.

Для решения задачи синтеза логики управления, гарантирующей набор ограничений со стороны ресурсов, авторами произведена попытка использовать возможности современных SMT-решателей (подход Satisfiability Modulo Theories) [9]. Экран соответствующего программного прототипа представлен на рисунке 2.

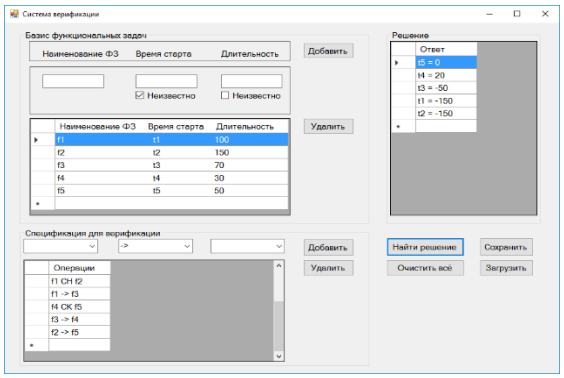

Рис. 2. Экран программного прототипа

Ранее нами также были успешно применены логическое программирование и возможности языка Пролог для проверки свойств синхронизации управляющих алгоритмов реального времени [8].

Проблема верификации существующей управляющей логики сводится в данному случае к проверке: 1) заданный набор правил действительно воплощает системный план с гарантией соблюдения временных ограничений; 2) границы потребления ресурсов и воздействия на окружающую среду не превышены; 3) в случае возникновения отказов оборудования логика управления своевременно выявит аномальную ситуацию и произведет переключение на резервную аппаратуру.

Заключение

В статье представлены средства описания логики управления сложными техническими системами транспортной отрасли в реальном времени. Авторами рассмотрены фундаментальные проблемы, связанные с формированием «устойчивой», или «согласованной», логики управления в условиях ограниченности доступных бортовых ресурсов. Рассмотрены важнейшие проблемы устойчивого управления — проблема верификация и проблема синтеза.

Будущие работы включают создание специализированных инструментальных программных средств, позволяющих автоматизировать верификацию логики управления, «зашитой» в коде управляющих программ, а также попытку использования возможностей программирования в ограничениях (constraint programming).

Благодарности

Авторы выражают благодарность коллегам из РКЦ «Прогресс», г. Самара, за многолетнее сотрудничество в области программной инженерии критически важного программного обеспечения, а также основателю данного научного направления профессору Анатолию Алексеевичу Калентьеву.

Литература

1. Управление космическими аппаратами зондирования Земли: Компьютерные технологии / Д. И. Козлов,

Г. П. Аншаков, Я. А. Мостовой, А. В. Соллогуб. — М.: Машиностроение, 1998. — 368 с.

2. Тюгашев А. А. Интегрированная среда для проектирования управляющих алгоритмов реального времени // Известия Российской академии наук. Теория и системы управления. 2006. № 2. С. 128–141.

3. Калентьев А. А. ИПИ/CALS-технологии в жизненном цикле комплексных программ управления / А. А. Калентьев, А. А. Тюгашев. — Самара: Изд-во Самарского научного центра РАН, 2006. — 266 с.

4. Tyugashev A. A. Language and Toolset for Visual Construction of Programs for Intelligent Autonomous Spacecraft Control // IFAC-PapersOnLine. 2016. Vol. 49, Is. 5. Pp. 120–125. DOI: 10.1016/J.IFACOL.2016.07.100.

5. Вычислительный алгоритм формирования программного движения в программном повороте малого космического аппарата / В. В. Салмин, А. В. Филатов, И. С. Ткаченко [и др.] // Вестник Самарского государственного аэрокосмического университета имени академика С. П. Королёва (национального исследовательского университета). 2015. Т. 14, № 2. С. 9–19. DOI: 10.18287/2412-7329-2015-14-2-9-19.

6. Sygurov Yu. M. Method for Modeling of Spacecraft Onboard Apparatus and Building of Consistent Control Logic with Limited Onboard Resources / Yu. M. Sygurov, A. A. Tyugashev // Journal of Physics: Conference Series. 2019. Vol. 1368. 042032. 8 p. DOI: 10.1088/1742-6596/1368/4/042032.

7. Дружинин В. В. Проблемы системологии: Проблемы теории сложных систем / В. В. Дружинин, Д. С. Конторов. — М.: Советское радио. Редакция кибернетической литературы, 1976. — 296 с.

8. Tyugashev A. Visual Builder of Rules for Spacecraft Onboard Real-Time Knowledge Base // Intelligent Decision Technologies 2016. Proceedings of the 8th KES International Conference on Intelligent Decision Technologies (KES-IDT 2016), (Puerto de la Cruz, Spain, June 15-17, 2016). Part II. / I. Charnovski, A. M. Caballero, et al. (eds) // Smart Innovation, Systems and Technologies. Vol. 57. Pp. 189–205. DOI: 10.1007/978-3-319-39627-9_17.

9. Tyugashev A. A. Application of SMT Solvers for Evaluation of Real-Time Control Logic of Spacecraft. // Journal of Physics: Conference Series. 2018. Vol.1096. 012156. 7 p. DOI: 10.1088/1742-6596/1096/1/012156.

Original English text © A. A. Tyugashev, A. P. Dolgintsev, I. A. Molodkin, S. E. Adadurov published in Proceedings of Models and Methods of Information Systems Research Workshop 2019 (MMISR 2019), CEUR Workshop Proceedings, 2020, Vol. 2556, Pp. 80–84.

Problems of Building the Intelligent Consistent Control Logic for Complex Technical Systems in Transport Industry

A. A. Tyugashev,
A. P. Dolgintsev
Samara State Transport University
Samara, Russia
a.tyugashev@samgups.ru,
dolgintsev@rambler.ru

I. A. Molodkin
Emperor Alexander I St. Petersburg
State Transport University
Saint Petersburg, Russia
imolodkin@pgups.ru

S. E. Adadurov
Railway Research Institute
of JSC Russian Railways
(JSC VNIIZhT)
Moscow, Russia

Abstract. **We can review Railroad Transportation, Aerial manned and unmanned vehicles, and Spacecrafts as examples of a complex technical system. Their subsystems contain many devices, sensors, and other equipment. There is an important problem how to build the intelligent real-time computer-based control logic for such complex of the subsystems. The paper is devoted to this problem. We focus on mathematical modeling and finding the ways of synthesis and verification of consistent control logic. The paper also presents some software tools developed by the authors.**

Keywords: **STM-solvers, control systems, control logic, adaptive control.**

REFERENCES

1. Kozlov D. I., Anshakov G. P., Mostovoy Ya. A., Sollogub A. V. Control of Earth observation spacecrafts: Computer technologies [Upravlenie kosmicheskimi apparatami zondirovaniya Zemli: Komp'yuternye tekhnologii]. Moscow, Mashinostroenie Publishers, 1998, 368 p.

2. Tyugashev A. A. Integrated Environment for Designing Real-Time Control Algorithms, *Journal of Computer and Systems Sciences International*, 2006, Vol. 45, No. 2, Pp. 287–300.
DOI: 10.1134/s1064230706020134.

3. Kalentyev A. A., Tyugashev A. A. CALS technology in lifecycle of complex control programs [IPI/CALS tekhnologii v zhiznennom tsikle kompleksnykh programm upravleniya]. Samara, Samara Scientific center of Russian Academy of Sciences, 2006, 266 p.

4. Tyugashev A. A. Language and Toolset for Visual Construction of Programs for Intelligent Autonomous Spacecraft Control, *IFAC-PapersOnLine*, 2016, Vol. 49, Is. 5, Pp. 120–125.
DOI: 10.1016/J.IFACOL.2016.07.100.

5. Salmin V. V., Filatov A. V., Tkachenko I. S., et al. Computational Algorithm of Forming Program Motion in a Scheduled Turn of Small Spacecraft [Vychislitel'nyy algoritm formirovaniya programmnogo dvizheniya v programmnom povorote malogo kosmicheskogo apparata], *Vestnik of the Samara State Aerospace University* [*Vestnik Samarskogo gosudarstvennogo aerokosmicheskogo universiteta imeni akademika S. P. Koroleva (natsional'nogo issledovatel'skogo universiteta)*], 2015, Vol. 14, No. 2. Pp. 9–19.
DOI: 10.18287/2412-7329-2015-14-2-9-19.

6. Sygurov Yu. M., Tyugashev A. A. Method for Modeling of Spacecraft Onboard Apparatus and Building of Consistent Control Logic with Limited Onboard Resources, *Journal of Physics: Conference Series*, 2019, Vol. 1368, 042032, 8 p.
DOI: 10.1088/1742-6596/1368/4/042032.

7. Druzhinin V. V., Kontorov D. S. Problems of systemology: Problems of theory of complex systems [Problemy sistemologii: Problemy teorii slozhnykh sistem], Moscow, Soviet Radio Publishers, 1976, 296 p.

8. Tyugashev A. Visual Builder of Rules for Spacecraft Onboard Real-Time Knowledge Base. *In: Charnovski I., Caballero A. M., et al. (eds) Intelligent Decision Technologies 2016. Proceedings of the 8th KES International Conference on Intelligent Decision Technologies (KES-IDT 2016), Part II, Puerto de la Cruz, Spain, June 15-17, 2016. Smart Innovation, Systems and Technologies*, Vol. 57, Pp. 189–205.
DOI: 10.1007/978-3-319-39627-9_17.

9. Tyugashev A. A. Application of SMT Solvers for Evaluation of Real-Time Control Logic of Spacecraft, *Journal of Physics: Conference Series*, 2018, Vol.1096, 012156, 7 p.
DOI: 10.1088/1742-6596/1096/1/012156.

Оценка гиперпараметров при анализе тональности русскоязычного корпуса текстов

аспирант Н. Е. Косых
Петербургский государственный университет путей сообщения Императора Александра I
Санкт-Петербург, Россия
nikitosagi@mail.ru

Аннотация. **Предложен подход к подбору наиболее производительного набора параметров для объекта классификатора текста. Для вычислений использован облачный сервис Google Colaboratory, выполняющий код на языке Python внутри браузера, используя виртуальные аппаратные ресурсы. Рабочая среда включает компоненты для разработки модели машинного обучения: библиотеку scikit-learn, содержащую методы обучения и их оценки; библиотеку для анализа и обработки данных — Pandas; библиотеку регулярных выражений – Re. Основное внимание уделяется точности в процессе прогнозирования тональности русскоязычного корпуса текстов. Для решения задачи биномиальной классификации текста использован наивный байесовский классификатор. Применение программного подхода к поиску оптимальных значений гиперпараметров по сетке позволяет улучшить точность работы классификатора на ограниченном наборе данных.**

***Ключевые слова*: классификация, гиперпараметры, точность, обучение с учителем, тональность.**

Введение

Машинное обучение — это подпространство в области компьютерных наук, где компьютер обучается выполнять определенные задачи, имея модель и набор входных данных, без необходимости в явном наборе правил.

Модель машинного обучения состоит из следующих элементов:

1. Математическое описание задачи машинного обучения.
2. Тренировочный алгоритм для поиска решений задачи.
3. Набор гиперпараметров, значения которых задается до начала обучения и не изменяется в процессе.

Как правило, гиперпараметры описывают способ решения или его структуру. Например, это может быть количество итераций, необходимых для прохода алгоритма обучения. Процесс выбора оптимальных значений гиперпараметров [1] состоит в том, чтобы указать диапазон допустимых значений, которые дают лучший результат в соответствии с критериями оценивания. В каждом конкретном случае выбора тренировочного алгоритма набор гиперпараметров может отличаться.

На практике стандартные алгоритмы машинного обучения обычно реализуются через библиотеки с открытым исходным кодом. Таким образом, пользователю остается только выбрать модель и указать диапазон гиперпараметров.

Контролируемое обучение

Модель машинного обучения анализируется данные и ищет закономерности. Множество факторов также влияют на успех в процессе распознавания шаблонов данных, а именно:

a) количество данных;
b) качество данных;
c) вычислительные мощности.

В рамках нашей задачи мы будем использовать один из подходов машинного обучения, а именно обучение с учителем (контролируемое обучение). Задача в настройке контролируемого обучения состоит в том, чтобы создать модель, способную предсказать ответ y_i на основании входных характеристик $x_i, i = 1, 2 \dots, n$. В нашем случае y_i рассматриваем как категориальную переменную, а решаемая задача — классификация.

В контролируемом обучении, как правило, данные разбиваются на два подмножества: обучающее и тестовое, и они используются для обучения модели. Первый набор данных задан значением и определенным классом; контролируемая модель обучения пытается найти шаблоны, которые можно использовать для прогнозирования ответов. Далее обученная модель проверяется на тестовой выборке, которая не содержит предопределенных меток классов, для оценки точности обучения.

Одним из показателей для оценки качества классификации является общая точность (AP), то есть доля правильно спрогнозированных значений в общей выборке. Вычисляется по формуле

$$accuracy(y, \hat{y}) = \frac{1}{n}\sum_{i=0}^{n-1} 1(\hat{y}_i = y_i)\,, \qquad (1)$$

где n — количество образцов;
$\hat{y}_i$ — прогнозируемое значение класса;
y_i — истинное значение класса.

Для задачи классификации был выбран корпус коротких текстов Юлии Рубцовой [2], сформированных на основе русскоязычных сообщений из Twitter. Корпус состоит из двух наборов помеченных данных, содержащих более 100 тысяч положительных и отрицательных записей.

В условиях ограниченности вычислительных мощностей для вычислений используем облачный сервис Google Colaboratory, который позволит нам выполнять код, написанный на Python, внутри браузера, используя виртуальные аппаратные ресурсы для обработки данных. Рабочая среда уже включает все необходимые компоненты для разработки модели машинного обучения: библиотеку scikit-learn [3], содержащую методы машинного

обучения и их оценки; библиотеку для анализа и обработки данных — Pandas; библиотеку регулярных выражений — Re.

Для чистоты эксперимента получим среднюю гармоническую величину по количеству записей в каждом наборе до соотношения 1 к 1.

Подготовка данных

Далее представим нашу обучающую модель в виде детерминированной последовательности действий.

1. Оценка зашумленности текста. Этап необходим для определения методов предварительной подготовки текста от посторонних, мешающих обучению — знаки препинания, цифры, иностранные литералы. В ином случае обученная модель может установить неправильные семантические зависимости.

2. Реализация функции предварительной обработки текста представлена на рисунке 1. Характер преобразований зависит исключительно от характера последовательностей, содержащихся в тексте.

```
def preprocess_text(text):
    text = text.lower().replace("ё", "е")
    text = re.sub('((www\.[^\s]+)|(https?://[^\s]+))', '', text)
    text = re.sub('@[^\s]+', '', text)
    text = re.sub('[^a-zA-Za-яА-Я1-9]+', ' ', text)
    text = re.sub(' +', ' ', text)
    text = re.sub('[a-zA-Z]+', '', text)
    return text.strip()
```

Рис. 1. Функция предварительной обработки текста

3. Далее, используя библиотечную функцию *train_test_split*, разбиваем подготовленные данные на обучающую и тестовую выборки.

4. Построение конвейера операций. В задаче машинного обучения есть стандартные процессы, которые необходимо автоматизировать на стадии обработки. Утилита Pipeline поможет решить эту задачу. Конвейеры работают по принципу объединения линейных операций преобразования данных. Цель состоит в том, чтобы ограничить данные, делая их доступными для оценки.

Построение классификатора

Чтобы выполнить машинное обучение на произвольные текстовых документах, нам сначала нужно преобразовать текстовое содержимое в векторное представление. Метод CountVectorizer() позволяет создать словарь объектов и преобразует документы в векторы объектов.

Однако у более длинных документов среднее значение частоты вхождения будет выше, чем у более коротких, даже если они в рамках одной предметной области. Чтобы избежать таких неравенств, необходимо разделить количество вхождений каждого слова в документе на общее количество слов в документе. Такой подход оценивается метрикой TF-IDF [4] и определяется по формуле вида

$$TF - IDF = tf(t,d) * idf(t)\,, \qquad (2)$$

где $tf(t,d)$ — частота слова — определяется как частота вхождения некоторого слова к общему числу слов документа:

$$tf(t,d) = \frac{f_d(t)}{max f_d(w)\{w \in d\}}\,; \qquad (3)$$

$idf(t)$ — обратная частота документа — является инверсией частоты, с которой слово встречается в документе:

$$idf(t,D) = \log\frac{|D|}{|\{d_i \in D \mid t \in d_i\}|} + 1\,, \qquad (4)$$

где |D| — общее число документов;
$|\{d_i \in D \mid t \in d_i\}|$ — число документов из коллекции D, в которых встречается t, при условии, что число вхождений слова t в документ не равно 0.

Теперь, когда данные приведены к нужному виду, мы можем обучить классификатор, чтобы попытаться предсказать тональность документа.

Чтобы упростить работу с трансформацией и подготовкой данных используем объект класса Pipeline, который ведет себя как составной классификатор. В первоначальной прогонке не будем использовать параметризированные конструкты методов (рис. 2).

```
nb = Pipeline([('vect', CountVectorizer()),
               ('tfidf', TfidfTransformer()),
               ('clf', MultinomialNB()),
              ])
nb.fit(x_train, y_train)
```

Рис. 2. Создание конвейера преобразований

Для решения нашей задачи биномиальной классификации текста мы использовали наивный байесовский классификатор [5]. Далее необходимо произвести оценку работы классификатора для предсказания тестовых данных. Оценки точности, отклика и полноты представлены на рисунке 3. Как можем заметить, точность прогнозирования на начальных настройках равна 0,731. Запомним это значение для дальнейшего сравнения.

```
Оценка точности: 0.731
              precision    recall  f1-score   support

    positive       0.71      0.78      0.74     22236
    negative       0.76      0.69      0.72     22534

    accuracy                           0.73     44770
   macro avg       0.73      0.73      0.73     44770
weighted avg       0.73      0.73      0.73     44770
```

Рис. 3. Оценка точности классификатора на тестовых данных

Для настройки классификатора MultinomialNB нам нужно создать сетку с параметрами, по которым будем перемещаться, перебирая их в цикле. Также нужно будет указать метрику, которую мы будем проверять, — в нашем случае это AP (общая точность) [6]. Таким образом, мы будем искать модель, наилучшую с точки зрения метрики. В процессе обучения у нас будет оценена точность в каждой из возможных комбинаций параметров. Это очень ресурсоемкая задача, поэтому будем ее проводить на части данных — выборке в 1 000 строк.

Классификатор будет оптимизирован путем проведения перекрестной проверки, которая выполняется с созданием объекта класса GridSearchCV, принимающего в качестве входных параметров объект классификатора, набор проверяемых параметров, функцию оценивания.

Производительность выбранных гиперпараметров и обученной модели измеряется на специальном тестовом наборе данных, который не использовался на этапе выбора модели.

В результате получим гиперпараметры с максимальной производительностью, среди которых: функция сглаживания, правило игнорирования термов df, использование последовательности из n элементов.

```
Performing grid search...
Лучшая оценка параметров: 0.600
Оценка точности %0.3f
Лучший набор параметров:
        clf__alpha: 1.0
        tfidf__use_idf: False
        vect__max_df: 0.5
        vect__ngram_range: (1, 3)
```

Рис. 4. Рекомендованные значения параметров

Применяя рекомендованные параметры к нашей обучающей модели, можно добиться увеличения производительности, в нашем случае — увеличения точности (рис. 5) прогнозирования классов.

```
Оценка точности: 0.751
              precision    recall  f1-score   support

    positive       0.72      0.82      0.77     22236
    negative       0.80      0.68      0.73     22534

    accuracy                           0.75     44770
   macro avg       0.76      0.75      0.75     44770
weighted avg       0.76      0.75      0.75     44770
```

Рис. 5. Применение рекомендованных параметров

Как видно из данных, приведенных на рисунках 3 и 5, оценка точности классификации текста изменяется от 0,731 до 0,751.

Заключение

Правильно подобранный набор гиперпараметров в процессе построения объекта классификатора позволит увеличить показатели эффективности работы классификатора при прогнозировании значений, что приведет к соответствующему снижению показателей ошибки или рисков при использовании алгоритма классификации в коммерческих целях.

Дальнейшие исследования целесообразно продолжить в направлении повышения эффективности использования Интернет-ресурсов для изучения отношения людей к происходящим в мире событиям, к продуктам, предлагаемым компаниями [7].

Литература

1. Hyperparameter optimization: Explanation of automatized algorithms // Dawid Kopczyk.
URL: http://dkopczyk.quantee.co.uk/hyperparameter-optimization (дата обращения 15.09.2020).

2. Рубцова Ю. В. Построение корпуса текстов для настройки тонового классификатора // Программные продукты и системы. 2015. № 1 (109). С. 72–78.
DOI: 10.15827/0236-235X.109.072-078.

3. Гребнев К. Н. Машинное обучение с помощью библиотеки scikit-learn языка Python // Математический вестник педвузов и университетов Волго-Вятского региона. 2017. № 19. С. 277–281.

4. Оценка семантической близости документов на основе латентно-семантического анализа с автоматическим выбором ранговых значений / С. А. Краснов, А. С. Илатовский, А. Д. Хомоненко, В. Н. Арсеньев // Труды СПИИРАН. 2017. Вып. 5 (54). С. 185–204.
DOI: 10.15622/sp.54.8.

5. Хомоненко А. Д. Нейросетевая аппроксимация характеристик многоканальных немарковских систем массового обслуживания / А. Д. Хомоненко, Е. Л. Яковлев // Труды СПИИРАН. 2015. Вып. 4 (41). С. 81–93.
DOI: 10.15622/sp.41.4.

6. Попков М. И. Автоматическая система классификации текстов для базы знаний предприятия // International Journal of Open Information Technologies. 2014. Т. 2, № 7. С. 11–18.

7. Emotion-Bracelet: A Web Service for Expressing Emotions through an Electronic Interface / A. Martinez, H. Estrada, A. Molina, et al. // Sensors. 2016. Vol. 16, Is. 12. P. 1980. Published online at November 24, 2016.
URL: http://www.mdpi.com/1424-8220/16/12/1980 (дата обращения 15.09.2020).
DOI: 10.3390/s16121980.

Estimation of Hyperparameters in the Analysis of the Tonality of the Russian-Language Text Corpus

PhD student N. E. Kosykh
Emperor Alexander I St. Petersburg State Transport University
St. Petersburg, Russia
nikitosagi@mail.ru

***Abstract*. An approach to the selection of the most productive set of parameters for a text classifier object is proposed. For the calculations, the Google Colaboratory cloud service was used, which executes Python code inside the browser using virtual hardware resources. The working environment includes components for developing a machine learning model: the scikit-learn library, which contains teaching methods and their assessment; library for data analysis and processing — Pandas; regex library — Re. The main attention is paid to the accuracy in the process of forecasting the tonality of the Russian-language corpus of texts. To solve the problem of binomial text classification, a naive Bayesian classifier was used. The application of a programmatic approach to finding the optimal values of hyperparameters on a grid improves the accuracy of the classifier on a limited data set.**

***Keywords*: classification, hyperparameters, accuracy, supervised learning, tonality.**

REFERENCES

1. Hyperparameter optimization: Explanation of automatized algorithms // Dawid Kopczyk.
Available at: http://dkopczyk.quantee.co.uk/hyperparameter-optimization (accessed 15 Sep 2020).

2. Rubtsova Yu. V. Constructing a Corpus for Sentiment Classification Training [Postroenie korpusa tekstov dlya nastroyki tonovogo klassifikatora], *Software & Systems [Programmnye produkty i sistemy]*, 2015, No. 1 (109), Pp. 72–78. DOI: 10.15827/0236-235X.109.072-078.

3. Grebnev K. N. Machine Learning Using the Scikit-Learn Python Library [Mashinnoe obuchenie s pomoshch'yu biblioteki scikit-learn yazyka Python], *Mathematical Bulletin of Pedagogical Universities and Universities of the Volgo-Vyatka Region [Matematicheskiy vestnik pedvuzov i universitetov Volgo-Vyatskogo regiona]*, 2017, No. 19, Pp. 277–281.

4. Krasnov S. A., Ilatovsky A. S., Khomonenko A. D., Arseniev V. N. Assessment of Semantic Similarity of Documents on the Basis of the Latent Semantic Analysis with the Automatic Choice of Rank Values [Otsenka semanticheskoy blizosti dokumentov na osnove latentno-semanticheskogo analiza s avtomaticheskim vyborom rangovykh znacheniy], *SPIIRAS Proceedings [Trudy SPIIRAN]*, 2017, Is. 5 (54), Pp. 185–204. DOI: 10.15622/sp.54.8.

5. Khomonenko A. D., Yakovlev E. L. Neural Network Approximation of Characteristics of Multi-Channel Non-Markovian Queuing Systems [Neyrosetevaya approksimatsiya kharakteristik mnogokanal'nykh nemarkovskikh sistem massovogo obsluzhivaniya], *Proceedings of SPIIRAS [Trudy SPIIRAN]*, 2015, Is. 4 (41), Pp. 81–93. DOI: 10.15622/sp.41.4.

6. Popkov M. I. Text Analytics for Enterprise Knowledge Base [Avtomaticheskaya sistema klassifikatsii tekstov dlya bazy znaniy predpriyatiya], *International Journal of Open Information Technologies*, 2014, Vol. 2, No. 7, Pp. 11–18.

7. Martinez A., Estrada H., Molina A., et al. EmotionBracelet: A Web Service for Expressing Emotions through an Electronic Interface, *Sensors*, 2016, Vol. 16, Is. 12, P. 1980. Published online at November 24, 2016.
Available at: http://www.mdpi.com/1424-8220/16/12/1980 (accessed 15 Sep 2020). DOI: 10.3390/s16121980.

Обоснование выбора метрики для оценки качества передачи потокового видео

аспирант Н. А. Гаврилова
Петербургский государственный университет путей сообщения Императора Александра I
Санкт-Петербург, Россия
wuuzee.nd@gmail.com

Аннотация. **Описываются характеристики стандарта H.264/AVC в части возможности появления ошибок во время передачи видео по сети. В работе проведено имитационное моделирование передачи видео через беспроводную сеть с целью исследования влияния случайной битовой ошибки в канале. Для оценивая качества передаваемого видео исследуется возможность применения метрик PSNR (отношение сигнала к шуму), SSIM (структурная схожесть) и VMAF (мультиметодная оценка на основе слияния известных метрик) и их субъективное сравнение. Показано, что метрика VMAF хорошо коррелирует с воспринимаемым качеством и ее целесообразно использовать в задачах оценки потокового видео.**

Ключевые слова: **передача видео, ошибки H.264, метрики качества видео, эталонные метрики оценки видео.**

Введение

При передаче видео по каналам связи важно гарантировать пользователю необходимый уровень воспринимаемого качества, что обеспечивается выбором сети и параметрами кодирования. Благодаря повышенной компрессии стандарт видео кодирования H.264/AVC (*англ.* Advanced Video Coding — расширенное кодирование видео) позволяет транслировать видео в низкоскоростных сетях без заметного ухудшения качества, что позволяет использовать этот стандарт для видеоприложений в беспроводных сетях. Но при резком уменьшении ширины канала передаваемой информации могут возникать ошибки, которые влияют на полученные данные. В подобной ситуации необходимо уметь оценивать качество передаваемого потокового видео с целью обнаружения таких битовых ошибок в канале.

Потоковое видео (*англ.* Streaming Video) — это технология буферизации и сжатия данных, позволяющая вести трансляцию мультимедийного контента (видео) через Интернет в режиме реального времени. Иными словами, потоковое видео подразумевает преобразование видео- и аудиоконтента в сжатый цифровой формат с его последующим распространением через компьютерные сети. Сжатые данные могут быть доставлены с использованием компьютерных сетей в силу их небольших размеров. Видео и аудио может быть постоянно потоковым, либо доставляться по требованию.

Виды ошибок

Возникающие при передаче по беспроводным сетям битовые ошибки могут по-разному влиять на качество декодированного видео. Их можно разделить по локализации ошибки в потоке [1]:

1. Битовая ошибка в ***различных частях*** битового потока.

Поскольку механизм сжатия использует удаление избыточностей в видеопоследовательности, относительно низкий уровень битовых ошибок может существенно повлиять на качество декодированного видео. Количество битовых ошибок выше допустимого может значительно ухудшить качество.

2. Битовая ошибка в ***заголовке*** видеопоследовательности.

Заголовок видеопоследовательности включает в себя важную информацию, такую как разрешение кадра, число кадров, и таблицу квантизации. Если ошибка исказила один из этих параметров, последовательность нельзя корректно декодировать. При небольшом количестве ошибочных битов вероятность искажения заголовка невелика, поскольку его размеры относительно всего потока небольшие.

3. Битовая ошибка в ***заголовке изображения***.

При ошибке в заголовке декодер может не распознать начало кадра. В худшем случае кадр будет потерян. В остальных случаях, при временном предсказывании, могут возникнуть серьезные ухудшения качества.

4. Битовая ошибка в ***группе кадров*** (GOP).

Ошибка в GOP или его заголовке не является существенной для правильного декодирования видеопоследовательности.

5. Битовая ошибка в ***коэффициентах DCT*** (дискретного косинусного преобразования).

Если искажается часть коэффициентов DCT, это может привести к «неправильному декодированию» кодов переменной длины VLC (*англ.* Variable-Length Coding).

Поскольку кодеки обрабатывают информацию поблочно, то минимальной единицей искажения видеопотока при воздействии одиночной ошибки является блок (4×4 или 16×16 в зависимости от кодирования). Следующей областью распространения ошибки является макроблок и слайс. Таким образом, одиночная ошибка при передаче может вызвать распространение ошибки не только в актуальном макроблоке, но и в слайсе и далее в кадре.

Существуют три возможных источника распространения ошибки [2]:

1. Пространственное предсказывание.

Восстановленный при декодировании макроблок, у которого соседние макроблоки искажены, также будет искажен.

2. Временное предсказывание.

Если происходит искажение кадра, то следующие кадры, использующие искаженный кадр как исходный, также будут искажены.

3. Энтропийное кодирование.

Поскольку используются коды VLC, ошибка в ключевом кадре может влиять на следующие кадры, если его границы определены неправильно. Таким образом, нарушается синхронизация следующих кадров, что влечет за собой неспособность декодера различить ключевые кадры [3].

Использование VLC приводит к рассинхронизации декодированной информации, приводя к тому, что часть информации до следующего кадра становится недекодируемой. В некоторых случаях даже после восстановления синхронизации декодированный сигнал не может быть корректно использован, поскольку потеряна дополнительная информация о способе ее использования, например тип кадра или вектора движения.

На практике можно наблюдать искажения, возникающие в результате воздействия ошибок при передаче и последующем декодировании: блочность изображения (*англ.* tiling); нечеткость, размазанность (*англ.* blurring); ошибки цветопередачи (*англ.* color errors); ошибочные блоки (*англ.* error block); дрожание (*англ.* jerkiness); эффект «комаров» (*англ.* mosquito noise); шум квантования (*англ.* quantization noise); размытость (*англ.* smearing) [4].

Оценка качества видео

При сравнении исходного и искаженного видеопотоков возможно вычислить влияние битовой ошибки на конечное качество видео. Традиционно качество видео оценивается с помощью субъективных и объективных показателей. Субъективная оценка качества всегда опирается на впечатление зрителя и определяется путем экспертной оценки и подсчетом среднего балла MOS (*англ.* Mean Opinion Score). Объективное качество можно оценить различными метриками [5].

В современной литературе описано достаточно большое число объективных метрик, которые можно разделить на три следующих класса [6]:

Эталонные (Full Reference, FR) предполагают наличие исходного видеопотока, который рассматривается как опорный, или эталонный, при сравнении, так как он не зашумлен и имеет идеальное качество.

Неэталонные (No Reference, NR) предполагают, что в процессе получения оценки качества видеопотока опорный или эталонный поток отсутствует. Такие метрики являются самыми сложными в реализации и зачастую ориентированы на конкретный вид искажения.

Псевдоэталонные (Reduced Reference, RR) предполагают, что некоторая часть информации об эталонном видеопотоке присутствует вместе с зашумленным, причем количество этой информации значительно меньше объема информации, требуемого для эталонного видеопотока.

Рассмотрим три эталонные метрики — PSNR (Peak Signal-to-Noise Ratio), SSIM (Structural Similarity Index Measuring), Netflix VMAF (Video Multimethod Assessment Fusion).

PSNR

Пиковое отношение сигнала к шуму наиболее часто используется для измерения уровня искажений при сжатии изображений. Так, для оригинального изображения x и искаженного изображения y PSNR рассчитывается как

$$PSNR = 10\,log_{10}\frac{L^2}{MSE}, \quad (1)$$

где L — динамический диапазон допустимых интенсивностей пикселей изображения;

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(x_i - y_i)^2. \quad (2)$$

SSIM

Индекс структурного сходства является одним из методов измерения схожести между двумя изображениями. SSIM-индекс — это метод полного сопоставления, другими словами, он проводит измерение качества на основе исходного изображения. Так, для оригинального изображения x и искаженного изображения y SSIM рассчитывается как [7]

$$SSIM(x,y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2)(\sigma_x^2 + \sigma_y^2)}, \quad (3)$$

где $\mu(x)$ — среднее значение изображения x;
$\mu(y)$ — среднее значение изображения y;
$\sigma(x)$ и $\sigma(y)$ — среднеквадратичное отклонение для изображения x и для изображения y соответственно; $\sigma(x,y)$ — ковариация;
C_1 и C_2 — поправочные коэффициенты.

VMAF

VMAF — новая метрика, разработанная компанией Netflix [8]. Она предсказывает субъективное качество видео на основе эталонной и искаженной последовательности видео. В VMAF применяются различные метрики и комбинируются между собой с помощью метода опорных векторов. Показатели качества, которые использует VMAF:

1. VIF (Visual Information Fidelity) — индекс визуальной достоверности информации [9]. Построение этого индекса основано на моделировании источника эталонного изображения, искаженного изображения и визуальных искажениях человека. VIF показывает потерю точности информации.

2. DLM (Detail Loss Metric) — метрика потери детализации [10]. Измеряет потерю деталей, которые отвлекают внимание пользователей.

3. MCPD (Mean Co-Located Pixel Difference) — средняя временная разность пикселей

4. AN-SNR (Anti-Noise Signal-to-Noise Ratio) — антишумовое соотношение сигнала к шуму.

Имитация битовых ошибок потокового видео

Беспроводные каналы связи характеризуются случайно распределенными и независимыми ошибками. В связи с этим при имитации беспроводного канала часто применяют модель «аддитивного белого гауссовского шума», или AWGN (*англ.* Adaptive White Gaussian Noise), при которой определенный бит в последовательности искажается с заданной вероятностью. Используемое значение вероятности описывают показателем количества ошибочных битов BER (Bit Error Rate). Различные значения BER поразному влияют на качество потокового видео.

Для исследования влияния битовой ошибки на качество видео проведено имитационное моделирование передачи видео через беспроводную сеть со случайными битовыми ошибками в канале.

Кодирование/декодирование исходной видеопоследовательности и моделирование беспроводной сети со случайными битовыми ошибками в канале производились с помощью программы VCDemo [11].

Для имитации передачи были применены следующие параметры:

- Протокол: GPRS.
- Пропускная способность канала: 1 000 кбит/сек.
- Случайная битовая ошибка с P = 0.05.

В результате моделирования получено видео с ошибками. На рисунке 2 видна область с ошибками отображения.

Рис.1. Кадр из эталонного видео

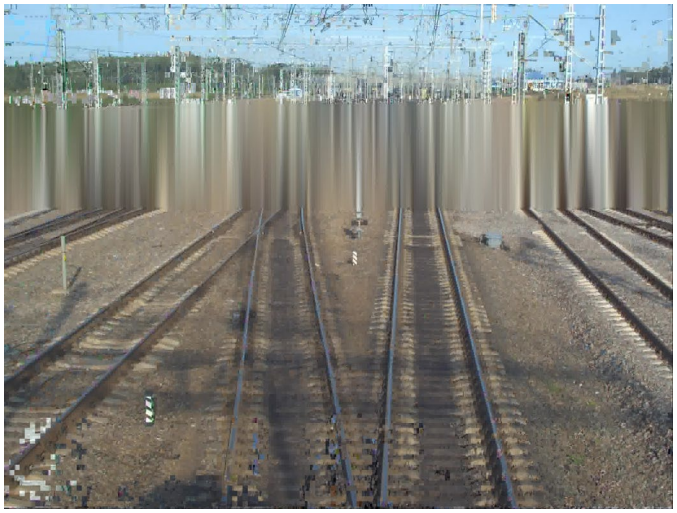

Рис. 2. Кадр из смоделированного видео

Далее для оценки качества искаженного видео были выбраны объективные метрики PSNR, SSIM, VMAF. В ходе эксперимента, были получены следующие результаты оценки качества искаженного видео (рис. 3).

Полученные результаты оценки видео показывают, что существующие метрики SSIM и VMAF достаточно хорошо оценивают качество кадра. Что касается PSNR, то он плохо коррелирует с воспринимаемым качеством. Также стоит отметить, что, субъективно, VMAF показал лучший результат, так как в отличии от PSNR и SSIM его оценочные результаты лучше коррелируют с субъективной оценкой.

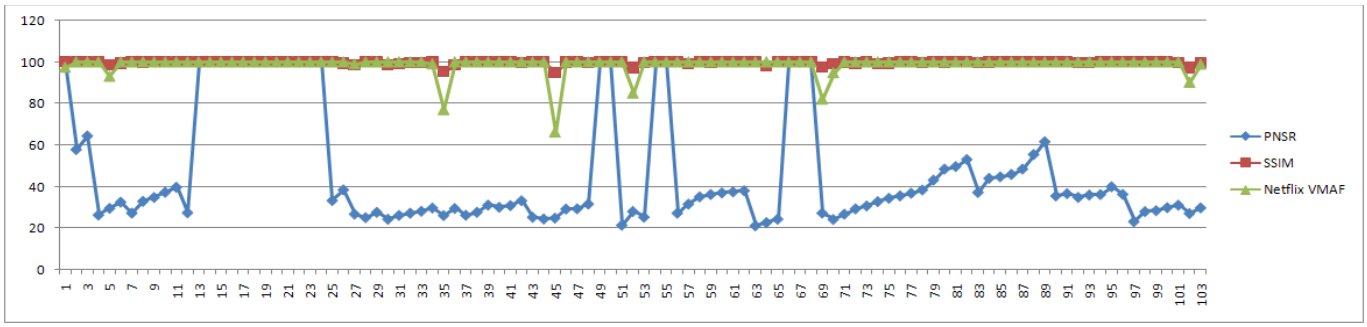

Рис. 3. Диаграмма полученных значений метрик PSNR, SSIM и VMAF

Эмпирические значения индекса VMAF от приемлемого качества к плохому представлены в таблице 1.

Таблица 1

Соответствие показателей качества VMAF

VMAF	**Качество по шкале ITU**	**Ухудшение изображения**
91–100	5 — прекрасное	незаметно
71–90	4 — хорошее	заметно, но не раздражает
51–70	3 — удовлетворительное	слегка раздражает
41–50	2 — плохое	раздражает
0–40	1 — очень плохое	сильно раздражает

Заключение

При передаче видео по сети из-за ненадежных каналов связи часто возникают ошибки, которые сильно влияют на получаемый результат. Важно правильно и достоверно оценивать такие ошибки, чтобы в дальнейшем использовать выбранные методы для оптимизации работы кодеков, сбора статистики, а также анализа воспринимаемой информации. Полученные в результате опыта данные показывают, что метрика VMAF хорошо коррелирует с воспринимаемым качеством и ее можно использовать в дальнейших задачах оценки качества потокового видео.

Литература

1. Romer M. MPEG-4 Video Quality Analysis // Video Communications Project, 2004.

2. Rodriguez E. R. Robust Error Detection Methods for H.264/AVC Videos / E. R. Rodriguez; Universitat Politecnica

de Catalunya; EPSC Technical University of Vienna; Institute of Communications and Radio-Frequency Engineering. — Vienna, 2008. — 65 p.

3. Kolkeri V. S. Error Concealment Techniques in H.264/AVC for Video Transmission Over Wireless Network / V. S. Kolkeri; University of Texas at Arlington. — 2009.

4. Иванов Ю. А. Оценка качества потокового видеостандарта H.264/AVC при передаче в нестабильных каналах связи широкополосных сетей беспроводного доступа 4G // Вестник Чувашского университета. 2010. № 3. С. 268–278.

5. Сидоров Д. В. Оценка качества изображений с использованием вейвлетов / Д. В. Сидоров, А. Н. Осокин, Н. Г. Марков // Известия Томского политехнического университета. Инжиниринг георесурсов. 2009. Т. 315, № 5. С. 104–107.

6. Wang Z. Modern Image Quality Assessment: Synthesis Lectures on Image, Video, and Multimedia Processing / Z. Wang, A. C. Bovik. — USA: Morgan & Claypool, 2006. — 156 p. DOI: 10.2200/S00010ED1V01Y200508IVM003.

7. Image quality assessment: From error visibility to structural similarity / Z. Wang, A. C. Bovik, H. R. Sheikh, E. P. Simoncelli // IEEE Transactions on Image Processing. 2004. Vol. 13, Is. 4. Pp. 600–612. DOI: 10.1109/TIP.2003.819861.

8. Toward A Practical Perceptual Video Quality Metric / Z. Li, A. Aaron, I. Katsavounidis, et al. — 06.06.2016 // The Netflix TechBlog. URL: http://netflixtechblog.com/toward-a-practical-perceptual-video-quality-metric-653f208b9652 (дата обращения 18.08.2020).

9. Sheikh H. R. Image Information and Visual Quality. / H. R. Sheikh, A. C. Bovik // IEEE Transactions on Image Processing. 2006. Vol. 15, Is. 2. Pp. 430–444.
DOI: 10.1109/TIP.2005.859378.

10. Image Quality Assessment by Separately Evaluating Detail Losses and Additive Impairments / S. Li, F. Zhang, K. N. Ngan, L. Ma // IEEE Transactions on Multimedia. 2011. Vol. 13, Is. 5. Pp. 935–949.
DOI: 10.1109/TMM.2011.2152382.

11. VcDemo // Delft University of Technology. URL: http://homepage.tudelft.nl/c7c8y/VcDemo.html (дата обращения 18.08.2020).

The Rationale of Choosing a Quality Assessment Metric of Streaming Video

PhD student N. A. Gavrilova
Emperor Alexander I Petersburg State Transport University
Saint Petersburg, Russia
wuuzee.nd@gmail.com

Abstract. **Describes the characteristics of the H. 264/AVC standard in terms of the possibility of errors during video transmission over the network. In this paper, a simulation of video transmission over a wireless network performed in order to study the effect of a random bit error in the channel. To assess the quality of transmitted video, we investigate the possibility of using the PSNR (signal-to-noise ratio), SSIM (structural similarity), and VMAF (video multi method estimation based on fusion known metrics) metrics and their subjective comparison. It shown that the VMAF metric correlates well with the perceived quality and it is appropriate to use it in problems of evaluating streaming video.**

Keywords: **video transmission, H.264 errors, video quality metrics, full-reference video metrics.**

REFERENCES

1. Romer M. MPEG-4 Video Quality Analysis // Video Communications Project, 2004.

2. Rodriguez E. R. Robust Error Detection Methods for H.264/AVC Videos. Universitat Politecnica de Catalunya, EPSC Technical University of Vienna, Institute of Communications and Radio-Frequency Engineering, Vienna, 2008, 65 p.

3. Kolkeri V. S. Error Concealment Techniques in H.264/AVC for Video Transmission Over Wireless Network. University of Texas at Arlington, 2009.

4. Ivanov Yu. A. Quality Estimation of H.264/AVC Video Stream in Case of Transmission Through Unstable Data Channels of Broadband Wireless Networks 4G [Otsenka kachestva potokovogo videostandarta H.264/AVC pri peredache v nestabil'nykh kanalakh svyazi shirokopolosnykh setey besprovodnogo dostupa 4G], *Bulletin of the Chuvash University [Vestnik Chuvashskogo universiteta]*, 2010, No. 3, Pp. 268–278.

5. Sidorov D. V., Osokin A. N., Markov N. G. Image Quality Estimation Using Wavelets [Otsenka kachestva izobrazheniy s ispol'zovaniem veyvletov], *Bulletin of the Tomsk Polytechnic University. Geo Assets Engineering [Izvestiya Tomskogo politekhnicheskogo universiteta. Inzhiniring georesursov]*, 2009. Vol. 315, No. 5, Pp. 104–107.

6. Wang Z., Bovik A. C. Modern Image Quality Assessment: Synthesis Lectures on Image, Video, and Multimedia Processing. USA, Morgan & Claypool, 2006, 156 p. DOI: 10.2200/S00010ED1V01Y200508IVM003.

7. Wang Z., Bovik A. C., Sheikh H. R., Simoncelli E. P. Image quality assessment: From error visibility to structural similarity, *IEEE Transactions on Image Processing*, 2004, Vol. 13, Is. 4, Pp. 600–612. DOI: 10.1109/TIP.2003.819861.

8. Li Z., Aaron A., Katsavounidis I., et al. Toward A Practical Perceptual Video Quality Metric, *The Netflix TechBlog* Published online at June 06, 2016. Available at: http://netflixtechblog.com/toward-a-practical-perceptual-video-quality-metric-653f208b9652 (accessed 18 Aug 2020).

9. Sheikh H. R., Bovik A. C. Image Information and Visual Quality, *IEEE Transactions on Image Processing*, 2006, Vol. 15, Is. 2, Pp. 430–444. DOI: 10.1109/TIP.2005.859378.

10. Li S., Zhang F., Ngan K. N., Ma L. Image Quality Assessment by Separately Evaluating Detail Losses and Additive Impairments, *IEEE Transactions on Multimedia*, 2011, Vol. 13, Is. 5, Pp. 935–949. DOI: 10.1109/TMM.2011.2152382.

11. VcDemo // Delft University of Technology. Available at: http://homepage.tudelft.nl/c7c8y/VcDemo.html (accessed 18 Aug 2020).

Оркестровка в области IT-технологий

аспирант Н. К. Уваров
Петербургский государственный университет
путей сообщения Императора Александра I
Санкт-Петербург, Россия
nick553@mail.ru

Аннотация. **Рассматривается смысл термина «оркестровка» в области IT-технологий. Характеризуются основные области оркестровки: облаков, контейнеров, конфигураций серверов. Кратко описывается язык BPEL, используемый для определения бизнес-процессов с применением веб-сервисов. Приведен пример BPEL-кода, оркеструющего процесс запроса на покупку билетов. Дается краткая характеристика программных средств, применяемых для оркестровки IT-технологий: Ansible, Amazon Elastic Container Service и IBM Cloud Orchestrator. Указываются достоинства применения оркестровки в IT-технологиях.**

Ключевые слова: **оркестровка IT-технологий, автоматизация, BPEL, управление серверами.**

Введение

Термин «оркестровка» хорошо знаком любителям музыки, где он обозначает процесс написания мелодии для оркестра или переложение существующего произведения на оркестровый инструментал. Это достаточно сложный процесс, выполняемый оркестратором, который назначает инструменты, пишет партитуру, тем самым создавая «сценарий» произведения, в котором синхронизируются звуки всех инструментов.

В сфере IT часто практикуется заимствование терминов из других областей для описания схожих по сути процессов. Термин «оркестровка» не стал исключением. В предлагаемой статье рассмотрено значение этого термина в области компьютерных технологий и краткая характеристика соответствующих программных средств.

Определение оркестровки

Оркестровка — автоматическое размещение, координация и управление сложными компьютерными системами и службами [1].

Чаще всего этот термин используется в контексте сервис-ориентированной архитектуры программных комплексов, где используется Web Service Orchestration. Для оркестровки сервисов используется язык Business Process Execution Language (WS-BPEL). Цель оркестровки сервисов заключается в создании исполняемого бизнес-процесса. Она описывает то, как сервисы должны взаимодействовать между собой, используя для этого обмен сообщениями, включая бизнес-логику и последовательность действий [2]. Основное отличие оркестровки от смежной технологии — хореографии — в том, что первый процесс подчиняется только одному из участников, а второй — всему коллективу.

Оркестровка и автоматизация

Часто термин «автоматизация» используют, когда говорят об автоматическом выполнении процессов. Это не совсем правильно. Автоматизация предполагает выполнение одной единственной задачи. Все объекты автоматизации находятся в одном домене.

Оркестровка предполагает управление объектами намного большего масштаба, такими как взаимодействие процессов, сервисов, приложений и т. д. Она включает в себя множество автоматизированных задач, представляющих в совокупности единый повторяющийся процесс.

Автоматизация некоторых задач — первый шаг для ввода технологии оркестровки. Автоматизация отвечает, в основном, за технические задачи, тогда как оркестровка управляет целыми рабочими нагрузками и потоками данных [3].

Области оркестровки

В области IT-технологий можно выделить оркестровки нескольких типов:

– оркестровка облака может быть использована для развертывания или обеспечения серверов, управления количеством памяти, создания виртуальных машин, управления сетью и т. д.;

– оркестровка контейнеров используется для управления контейнерной виртуализацией, включая в себя автоматизацию операций контейнерных ОС, а также развертывание и расширение приложений в них;

– оркестровка конфигураций серверов, управления серверами, размещения приложений.

Business Process Execution Language

Язык BPEL используют для определения того, как будет проходить бизнес-процесс с использованием веб-сервисов. Он нацелен на моделирование поведения процессов через язык для спецификации как исполняемых, так и абстрактных бизнес-процессов [4]. Это расширяет возможности модели взаимодействия веб-сервисов и позволяет ей поддерживать бизнес-транзакции. Сообщения на BPEL, в основном, используются для пробуждения удаленных сервисов, оркестровки процесса выполнения и управления событиями и исключениями [5].

BPEL стандартизирован компанией OASIS в 2004 году.

Средства работы с сообщениями в BPEL зависят от WSDL (Web Services Description Language — языка описания веб-сервисов), который описывает входящие и исходящие сообщения.

Также BPEL поддерживает:

– механизм корреляции сообщений, основанный на собственности;

– переменные XML и WSDL;

– расширяемый механизм плагинов для возможности написания выражений на нескольких языках;

– операторы структурного программирования: if-then-elseif-else, while, sequence и flow;

– систему обозначений области видимости, позволяющую реализовать механизм инкапсуляции;

– контроль одновременного обращения к переменным.

Далее представлен листинг BPEL-кода, оркеструющего процесс получения запроса на покупку билетов от пользователя, а затем отправления запроса в авиакомпанию и получения билетов [6].

```
<PROCESS NAME=«TICKETORDER»>»
  <PARTNERS>
   <PARTNER NAME=«CUSTOMER»
   SERVICELINKTYPE=«AGENTLINK»
   MYROLE=«AGENTSERVICE»/>
   <PARTNER NAME=«AIRLINE»
   SERVICELINKTYPE=«BUYERLINK»
   MYROLE=«TICKETREQUESTER»
   PARTNERROLE=«TICKETSERVICE»/>
 </PARTNERS>
 <CONTAINERS>
   <CONTAINER NAME=« ITINERARY»
   MESSAGETYPE=«ITINERARYMESSAGE»/>
   <CONTAINER NAME=«TICKETS»
   MESSAGETYPE=«TICKETSMESSAGE»/>
  </CONTAINERS>
  <FLOW>
    <LINKS>
      <LINK NAME=«ORDER-TO-AIRLINE»/>
      <LINK NAME=«AIRLINE-TO-AGENT»/>
    </LINKS>
   <RECEIVE PARTNER=«CUSTOMER»
      PORTTYPE=«ITINERARYPT»
      OPERATION=«SENDITINERARY»
      CONTAINER=«ITINCRARY»
      <SOURCE MNKNAME»ORDER-TO-AIRLINE»/>
   </RECEIVE>
   <INVOKE PARTNER=«AIRLINE»
      PORTTYPE=«TICKETORDERPT»
      OPERATION=«REQUESTTICKETS»
      INPUTCONTAINER=«ITINERARY»
      < TARGET
      LINKNAME»ORDER-TO-AIRLINE»/>
       <SOURCE
      LINKNAME»AIRLINE-TO-AGENT»/>
   </INVOKE>
   <RECEIVE PARTNER=«AIRLINE»
      PORTTYPE=« ITINERARY PT»
      OPERATION=«SENDTICKETS»
      CONTAINER=«TICKETS»
      < TARGET
      LINKNAME»AIRLINE-TO-AGENT»/>
    </RECEIVE>
  </FLOW>
</PROCESS>
```

ПРОГРАММНЫЕ РЕШЕНИЯ ДЛЯ ОРКЕСТРОВКИ

Существует множество решений для оркестровки от различных компаний. В этой статье описаны некоторые из них, такие как Ansible, Amazon Elastic Container Service и IBM Cloud Orchestrator.

Ansible — это программное обеспечение с открытым исходным кодом для управления облаками, конфигурациями, развертыванием приложений, внутрисервисной оркестровкой. Написан на языке программирования Python, с использованием декларативного языка разметки для описания конфигураций. Работает в основном на Unix-системах, но также имеет возможность конфигурировать и Windows.

Работа Ansible заключается в соединении пользовательских узлов и отправлении на них небольших программ, которые называются модулями Ansible. Эти модули являются моделями желаемого состояния системы. Затем они запускаются и удаляются по завершении.

Ansible использует протокол SSH, что увеличивает безопасность выполнения процессов. Все машины записываются в один ini-файл, с помощью которого легко расширить или сузить список используемого оборудования [7].

Для оркестровки используются специальные YAML-файлы под названием Playbooks. В них для каждого хоста определяется конкретный набор ролей, каждая из которых обозначает вызов задания Ansible.

Для более наглядного контроля оркестровки существует Ansible Tower — программное обеспечение с понятным графическим UI и работающее через RESTAPI. Окно работы с Ansible Tower приведено на рисунке 1 [8].

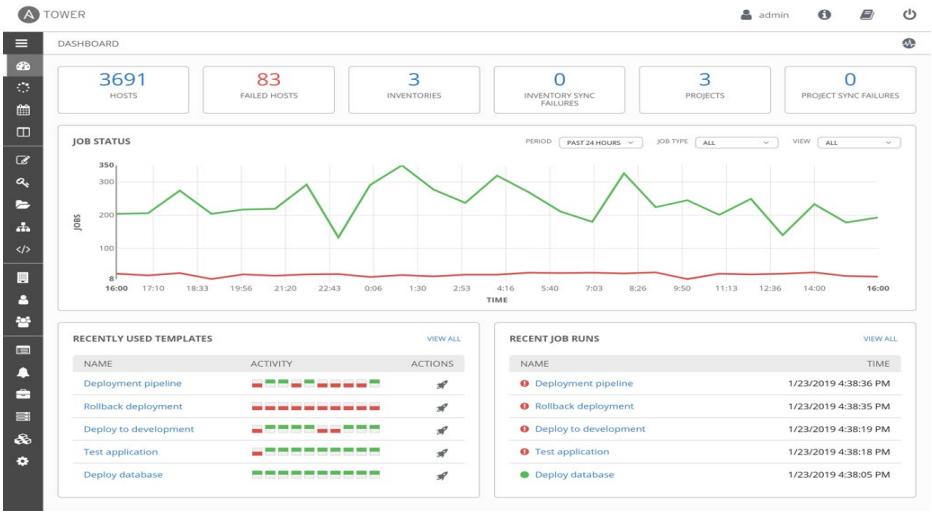

Рис. 1. UI Ansible Tower

Amazon Elastic Container Service (ECS) — это легкорасширяемый и быстрый сервис для управления Docker-контейнерами, осуществляющий их запуск, остановку и управление на кластере. Для взаимодействия с контейнерами используются простые обращения к API.

ECS может управлять расположением контейнеров на кластере в зависимости от требуемых ресурсов, изоляционных политик и доступности.

После установки кластера необходимо обозначить задачи и сервисы, которые определяют, какие образы контейнеров Docker будут запускаться на кластере. Образы контейнеров хранятся в контейнерных регистрах, которые могут находится как внутри, так и снаружи построенной инфраструктуры.

Архитектура ECS состоит из следующих элементов:

1. Контейнеры и образы. Для развертывания приложений в ECS требуется помещать их в контейнеры. Контейнер содержит все необходимое для запуска приложения. В основном они создаются из шаблонов, называемых образами.

2. Определения задач. Существуют для подготовки перед запуском приложений. Определение задачи — это текстовый файл в формате JSON, в котором описаны от одного до десяти контейнеров, в совокупности представляющие приложение. В них определяются параметры приложения: какие контейнеры использовать, какой тип запуска, какие порты должны быть открытыми и т. д.

3. Задачи и планирование. Задача — это конкретизация файла определения задач для конкретного кластера. Каждая задача обладает своей областью изоляции, внутри которой запрещен доступ к ее ресурсам для других задач. Планировщик отвечает за размещение задач на кластере.

4. Кластеры. Логическая группа ресурсов.

5. Агент контейнеров. Запускается на каждом ресурсе инфраструктуры, собирает информацию о запущенных на ресурсе задачах, а также управляет их запуском в зависимости от поступающих от ECS сигналов [9].

IBM Cloud Orchestrator — это программное обеспечение (ПО) для управления облаком IT-служб, которое позволяет управлять общедоступными, частными и гибридными облаками посредством простого в использовании интерфейса [10].

В Cloud Orchestrator интегрированы возможности многих других продуктов IBM.

Данное ПО предоставляет «бесшовную» интеграцию как публичных, так и частных облачных сред. Оно полностью автоматизирует доставку сервисов в частное облако и дает возможность эксплуатации этих сервисов на ресурсах, расположенных в публичных облаках.

Главные компоненты IBM Cloud Orchestrator — это движок процессов и связанный с ним пользовательский интерфейс, который используется для их создания. Для этого система использует возможности IBM Business Process Manage, а также некоторых других продуктов для мониторинга, замеров и администрирования.

Архитектура IBM Cloud Orchestrator состоит из следующих элементов:

1. Infrastructure-as-a-Service. Этот компонент управляет доступом к вычислительным, сетевым и ресурсам хранения данных.

2. Software Stacks. Это не столько отдельный компонент, сколько концепт, в котором при развертывании виртуальных машин также возможно установить различные пакеты приложений при первом запуске этих систем.

3. Workflow Orchestration. Этот компонент снабжен графическим редактором, в котором пользователю дана возможность изменять или расширять процедуры, запускающиеся при инициации запроса пользователя.

4. IBM Cloud Orchestrator Catalog. Это веб-сайт, в котором находятся различные профили автоматизации, которые возможно загрузить и использовать в работе.

5. Public Cloud Gateway. Ответственен за интеграцию с публичными облачными средами.

6. Development tools. Этот компонент обеспечивает возможность включения в проект инструментов разработчика из IBM Rational Team Concert и некоторых плагинов для автоматизации. Сами эти инструменты в состав IBM Cloud Orchestrator не входят.

Заключение

В современных корпоративных системах оркестровка является важным процессом, который позволяет автоматизировать большое количество задач. Это намного упрощает и ускоряет работу серверной части, что положительно сказывается на работе бизнеса.

При грамотном использовании существующего программного обеспечения для оркестровки, оно быстро окупит себя и станет неотъемлемой частью IT-инфраструктуры компании.

Литература

1. Erl T. Service-Oriented Architecture: Concepts, Technology, and Design. — Prentice Hall PTR, 2005. — 792 p.

2. Черняк Л. Сервисы и сложные системы. — 18.01.2008 // Открытые системы. СУБД. 2007. Вып. 10. URL: http://www.osp.ru/os/2007/10/4705804 (дата обращения 19.08.2020).

3. IT Automation vs IT Orchestration: How To Benefit from Both // BMC Blogs. URL: http://www.bmc.com/blogs/it-orchestration-vs-automation-whats-the-difference (дата обращения 19.08.2020).

4. What is BPEL (Business Process Execution Language)? // SearchAppArchitecture.
URL: http://searchapparchitecture.techtarget.com/definition/BPEL-Business-Process-Execution-Language (дата обращения 19.08.2020).

5. Business Process Execution Language for Web Services Specification, version 1.1 dated May 5, 2003 / T. Andrews, F. Curbera, H. Dholakia, et al.
URL: http://download.boulder.ibm.com/ibmdl/pub/software/dw/specs/ws-bpel/ws-bpel.pdf (дата обращения 19.08.2020).

6. Колесов А. Автоматизация бизнес-процессов с помощью BPEL (Business Process Execution Language). — 09.01.2008 // ECM-Journal.
URL: http://ecm-journal.ru/docs/Avtomatizacija-biznes-processov-s-pomoshhju-BPEL-Business-Process-Execution-Language.aspx (дата обращения 19.08.2020).

7. How Ansible Works // Red Hat Ansible.
URL: http://www.ansible.com/overview/how-ansible-works (дата обращения 19.08.2020).

8. Red Hat Ansible Tower // Red Hat Ansible.
URL: http://www.ansible.com/products/tower (дата обращения 19.08.2020).

9. What is Amazon Elastic Container Service? // Amazon Elastic Container Service Developer Guide / Amazon Web Services.
URL: http://docs.aws.amazon.com/AmazonECS/latest/developerguide/Welcome.html (дата обращения 19.08.2020).

10. IBM Cloud Orchestrator. Обзор // IBM — Российская Федерация. URL: http://www.ibm.com/ru-ru/marketplace/deployment-automation (дата обращения 19.08.2020).

Orchestration in the Field of IT Technologies

PhD student N. K. Uvarov
Emperor Alexander I St. Petersburg
State Transport University
St. Petersburg, Russia
nick553@mail.ru

Abstract. **The meaning of the term "orchestration" in the field of IT technologies is considered. It describes the main areas of orchestration: clouds, containers, and server configurations. The BPEL language used for defining business processes using web services is briefly described. An example of a BPEL code that orchestrates the ticket purchase request process is provided. A brief description of the software tools used for orchestration of IT technologies is given: Ansible, Amazon Elastic Container Service, and IBM Cloud Orchestrator. The advantages of using orchestration in IT technologies are indicated.**

Keywords: **IT-technology orchestration, automation, BPEL, server management.**

REFERENCES

1. Erl T. Service-Oriented Architecture: Concepts, Technology & Design, Prentice Hall, 2005, 792 p.

2. Chernyak L. Services and complex systems [Servisy i slozhnye sistemy], *Open Systems. DBMS [Otkrytye sistemy. SUBD]*, 2007, Is. 10. Published at January 18, 2008.
Available at: http://www.osp.ru/os/2007/10/4705804 (accessed 19 Aug 2020).

3. IT Automation vs IT Orchestration: How To Benefit from Both, *BMC Blogs*.
Available at: http://www.bmc.com/blogs/it-orchestration-vs-automation-whats-the-difference (acessed 19 Aug 2020).

4. What is BPEL (Business Process Execution Language)? *SearchAppArchitecture*.
Available at: http://searchapparchitecture.techtarget.com/definition/BPEL-Business-Process-Execution-Language (accessed 19 Aug 2020).

5. Andrews T., Curbera F., Dholakia H., et al. Business Process Execution Language for Web Services Specification, version 1.1 dated May 5, 2003.
Available at: http://download.boulder.ibm.com/ibmdl/pub/software/dw/specs/ws-bpel/ws-bpel.pdf (accessed 19 Aug 2020).

6. Kolesov A. Business-Processes Automatization with BPEL (Business Process Execution Language) [Avtomatizatsiya biznes-protsessov s pomoshch'yu BPEL (Business Process Execution Language)], *ECM-Journal*. Published at January 09, 2008.
Available at: http://ecm-journal.ru/docs/Avtomatizacija-biznes-processov-s-pomoshhju-BPEL-Business-Process-Execution-Language.aspx (accessed 19 Aug 2020).

7. How Ansible Works, *Red Hat Ansible*.
Available at: http://www.ansible.com/overview/how-ansible-works (accessed 19 Aug 2020).

8. Red Hat Ansible Tower, *Red Hat Ansible*.
Available at: http://www.ansible.com/products/tower (accessed 19 Aug 2020).

9. What is Amazon Elastic Container Service? *Amazon Elastic Container Service Developer Guide*, *Amazon Web Services.*
Available at: http://docs.aws.amazon.com/AmazonECS/latest/developerguide/Welcome.html (accessed 19.08.2020).

10. IBM Cloud Orchestrator. Overview. [IBM Cloud Orchestrator. Obzor], *IBM — Russian Federation [IBM — Rossiyskaya Federatsiya]*. Available at: http://www.ibm.com/ru-ru/marketplace/deployment-automation (accessed 19 Aug 2020).