

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

9(217)
2014

ТЕОРЕТИЧЕСКИЙ И ПРИКЛАДНОЙ НАУЧНО-ТЕХНИЧЕСКИЙ ЖУРНАЛ

Издается с ноября 1995 г.

УЧРЕДИТЕЛЬ

Издательство "Новые технологии"

СОДЕРЖАНИЕ

МОДЕЛИРОВАНИЕ И ОПТИМИЗАЦИЯ

- Иванников А. Д., Стемповский А. Л. Формализация задачи отладки проектов цифровых систем. 3
- Дудаков Н. С., Шумилов Ю. Ю. Гибридная система управления хранением и оптимизации данных в АСУ 10
- Грушин В. А., Архипова Я. А. Динамическая модель краткосрочного прогнозирования социально-экономических процессов 15

СИСТЕМЫ АВТОМАТИЗИРОВАННОГО ПРОЕКТИРОВАНИЯ

- Струченков В. И. Алгоритмы нелинейного программирования для САПР железнодорожных трасс. 20
- Салеев Д. В., Преображенский А. П. Анализ особенностей САПР для ПЛИС 28

БЕЗОПАСНОСТЬ ИНФОРМАЦИИ

- Щеглов К. А., Щеглов А. Ю. Защита от атак на уязвимости приложений . . 34

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ В ОБРАЗОВАНИИ

- Пастушак Т. Н., Соколов С. С., Ковальногова Н. М., Маньковская А. В. Приоритетные направления развития электронной информационно-образовательной среды отраслевого университетского комплекса 40
- Смагин А. А., Липатова С. В., Курилова О. Л. Метод численной оценки профессиональных и общекультурных компетенций студента и выпускника вуза 46

СИСТЕМЫ СВЯЗИ И ТЕЛЕКОММУНИКАЦИИ

- Тарасов В. Н., Бахарева Н. Ф., Горелов Г. А., Малахов С. В. Анализ входящего трафика на уровне трех моментов распределений временных интервалов. 55

СОЦИАЛЬНО-ЭКОНОМИЧЕСКИЕ СИСТЕМЫ

- Сербулов Ю. С., Глухов Д. А., Малышев В. А. Модель подсистемы нейтрализации внешних воздействий на производственно-экономическую систему в условиях конкурентной борьбы. 60

Журнал в журнале НЕЙРОСЕТЕВЫЕ ТЕХНОЛОГИИ

- Гриняк В. М., Головченко Б. С., Девятисильный А. С. Нейронечеткая обучаемая система распознавания опасного сближения судов 68
- Чеботарев В. В. Логическая нейронная сеть в основе контроля данных о перевозке грузов на железнодорожном транспорте 74

Главный редактор:

СТЕМПОВСКИЙ А. Л.,
акад. РАН, д. т. н., проф.

Зам. главного редактора:

ИВАННИКОВ А. Д., д. т. н., проф.
ФИЛИМОНОВ Н. Б., д. т. н., с.н.с.

Редакционный совет:

БЫЧКОВ И. В., акад. РАН, д. т. н.
ЖУРАВЛЕВ Ю. И.,
акад. РАН, д. ф.-м. н., проф.
КУЛЕШОВ А. П.,
акад. РАН, д. т. н., проф.
ПОПКОВ Ю. С.,
чл.-корр. РАН, д. т. н., проф.
РУСАКОВ С. Г.,
чл.-корр. РАН, д. т. н., проф.
СОЙФЕР В. А.,
чл.-корр. РАН, д. т. н., проф.
СОКОЛОВ И. А., акад.
РАН, д. т. н., проф.
СУЕТИН Н. В., д. ф.-м. н., проф.
ЧАПЛЫГИН Ю. А.,
чл.-корр. РАН, д. т. н., проф.
ШАХНОВ В. А.,
чл.-корр. РАН, д. т. н., проф.
ШОКИН Ю. И.,
акад. РАН, д. т. н., проф.
ЮСУПОВ Р. М.,
чл.-корр. РАН, д. т. н., проф.

Редакционная коллегия:

АВДОШИН С. М., к. т. н., доц.
АНТОНОВ Б. И.
БАРСКИЙ А. Б., д. т. н., проф.
ВАСЕНИН В. А., д. ф.-м. н., проф.
ГАЛУШКИН А. И., д. т. н., проф.
ДИМИТРИЕНКО Ю. И., д. ф.-м. н., проф.
ДОМРАЧЕВ В. Г., д. т. н., проф.
ЗАГИДУЛЛИН Р. Ш., к. т. н., доц.
ЗАРУБИН В. С., д. т. н., проф.
ИСАЕНКО Р. О., к. т. н., с.н.с.
КАРПЕНКО А. П., д. ф.-м. н., проф.
КОЛИН К. К., д. т. н., проф.
КУЛАГИН В. П., д. т. н., проф.
КУРЕЙЧИК В. М., д. т. н., проф.
КУХАРЕНКО Б. Г., к. ф.-м. н., доц.
ЛЬВОВИЧ Я. Е., д. т. н., проф.
МИХАЙЛОВ Б. М., д. т. н., проф.
НЕЧАЕВ В. В., к. т. н., проф.
РЯБОВ Г. Г., чл.-корр. РАН, д. т. н., проф.
СОКОЛОВ Б. В., д. т. н., проф.
УСКОВ В. Л., к. т. н. (США)
ФОМИЧЕВ В. А., д. т. н., проф.
ЧЕРМОШЕНЦЕВ С. Ф., д. т. н., проф.
ШИЛОВ В. В., к. т. н., доц.

Редакция:

БЕЗМЕНОВА М. Ю.
ГРИГОРИН-РЯБОВА Е. В.
ЛЫСЕНКО А. В.
ЧУГУНОВА А. В.

Информация о журнале доступна по сети Internet по адресу <http://novtex.ru/IT>.

Журнал включен в систему Российского индекса научного цитирования.

Журнал входит в Перечень научных журналов, в которых по рекомендации ВАК РФ должны быть опубликованы научные результаты диссертаций на соискание ученой степени доктора и кандидата наук.

CONTENTS

MODELING AND OPTIMIZATION

- Ivannikov A. D., Stempkovsky A. L.** Formal Mathematical Representation for the Task of Digital System Projects Debugging 3
- Dudakov N. S., Shumilov Y. Y.** The Hybrid Data Management and Optimization System for the ACS 10
- Grushin V. A., Arkhipova Y. A.** Short-Term Prediction Dynamic Model of Socio-Economic Processes. 15

CAD-SYSTEMS

- Struchenkov V. I.** Nonlinear Programming Algorithms for CAD of Railways Routes . . 20
- Saleev D. V., Preobrazhenskiy A. P.** An Analysis of a Features of CAD for FGPA. . . 28

CRYPTOSAFETY INFORMATION

- Shcheglov K. A., Shcheglov A. Yu.** Securing Against Applications Threats. 34

INFORMATION TECHNOLOGIES IN EDUCATION

- Pastuschak T. N., Sokolov S. S., Kovalnogova N. M., Mankovskaya A. V.** Priority Areas for the Development of Electronic Informational and Educational Environment of the University Complex 40
- Smagin A. A., Lipatova S. V., Kurilova O. L.** Method of Numerical Evaluation of Professional and Common Cultural Competences in Undergraduates and Graduates . . 46

CONNECTIONS AND TELECOMMUNICATIONS SYSTEMS

- Tarasov V. N., Bahareva N. F., Gorelov G. A., Malakhov S. V.** Analyzing the Incoming Traffic at the Three Moments Distribution of Time Intervals. 55

SOCIAL AND ECONOMIC SYSTEMS

- Serbulov Y. S., Gluhov D. A., Malyshev V. A.** Model Partition Neutralize External Effects on Industrial and Economic Systems in a Competitive Environment 60

Journal-in-journal NEUROTECHNOLOGIES

- Grinyak V. M., Golovchenko B. S., Devyatisilny A. S.** Neuro-Fuzzy Decision-Making System for Ships Collision Avoidance 68
- Chebotaev V. V.** Logical Neural Network in the Basis of Monitoring Data on the Carriage of Goods by Rail. 74

Editor-in-Chief:

Stempkovsky A. L., Member of RAS,
Dr. Sci. (Tech.), Prof.

Deputy Editor-in-Chief:

Ivannikov A. D., Dr. Sci. (Tech.), Prof.
Filimonov N. B., Dr. Sci. (Tech.), Prof.

Chairman:

Bychkov I. V., Member of RAS,
Dr. Sci. (Tech.), Prof.
Zhuravljov Yu. I., Member of RAS,
Dr. Sci. (Phys.-Math.), Prof.
Kuleshov A. P., Member of RAS,
Dr. Sci. (Tech.), Prof.
Popkov Yu. S., Corresp. Member of RAS,
Dr. Sci. (Tech.), Prof.
Rusakov S. G., Corresp. Member of RAS,
Dr. Sci. (Tech.), Prof.
Soifer V. A., Corresp. Member of RAS,
Dr. Sci. (Tech.), Prof.
Sokolov I. A., Member of RAS,
Dr. Sci. (Phys.-Math.), Prof.
Suetin N. V.,
Dr. Sci. (Phys.-Math.), Prof.
Chaplygin Yu. A., Corresp. Member of RAS,
Dr. Sci. (Tech.), Prof.
Shakhnov V. A., Corresp. Member of RAS,
Dr. Sci. (Tech.), Prof.
Shokin Yu. I., Member of RAS,
Dr. Sci. (Tech.), Prof.
Yusupov R. M., Corresp. Member of RAS,
Dr. Sci. (Tech.), Prof.

Editorial Board Members:

Avdoshin S. M., Cand. Sci. (Tech.), Ass. Prof.
Antonov B. I.
Barsky A. B., Dr. Sci. (Tech.), Prof.
Vasenin V. A., Dr. Sci. (Phys.-Math.), Prof.
Galushkin A. I., Dr. Sci. (Tech.), Prof.
Dimitrienko Yu. I., Dr. Sci. (Phys.-Math.), Prof.
Domrachev V. G., Dr. Sci. (Tech.), Prof.
Zagidullin R. Sh., Cand. Sci. (Tech.), Ass. Prof.
Zarubin V. S., Dr. Sci. (Tech.), Prof.
Isaenko R. O., Cand. Sci. (Tech.)
Karpenko A. P., Dr. Sci. (Phys.-Math.), Prof.
Kolin K. K., Dr. Sci. (Tech.)
Kulagin V. P., Dr. Sci. (Tech.), Prof.
Kureichik V. M., Dr. Sci. (Tech.), Prof.
Kukhareno B. G., Cand. Sci. (Phys.-Math.)
Ljvovich Ya. E., Dr. Sci. (Tech.), Prof.
Mikhailov B. M., Dr. Sci. (Tech.), Prof.
Nechaev V. V., Cand. Sci. (Tech.), Ass. Prof.
Ryabov G. G., Corresp. Member of RAS,
Dr. Sci. (Tech.), Prof.
Sokolov B. V., Dr. Sci. (Tech.)
Uskov V. L. (USA), Dr. Sci. (Tech.)
Fomichev V. A., Dr. Sci. (Tech.), Prof.
Chermoshentsev S. F., Dr. Sci. (Tech.), Prof.
Shilov V. V., Cand. Sci. (Tech.), Ass. Prof.

Editors:

Bezmenova M. Yu.
Grigorin-Ryabova E. V.
Lysenko A. V.
Chugunova A. V.

Complete Internet version of the journal at site: <http://novtex.ru/IT>.

According to the decision of the Higher Certifying Commission of the Ministry of Education of Russian Federation, the journal is inscribed in "The List of the Leading Scientific Journals and Editions wherein Main Scientific Results of Theses for Doctor's or Candidate's Degrees Should Be Published"

МОДЕЛИРОВАНИЕ И ОПТИМИЗАЦИЯ MODELING AND OPTIMIZATION

УДК 004.414.23

А. Д. Иванников, д-р техн. наук, зам. директора по научной работе, e-mail: ADI@iprm.ru,
А. Л. Стемповский, д-р техн. наук, академик РАН, директор,
Институт проблем проектирования в микроэлектронике РАН

Формализация задачи отладки проектов цифровых систем

Приведена модель требуемого внешнего поведения проектируемой цифровой системы, заданной схемой соединения цифровых блоков и программным обеспечением. В качестве такой модели предлагается использовать семейство стационарных динамических систем, которое задается множеством терминальных переменных, множеством значений этих переменных, а также множеством допустимых взаимодействий цифровой системы с внешней средой. Задача отладки проектов цифровых систем формулируется как задача сравнения двух семейств стационарных динамических систем, с одной стороны, семейства, заданного требованиями технического задания на проектирование, а с другой стороны, семейства, определяемого схемой соединения блоков и текстом программного обеспечения цифровой системы.

Ключевые слова: моделирование цифровых систем, отладка проектов, модель цифровой системы, сложные цифровые схемы и микросистемы, стационарные динамические системы, допустимые взаимодействия

Введение

При проектировании сложных цифровых схем и микросистем (в ряде случаев будем использовать обобщенный термин — цифровых систем) разработчик должен обеспечить, прежде всего, требуемое внешнее поведение цифровой системы, т. е. требуемое взаимодействие системы с внешней средой. При этом существенным является как последовательность выходных сигналов цифровой системы, так и моменты времени появления и изменений этих сигналов. Временные ограничения обычно задаются интервалами значений. Например, требуется, чтобы выдача определенных данных цифровой системой на исполнительное устройство осуществлялась после, но не более чем через 2 мкс после подачи на цифровую систему сигнала запроса. Логические последовательности сигналов, которыми сложные цифровые схемы и микросистемы (цифровые системы) обмениваются с внешней средой, зависят как от внешних сигналов, поступающих на цифровую систему, так и от ее внутреннего состояния.

В связи с изложенным выше математической моделью внешнего поведения проектируемой цифровой системы может служить семейство стационарных динамических систем, что впервые предложено в работе [1].

Будем использовать приведенное в работе [2] определение динамической системы в виде

$$\Sigma = (\mathbf{T}, \mathbf{X}, \mathbf{U}, \mathbf{\Omega}, \mathbf{Y}, \mathbf{\Gamma}, \varphi, \eta),$$

где \mathbf{T} — множество моментов времени, в общем случае действительная полуось $\{t | t \geq 0\}$; \mathbf{X} — множество внутренних состояний; \mathbf{U} — множество мгновенных значений входных воздействий, называемое также множеством входных состояний; $\mathbf{\Omega}$ — множество отображений, каждое из которых $\omega: \mathbf{T} \rightarrow \mathbf{U}$ есть допустимое входное воздействие; \mathbf{Y} — множество мгновенных значений выходных воздействий, называемое также множеством выходных состояний; $\mathbf{\Gamma}$ — множество отображений, каждое из которых $\gamma: \mathbf{T} \rightarrow \mathbf{Y}$ есть возможное выходное воздействие; $\varphi: \mathbf{T} \times \mathbf{T} \times \mathbf{X} \times \mathbf{\Omega} \rightarrow \mathbf{X}$ — переходная функция состояния, ее значениями служат внутренние состояния $x(t) = \varphi(t, \tau, x, \omega)$, $x(t) \rightarrow \mathbf{X}$, в которых оказывается система в момент времени t , $t \in \mathbf{T}$, если в начальный момент времени τ , $\tau \in \mathbf{T}$, она была в начальном состоянии $x(\tau) \in \mathbf{X}$ и на нее действовало входное воздействие $\omega \in \mathbf{\Omega}$; $\eta: \mathbf{T} \times \mathbf{X} \rightarrow \mathbf{Y}$ — выходное отображение, определяющее выходные воздействия $y(t) = \eta(t, x(t))$.

Динамическая система является стационарной, если ее реакция на заданный отрезок входного воздействия при условии, что система находится в заданном внутреннем состоянии, не зависит от начального момента времени [2]. В частности, переходная функция принимает вид $\varphi: \mathbf{T} \times \mathbf{X} \times \mathbf{\Omega} \rightarrow \mathbf{X}$, ее значениями служат состояния $x(t) = \varphi(t, x, \omega)$, где x — состояние системы в момент $t = 0$, выходное отображение $\eta: \mathbf{X} \rightarrow \mathbf{Y}$ не зависит от времени.

В приведенном определении динамической системы каждому внутреннему состоянию x однознач-

но соответствует мгновенное значение выходного воздействия y . Более удобным для нас является использование выходного отображения в виде $\eta: \mathbf{X} \rightarrow \mathbf{Y}$, предложенное в работе [3]. Такой вид выходного отображения не искажает приведенного выше определения стационарной динамической системы и позволяет получить в качестве частного случая последней конечный автомат Мили.

Модель внешнего поведения цифровой системы

Математическое описание внешнего поведения семейства стационарных динамических систем, являющегося математической моделью цифровой системы, включает следующее.

1. Конечное множество терминальных переменных \mathbf{P} , описывающих внешние линии и шины цифровой системы. Переменная $p \in \mathbf{P}$ всегда имеет одно из значений конечного множества \mathbf{Z}_p , причем $\mathbf{Z}_p = \mathbf{Z}_p^* \times \text{DIR}_p$, где \mathbf{Z}_p^* — множество логических значений переменной p , в частности, для шины из n линий имеется 2^n значений или $2^n + 1$ значений, если шина имеет состояние с высоким выходным сопротивлением; DIR_p — множество направлений передачи сигнала. Во множестве \mathbf{P} можно выделить три непересекающихся подмножества: $\mathbf{P}' = \{p | \text{DIR}_p = \{1\}\}$; $\mathbf{P}'' = \{p | \text{DIR}_p = \{0\}\}$; $\mathbf{P}''' = \{p | \text{DIR}_p = \{0,1\}\}$, причем $\mathbf{P} = \mathbf{P}' \cup \mathbf{P}'' \cup \mathbf{P}'''$.

Терминальные переменные подмножеств \mathbf{P}' , \mathbf{P}'' , \mathbf{P}''' описывают входные, выходные и двунаправленные шины и линии цифровой системы соответственно; направление передачи $\text{dir}_p = 1$ указывает на входной, а $\text{dir}_p = 0$ — на выходной сигнал. Состояние шины или линии p с высоким выходным сопротивлением представляется значением из множества \mathbf{Z}_p^* при $\text{dir}_p = 0$.

Множество мгновенных значений терминальных переменных, т. е. внешних состояний цифровой системы, есть $\mathbf{Q} = \prod_{p \in \mathbf{P}} \mathbf{Z}_p$.

2. Множество моментов времени $\mathbf{T} = \{t | t \geq 0\}$.

3. Множество Ψ допустимых взаимодействий с внешней средой, каждое из которых есть отображение $\psi: [0, t) \rightarrow \mathbf{Q}$, $t \in \mathbf{T}$. Каждое ψ может быть представлено как конечное множество отображений $\psi_p: [0, t) \rightarrow \mathbf{Z}_p$, $p \in \mathbf{P}$, $t \in \mathbf{T}$.

Отличие допустимых взаимодействий друг от друга может быть обусловлено:

- различной длительностью взаимодействий;
- различной последовательностью функций, выполняемых цифровой системой в зависимости от внешних сигналов;
- различной последовательностью появления ряда входных или выходных сигналов, если эта последовательность безразлична, при выполнении одной и той же последовательности функций; наличием безразличных входных событий (переключений

сигналов), не оказывающих влияния на функционирование цифровой системы;

— разбросом временных параметров составляющих цифровой системы, что обуславливает разброс моментов времени появления выходных сигналов, и разбросом временных параметров внешней среды, что обуславливает различное время реакции внешней среды на выходные сигналы цифровой системы.

Таким образом, внешнее поведение семейства стационарных динамических систем, являющихся математической моделью цифровой системы, при известных \mathbf{T} , \mathbf{P} , $\{\mathbf{Z}_p | p \in \mathbf{P}\}$ задается множеством допустимых взаимодействий Ψ .

Модель допустимых взаимодействий цифровой системы

Рассмотрим подробнее задание допустимых взаимодействий $\psi: [0, t) \rightarrow \mathbf{Q}$. Событием по переменной p назовем изменение переменной p со значения $z_1 \in \mathbf{Z}_p$ на значение $z_2 \in \mathbf{Z}_p$ в момент времени t .

Обозначим такое событие χ_{p, z_1, z_2}^t .

В цифровых системах для каждого конечного временного интервала число событий по терминальным переменным, т. е. число изменений их значений, конечно. В связи с этим любое допустимое взаимодействие $\psi \in \Psi$ может быть представлено в виде вектора $(z_{p_1}^H, \dots, z_{p_k}^H)$ начальных значений переменных p_1, \dots, p_k (k — мощность множества \mathbf{P}) в момент времени $t = 0$ и последовательности событий по переменным множества \mathbf{P} с конечным числом событий за любой конечный интервал времени:

$$\psi = (z_{p_1}^H, \dots, z_{p_k}^H), \chi_{p_{i_1}, z_{j_1}, z_{j_2}}^{t_1}, \chi_{p_{i_2}, z_{j_3}, z_{j_4}}^{t_2}, \chi_{p_{i_3}, z_{j_5}, z_{j_6}}^{t_3}, \dots, (1)$$

где $t_1 \leq t_2 \leq t_3 \leq \dots$ — упорядоченная последовательность времен событий; $p_{i_1}, p_{i_2}, p_{i_3}$ — переменные, принадлежащие множеству \mathbf{P} ; $z_{j_1}, z_{j_3}, z_{j_5}, \dots$ — значения переменных непосредственно перед событием; $z_{j_2}, z_{j_4}, z_{j_6}, \dots$ — значения переменных непосредственно после события, причем при выделении из (1) начального значения любой терминальной переменной p_i и подпоследовательности событий по этой переменной получаем отображение $\psi_{p_i}: [0, t) \rightarrow \mathbf{Z}_{p_i}$, которое имеет вид

$$\psi_{p_i} = (z_{p_i}^H), \chi_{p_i, z_{\tau_1}, z_{\tau_2}}^{\tau_1}, \chi_{p_i, z_{\tau_2}, z_{\tau_3}}^{\tau_2}, \chi_{p_i, z_{\tau_3}, z_{\tau_4}}^{\tau_3}, \dots$$

где $z_{p_i}^H$ — начальные значения переменной p_i ; $\tau_1 < \tau_2 < \tau_3 < \dots$ — упорядоченная последовательность времен событий по данной переменной; $z_{\tau_1}, z_{\tau_2}, z_{\tau_3}, \dots$ — значения переменной p_i из множества \mathbf{Z}_{p_i} .

Представим последовательность (1) в виде двух подпоследовательностей: последовательности входных для цифровой системы событий — входного воздействия

$$\omega = (z_{p_1}^H, \dots, z_{p_n}^H), \chi_{p_{i_1}, z_{j_1}, z_{j_2}}^{t_1}, \chi_{p_{i_2}, z_{j_3}, z_{j_4}}^{t_2}, \chi_{p_{i_3}, z_{j_5}, z_{j_6}}^{t_3}, \dots, (2)$$

где $z_{p_1}^H, \dots, z_{p_n}^H$ — начальные значения переменных множества $\mathbf{P}' \cup \mathbf{P}''$, причем для переменных множества \mathbf{P}'' логические значения задаются только в том случае, если в начальный момент времени у них $dir_p = 1$; в противном случае для них достаточно задания направления передачи $dir_p = 0$; n —

число переменных множества $\mathbf{P}' \cup \mathbf{P}''$; $\chi_{p_{i_1}, z_{j_1}, z_{j_2}}^{t_1}, \chi_{p_{i_2}, z_{j_3}, z_{j_4}}^{t_2}, \chi_{p_{i_3}, z_{j_5}, z_{j_6}}^{t_3}, \dots$ — события по переменным множества \mathbf{P}' , а также события по переменным множества \mathbf{P}'' , для которых начальное, конечное или оба значения в событии имеют $dir_p = 1$; $t_1 \leq t_2 \leq t_3 \leq \dots$ — упорядоченная последовательность времен входных событий, и последовательности выходных событий — выходного воздействия

$$\gamma = (z_{p_1}^H, \dots, z_{p_m}^H), \chi_{p_{i_1}, z_{j_1}, z_{j_2}}^{t_1}, \chi_{p_{i_2}, z_{j_3}, z_{j_4}}^{t_2}, \chi_{p_{i_3}, z_{j_5}, z_{j_6}}^{t_3}, \dots, (3)$$

где $z_{p_1}^H, \dots, z_{p_m}^H$ — начальные значения переменных множества $\mathbf{P}'' \cup \mathbf{P}'''$, причем для переменных множества \mathbf{P}''' логические значения задаются только в том случае, если в начальный момент времени у них $dir_p = 0$, в противном случае для них достаточно задания направления передачи $dir_p = 1$; m — число переменных множества $\mathbf{P}'' \cup \mathbf{P}'''$; $\chi_{p_{i_1}, z_{j_1}, z_{j_2}}^{t_1}, \chi_{p_{i_2}, z_{j_3}, z_{j_4}}^{t_2}, \chi_{p_{i_3}, z_{j_5}, z_{j_6}}^{t_3}, \dots$ — события по переменным множества \mathbf{P}'' , а также события по переменным множества \mathbf{P}''' , для которых начальное, конечное или оба значения в событии имеют $dir_p = 0$; $t_1 \leq t_2 \leq t_3 \leq \dots$ — упорядоченная последовательность времен выходных событий.

Полученные входное ω и выходное γ воздействия могут содержать одинаковые события $\chi_{p_{i_1}, z_{j_1}, z_{j_2}}^t$ по переменным множества \mathbf{P}''' , если одно из значений z_{j_1}, z_{j_2} имеет $dir_p = 1$, а другое $dir_p = 0$.

Входное воздействие есть $\omega: [0, t) \rightarrow \mathbf{U}, t \in \mathbf{T}, \mathbf{U} = (\prod_{p \in \mathbf{P}'} \mathbf{Z}_p) \times (\prod_{p \in \mathbf{P}''} ((\mathbf{Z}_p^* \times \{1\}) \cup \{0\}))$; \mathbf{U} — множество мгновенных значений входных воздействий, называемое также множеством входных состояний;

состояние переменной $p \in \mathbf{P}''$, равное 0, указывает на передачу сигнала на выход и не определяет логического значения переменной.

Выходное воздействие есть $\gamma: [0, t) \rightarrow \mathbf{Y}, t \in \mathbf{T}, \mathbf{Y} = (\prod_{p \in \mathbf{P}''} \mathbf{Z}_p) \times (\prod_{p \in \mathbf{P}'''} ((\mathbf{Z}_p^* \times \{0\}) \cup \{1\}))$; \mathbf{Y} — множество мгновенных значений выходных воздействий, называемое также множеством выходных состояний; состояние переменной $p \in \mathbf{P}$, равное 1, указывает на передачу сигнала на вход и не определяет логического значения переменной.

Часто моменты подачи входных сигналов на цифровую систему определяются готовностью системы принять эти сигналы, на что указывают определенные выходные сигналы системы. Выполнение какой-либо операции, например считывания данных цифровой системой, может инициироваться не сигналами внешней среды, а самой системой. В связи с этим использование в качестве аргументов функционирования цифровой системы входных воздействий не всегда удобно.

Выделим из последовательности событий (1) — взаимодействия ψ — последовательность выходных событий, которые по заданному протоколу обмена обуславливают моменты времени входных событий. Назовем переменные, по которым происходят эти выходные события, переменными управления обменом $\mathbf{P}^0, \mathbf{P}^0 \subset \mathbf{P}'' \cup \mathbf{P}'''$. Выделенную из (1) подпоследовательность событий вместе с начальными значениями переменных управления обменом назовем выходным воздействием управления обменом

$\gamma^0 = (z_{p_1}^H, \dots, z_{p_q}^H), \chi_{p_{i_1}, z_{j_1}, z_{j_2}}^{ot_1}, \chi_{p_{i_2}, z_{j_3}, z_{j_4}}^{ot_2}, \chi_{p_{i_3}, z_{j_5}, z_{j_6}}^{ot_3}, \dots, (4)$

где $z_{p_1}^H, \dots, z_{p_q}^H$ — начальные значения переменных множества \mathbf{P}^0 ; q — мощность множества \mathbf{P}^0 ; $\chi_{p_{i_1}, z_{j_1}, z_{j_2}}^{ot_1}, \chi_{p_{i_2}, z_{j_3}, z_{j_4}}^{ot_2}, \chi_{p_{i_3}, z_{j_5}, z_{j_6}}^{ot_3}, \dots$ — события по переменным множества \mathbf{P}^0 ; $t_1 \leq t_2 \leq t_3 \leq \dots$ — упорядоченная последовательность времен выходных событий управления обменом.

Выходное воздействие управления обменом есть $\gamma^0: [0, t) \rightarrow \mathbf{Y}^0, t \in \mathbf{T}, \mathbf{Y}^0 = (\prod_{p \in \mathbf{P}^0 \cap \mathbf{P}''} \mathbf{Z}_p) \times (\prod_{p \in \mathbf{P}^0 \cap \mathbf{P}'''} ((\mathbf{Z}_p \times \{0\}) \cup \{1\}))$; \mathbf{Y}^0 — множество мгновенных значений выходных воздействий управления обменом.

Объединим подпоследовательности событий входного воздействия ω (2) и выходного воздействия обмена γ^0 (4) одного и того же ψ и вместе с начальными значениями соответствующих переменных

назовем полученную подпоследовательность событий входным взаимодействием

$$\mu = (z_{p_1}^H, \dots, z_{p_{n+q}}^H), \chi_{p_{i_1}, z_{j_1}, z_{j_2}}^{t_1}, \chi_{p_{i_2}, z_{j_3}, z_{j_4}}^{t_2}, \chi_{p_{i_3}, z_{j_5}, z_{j_6}}^{t_3}, \dots, (5)$$

где $\chi_{p_{i_1}, z_{j_1}, z_{j_2}}^{t_1}, \chi_{p_{i_2}, z_{j_3}, z_{j_4}}^{t_2}, \chi_{p_{i_3}, z_{j_5}, z_{j_6}}^{t_3}, \dots$ — входные события и выходные события управления обменом; $t_1 \leq t_2 \leq t_3 \leq \dots$ — упорядоченная последовательность времен событий входного взаимодействия.

Входное взаимодействие есть $\mu: [0, t) \rightarrow \mathbf{U} \times \mathbf{Y}^0$, $t \in \mathbf{T}$.

В дальнейшем в качестве аргументов функционирования цифровых систем будем использовать входные взаимодействия, что дает возможность рассматривать режимы работы, инициируемые как внешними входными сигналами, так и самими цифровыми системами. В простейшем случае, когда подача сигналов на проектируемую систему не зависит от ее выходных сигналов, множество выходных переменных управления обменом \mathbf{P}^0 пусто и входное взаимодействие μ совпадает с входным воздействием ω .

При известных $\mathbf{T}, \mathbf{P}, \{Z_p | p \in \mathbf{P}\}$ каждое допустимое взаимодействие ψ описанным выше образом определяет входное ω и выходное γ воздействия, выходное воздействие управления обменом γ^0 и входное взаимодействие ω на общем интервале времени $[0, t)$. Тогда множество допустимых взаимодействий Ψ задает множество входных воздействий Ω , выходных воздействий Γ , выходных воздействий управления обменом Γ^0 и входных взаимодействий \mathbf{M} . Кроме того, на $\Omega \times \Gamma, \Omega \times \Gamma^0$ и $\mathbf{M} \times \Gamma$ определены отношения b, b' и b'' соответственно: $(\omega, \gamma) \in b$, если ω и γ образуют допустимое взаимодействие $\psi \in \Psi$; $(\omega, \gamma^0) \in b'$, если ω и γ^0 образуют входное взаимодействие $\mu \in \mathbf{M}$; $(\mu, \gamma) \in b''$, если μ и γ образуют допустимое взаимодействие $\psi \in \Psi$. Множество допустимых взаимодействий может быть задано как $\Psi = (\Omega, \Gamma, b)$ или как $\Psi = (\mathbf{M}, \Gamma, b'')$; множество входных взаимодействий — как $\mathbf{M} = (\Omega, \Gamma^0, b')$.

Каждому допустимому входному взаимодействию μ соответствует непустое множество выходных воздействий $\Gamma_\mu = \{\gamma | b''(\mu, \gamma)\}$. Все элементы $\gamma \in \Gamma_\mu$ различаются только моментами времени выходных событий в допустимых пределах и возможно значениями выходных переменных на определенных отрезках времени, если эти значения безразличны. При известных $\mathbf{T}, \mathbf{P}, \{Z_p | p \in \mathbf{P}\}$ множество допустимых взаимодействий

$$\Psi = (\mathbf{M}, \{\Gamma_\mu | \mu \in \mathbf{M}\}), (6)$$

причем при $\mu \in \mathbf{M}, \Gamma_\mu \neq \emptyset$.

Учет последовательности выполняемых функций и временных ограничений в модели допустимых взаимодействий

Каждая цифровая система в процессе функционирования выполняет ту или иную последовательность функций (операций) из конечного алфавита функций \mathbf{K} , свойственных конкретной системе. Выполнение каждой функции вызывается одним из входных взаимодействий определенного класса, причем каждое входное взаимодействие этого класса содержит конечное число событий.

Обозначим через f конечную последовательность функций, а через \mathbf{F} в общем случае счетное множество конечных последовательностей f . Каждая последовательность функций f , начинающаяся с момента времени $t = 0$ (например включения питания), задается по крайней мере одним входным взаимодействием $\mu^f \in \mathbf{M}$. Этот факт следует из того, что \mathbf{M} содержит все допустимые входные взаимодействия для любой допустимой последовательности функций цифровой системы.

В большинстве случаев одни и те же функции цифровая система может выполнять с различными наборами данных, что обуславливает задание различными μ одной и той же последовательности функций f . В связи с тем, что для различных экземпляров цифровой системы задержки выходных событий управления обменом относительно входных событий различаются в определенных пределах, а также в связи с допустимостью варьирования моментов времени входных событий относительно друг друга и относительно выходных событий управления обменом, множество \mathbf{M} содержит континуальное подмножество $\mathbf{M}^f \subset \mathbf{M}$ входных взаимодействий, каждое из которых вызывает выполнение цифровой системой конечной последовательности функций f . Множество входных взаимодействий может быть представлено в виде

$$\mathbf{M} = \bigcup_{f \in \mathbf{F}} \mathbf{M}^f; \mathbf{M}^{f'} \cap \mathbf{M}^{f''} = \emptyset \text{ при } f' \neq f''. (7)$$

Входное взаимодействие $\mu \in \mathbf{M}^f$ содержит конечное множество событий

$$\{\chi_{p_{i_1}, z_{j_1}, z_{j_2}}^{t_1}, \dots, \chi_{p_{i_n}, z_{j_{2n-1}}, z_{j_{2n}}}^{t_n}\}, (8)$$

где n — число событий в μ .

Множество \mathbf{M}^f содержит также входные взаимодействия, времена событий в которых (событий множества (8)) различаются в определенных пределах. Ограничения на эти различия могут быть заданы в виде

$$t_{\min}^{l,m} \leq t_m - t_l \leq t_{\max}^{l,m}, (l, m) \in \mathbf{C}, \mathbf{C} \subset \{1, 2, \dots, n\} \times \{1, 2, \dots, n\}, (9)$$

где $t_{\min}^{l,m}, t_{\max}^{l,m}$ — минимально и максимально допустимые промежутки времени между l -м и m -м событиями; \mathbf{C} — конечное множество пар событий множества (8), для которых заданы временные ограничения.

Входное взаимодействие μ множества \mathbf{M}^f кроме событий (8) может содержать на определенных интервалах времени безразличные события, не влияющие на функционирование цифровой системы. Пригодный для практики метод задания всего класса \mathbf{M}^f должен быть рассмотрен отдельно, а сейчас отметим, что во множестве ограничений (9) фигурируют времена существенных событий, т. е. событий множества (8).

Выделим из множества \mathbf{C} все пары (l, m) , для которых t_m есть время выходного события управления обменом, в множество $\mathbf{C}_{\text{вых}}$, а все пары (l, m) , для которых t_m есть время входного события, в множество $\mathbf{C}_{\text{вх}}$. Тогда ограничения на моменты времени выходных событий обмена есть

$$t_{\min}^{l,m} \leq t_m - t_l \leq t_{\max}^{l,m}, (l, m) \in \mathbf{C}_{\text{вых}}, \quad (10)$$

а ограничения на моменты времени входных событий

$$t_{\min}^{l,m} \leq t_m - t_l \leq t_{\max}^{l,m}, (l, m) \in \mathbf{C}_{\text{вх}}. \quad (11)$$

Рассмотрим пространство $\mathbf{G} = \prod_{\mathbf{C}_{\text{вых}}} \{t_m - t_l | t_m - t_l \geq 0\}$. Каждая точка $g \in \mathbf{G}$ определяет конкретные значения задержек выходных событий управления обменом. В пространстве \mathbf{G} выделим область $\mathbf{G}_f \in \mathbf{G}$, для всех точек которой выполняются ограничения (10). Область \mathbf{G}_f определяет задержки выходных событий управления обменом, допустимые по техническому заданию. Если техническое задание непротиворечиво, а мы рассматриваем именно такой случай, то $\mathbf{G}_f \neq \emptyset$.

Для любой точки $g \in \mathbf{G}_f$ в связи с допустимостью таких задержек выходных событий управления обменом существует непустое множество входных взаимодействий \mathbf{M}_g^f , обеспечивающих выполнение системой последовательности функций f :

$$\mathbf{M}^f = \bigcup_{g \in \mathbf{G}_f} \mathbf{M}_g^f, \mathbf{M}_g^f \neq \emptyset,$$

где \mathbf{M}_g^f — множество входных взаимодействий, обеспечивающих выполнение цифровой системой конечной последовательности функций f при фиксированных задержках выходных событий управления обменом, определяемых $g \in \mathbf{G}_f$.

Таким образом, множество допустимых входных взаимодействий представимо в виде

$$\mathbf{M} = \bigcup_{f \in \mathbf{F}} \mathbf{M}^f; \mathbf{M}^{f'} \cap \mathbf{M}^{f''} = \emptyset \text{ при } f' \neq f''; \quad (12)$$

$$\bigcup_{g \in \mathbf{G}_f} \mathbf{M}_g^f; \mathbf{G}_f \neq \emptyset, \mathbf{M}_g^f \neq \emptyset \text{ при } f \in \mathbf{F}.$$

Структура множества входных взаимодействий (12) вместе с представлением множества допустимых взаимодействий (6) позволяют сформулировать задачу отладки цифровых систем на этапе проектирования в терминах теории динамических систем.

Задача отладки в терминах теории динамических систем

Пусть техническим заданием определено требуемое поведение цифровой системы в виде $(\mathbf{T}, \mathbf{P}, \{\mathbf{Z}_p | p \in \mathbf{P}\}, \mathbf{M}, \{\Gamma_\mu | \mu \in \mathbf{M}\})$, т. е. для каждого допустимого входного взаимодействия μ определено непустое множество возможных выходных воздействий Γ_μ . Задано также множество конечных последовательностей выполняемых цифровой системой функций \mathbf{F} , причем $\mathbf{M} = \bigcup_{f \in \mathbf{F}} \mathbf{M}^f; \mathbf{M}^{f'} \cap \mathbf{M}^{f''} = \emptyset$ при $f' \neq f''$. Кроме того, для любого f

$$\mathbf{M}^f = \bigcup_{g \in \mathbf{G}_f} \mathbf{M}_g^f, \mathbf{G}_f \neq \emptyset; \mathbf{M}_g^f \neq \emptyset.$$

В свою очередь, спроектированная цифровая система задается в виде принципиальной схемы \mathbf{C}_x технических средств и текста программного или микропрограммного обеспечения $\mathbf{\Pi}$, т. е. в виде пары $(\mathbf{C}_x, \mathbf{\Pi})$.

Схема технических средств

$$\mathbf{C}_x = (\mathbf{И}, \mathbf{У}, \mathbf{P}, \{\mathbf{Z}_p | p \in \mathbf{P}\}, \xi), \quad (13)$$

где $\mathbf{И}$ — конечный комплект цифровых блоков; $\mathbf{У}$ — конечное множество узлов принципиальной схемы; \mathbf{P} — множество переменных в терминальных узлах (шинах и линиях) схемы, $\mathbf{P} \in \mathbf{У}$; \mathbf{Z}_p — конечное множество значений переменной p . Причем $\mathbf{Z}_p = \mathbf{Z}_p^* \times \text{DIR}_p$, \mathbf{Z}_p^* — множество логических значений; DIR_p — множество направлений передачи сигнала. Для входных шин и линий $\text{DIR}_p = \{1\}$, для выходных $\text{DIR}_p = \{0\}$, для двунаправленных $\text{DIR}_p = \{0, 1\}$; ξ — отображение множества выводов цифровых блоков во множество узлов.

В (13) для всех переменных p , описывающих шины из нескольких линий, под узлом u понимается соединение нескольких шин с одинаковым числом линий, что физически соответствует числу узлов, равному числу линий в шинах.

Программное или микропрограммное обеспечение

$$\mathbf{\Pi} = (\mathbf{Я}, \{z_{я} | я \in \mathbf{Я}\}),$$

где $\mathbf{Я}$ — конечное множество ячеек памяти, хранящих программу или микропрограмму; $z_{я}$ — состояние ячейки памяти $я$.

Программное или микропрограммное обеспечение может быть задано также в виде текста на ассемблере (микроассемблере) или языке высокого уровня. При известном трансляторе множество ячеек памяти, хранящих программу, и их состояния также однозначно определены.

Спроектированную цифровую систему в целом можно представить как

$$(\mathbf{Сх}, \mathbf{\Pi}) = (\mathbf{И}, \mathbf{У}, \mathbf{Р}, \{\mathbf{Z}_p | p \in \mathbf{P}\}, \xi, \mathbf{Я}, \{z_{я} | я \in \mathbf{Я}\}). (14)$$

Спроектированная цифровая система $(\mathbf{Сх}, \mathbf{\Pi})$, представленная в виде (14), определяет некоторое семейство $\mathbf{М}$ стационарных динамических систем, которое может быть задано своим внешним поведением в виде (6), (7). Будем обозначать множество входных взаимодействий и выходных воздействий семейства динамических систем $\mathbf{М}$ как \mathbf{M}_M и $\mathbf{\Gamma}_M$ соответственно, а множество возможных выходных воздействий для заданного входного взаимодействия μ как $\mathbf{\Gamma}_{\mu M}$. При сравнении внешнего поведения спроектированной цифровой системы с требуемым по техническому заданию будем считать, что в обоих случаях используются одинаковые $\mathbf{T}, \mathbf{P}, \{\mathbf{Z}_p | p \in \mathbf{P}\}$.

Цифровая система может быть спроектирована таким образом (хотя это является ошибкой), что при каких-то допустимых сочетаниях задержек блоков, в какие-то интервалы времени на одном или нескольких выходах логические сигналы будут не определены. Дополним множество значений переменных множеств \mathbf{P}'' и \mathbf{P}''' значением u (неопределенный сигнал) следующим образом:

$$\tilde{\mathbf{Z}}_p = (\mathbf{Z}_p^* \cup \{u\}) \times \{0\} \text{ для } p \in \mathbf{P}'';$$

$$\tilde{\mathbf{Z}}_p = (\mathbf{Z}_p^* \cup \{u\}) \times \{0\} \cup (\mathbf{Z}_p^* \times \{1\}) \text{ для } p \in \mathbf{P}''';$$

$$\tilde{\mathbf{Z}}_p = \mathbf{Z}_p \text{ для } p \in \mathbf{P}'.$$

Тогда и в случае отсутствия в какие-то интервалы времени определенных логических сигналов на выходах цифровой системы выходные воздействия представимы в виде (3), если вместо $\{\mathbf{Z}_p | p \in \mathbf{P}\}$ использовать $\{\tilde{\mathbf{Z}}_p | p \in \mathbf{P}\}$. Множество возможных выходных воздействий при известном μ , рассматриваемых в алфавитах $\tilde{\mathbf{Z}}_p$ будем обозначать $\tilde{\mathbf{\Gamma}}_{\mu M}$. При этом для каждого $\mu \in \mathbf{M}_M$ существует непустое множество $\tilde{\mathbf{\Gamma}}_{\mu M}$, состоящее из выходных воздействий, соответствующих различным сочетаниям задержек блоков цифровой системы. Выходные воздействия управления обменом по-прежнему будут рассматриваться в алфавитах \mathbf{Z}_p . В случае если в какой-то последовательности событий логическое значение выходной переменной управления обменом $p \in \mathbf{P}^0$ не определено, будем считать, что такая

последовательность входным взаимодействием не является.

Пусть пара $(\mathbf{Сх}, \mathbf{\Pi})$ определяет $\mathbf{T}, \mathbf{P}, \{\mathbf{Z}_p | p \in \mathbf{P}\}, \{\tilde{\mathbf{Z}}_p | p \in \mathbf{P}\}, \mathbf{M}_M, \{\tilde{\mathbf{\Gamma}}_{\mu M} | \mu \in \mathbf{M}_M\}$, а также \mathbf{F}_M , такое, что $\mathbf{M}_M = \bigcup_{f \in \mathbf{F}_M} \mathbf{M}_M^f$; $\mathbf{M}_M^f \neq \emptyset$; $\mathbf{M}_M^{f'} \cap \mathbf{M}_M^{f''} = \emptyset$ при $f' \neq f''$.

Обозначим через τ вектор задержек множества блоков, входящих в схему технических средств цифровой системы, а через \mathfrak{Z} множество векторов возможных задержек блоков системы. Каждому $\tau \in \mathfrak{Z}$ соответствуют свои задержки выходных событий управления обменом. Следовательно, существует отображение $a_f: \mathfrak{Z} \rightarrow \mathbf{G}$. Если при каких-либо значениях задержек, т. е. при каком-либо $\tau \in \mathfrak{Z}$, логическое значение выходной переменной, участвующей во входном взаимодействии, не определено, то $a_f(\tau)$ не существует. Таким образом, отображение a_f в общем случае является частичным.

Для того чтобы при любых возможных сочетаниях задержек блоков выполнялись ограничения (12), необходимо, чтобы при $\tau \in \mathfrak{Z}$ $a_f(\tau) \in \mathbf{G}_f$.

Для каждого $\tau \in \mathfrak{Z}$ существует свое множество допустимых входных взаимодействий \mathbf{M}_{tM}^f и $\mathbf{M}_M^f = \bigcup_{t \in \mathbf{T}} \mathbf{M}_{tM}^f$, $\mathbf{M}_{tM}^f \neq \emptyset$.

Чтобы при любых возможных сочетаниях задержек блоков, используемых в цифровой системе, входные взаимодействия, допустимые по техническому заданию, были допустимы для спроектированной цифровой системы, необходимо следующее:

$$\mathbf{F}_M \supseteq \mathbf{F} \text{ при } \tau \in \mathfrak{Z}, \mathbf{M}_{tM}^f \supseteq \mathbf{M}_{af(t)}^f.$$

Кроме того, при правильном проектировании цифровой системы $\tilde{\mathbf{\Gamma}}_{\mu M} \subset \mathbf{\Gamma}_{\mu}$ для всех $\mu \in \mathbf{M} \cap \mathbf{M}_M$, где $\mathbf{M} \cap \mathbf{M}_M = \bigcup_{f \in \mathbf{F}} \bigcup_{t \in \mathbf{T}} \mathbf{M}_{af(t)}^f$. Выполнение этого соотношения в качестве необходимого условия требует, чтобы выходные воздействия γ , где $\gamma \in \tilde{\mathbf{\Gamma}}_{\mu M}$, не содержали значения u (не определено) какой-либо переменной. При этом условии алфавиты фактически используемых значений терминальных переменных $\tilde{\mathbf{Z}}_{p\text{факт}} = \mathbf{Z}_p$, $p \in \mathbf{P}$.

Задача отладки проекта цифровой системы может быть сформулирована следующим образом.

Доказать, что

$$\mathbf{F}_M \supseteq \mathbf{F}; \quad (15)$$

$$a_f(\tau) \in \mathbf{G}_f \text{ и } \mathbf{M}_{tM}^f \supseteq \mathbf{M}_{af(t)}^f \text{ при } \tau \in \mathfrak{Z}, f \in \mathbf{F}; \quad (16)$$

$$\tilde{\mathbf{\Gamma}}_{\mu M} \subseteq \mathbf{\Gamma}_{\mu} \text{ при } \mu \in \bigcup_{f \in \mathbf{F}} \bigcup_{t \in \mathbf{T}} \mathbf{M}_{af(t)}^f. \quad (17)$$

В случае если это неверно, внести в (Сх, П) такие исправления, чтобы соотношения (15)–(17) выполнялись.

Подзадачи отладки цифровых систем

Для того чтобы выполнить отладку проекта цифровой системы, необходимо решить следующие подзадачи:

- 1) проверить включение $\mathbf{F}_M \supseteq \mathbf{F}$;
- 2) выбрать такое конечное множество подмножеств входных взаимодействий \mathbf{M}^{f_j} , $j = 1, \dots, n$, что:
 - а) из того, что при $t \in \mathfrak{Z}$ $\mathbf{a}_{f_j}(t) \in \mathbf{G}_{f_j}$, $\mathbf{M}_{tM}^{f_j} \supseteq \mathbf{M}_{af(t)}^{f_j}$ для всех j следует, что при $t \in \mathfrak{Z}$ $\mathbf{a}_f(t) \in \mathbf{G}_f$ и $\mathbf{M}_{tM}^f \supseteq \mathbf{M}_{af(t)}^f$ для всех $f \in \mathbf{F}$;
 - б) из того, что $\tilde{\Gamma}_{\mu M} \subset \Gamma_{\mu}$ для $\mu \in \bigcup_{t \in \mathfrak{Z}} \mathbf{M}_{af(t)}^{f_j}$ и всех j следует, что $\tilde{\Gamma}_{\mu M} \subset \Gamma_{\mu}$ для всех $\mu \in \bigcup_{f \in \mathbf{F}} \bigcup_{t \in \mathfrak{Z}} \mathbf{M}_{af(t)}^f$;
- 3) проверить соотношения $\mathbf{a}_{f_j}(t) \in \mathbf{G}_{f_j}$ и $\mathbf{M}_{tM}^{f_j} \supseteq \mathbf{M}_{af(t)}^{f_j}$ при $t \in \mathfrak{Z}$ и $j = 1, \dots, n$;
- 4) проверить соотношения $\tilde{\Gamma}_{\mu M} \subset \Gamma_{\mu}$ при $\mu \in \bigcup_{t \in \mathfrak{Z}} \mathbf{M}_{af(t)}^{f_j}$ для $j = 1, \dots, n$;
- 5) при невыполнении условий, проверяемых в пп. 1, 3, 4, иметь возможность локализовать ошибки в (Сх, П) и исправить их, а затем повторить указанные действия.

Подзадача 2 решается путем представления технического задания в виде формальной спецификации, по которой осуществляется выбор конечного числа \mathbf{M}^{f_j} , $j = 1, \dots, n$, исходя из требуемого внешнего поведения цифровой системы.

Практическое решение подзадачи 1 основывается на проверке внутренних состояний цифровой системы, в которые последняя приходит после выполнения различных функций. При отладке эта подзадача решается во время моделирования цифровой системы на выбранном уровне.

Решение подзадачи 3 заключается в проверке появления выходных сигналов управления обменом и соблюдения ограничений на моменты переключений сигналов вида (9) во входных взаимодействиях при всех возможных сочетаниях задержек. Такая проверка должна проводиться для всех подмножеств входных взаимодействий \mathbf{M}^{f_k} , $k = 1, \dots, n$.

Подзадача 3 представляет собой задачу верификации временных диаграмм работы цифровой системы и решается путем моделирования схемы с учетом разброса задержек блоков [4].

Для решения подзадачи 4 осуществляется проверка включений $\tilde{\Gamma}_{\mu M} \in \Gamma_{\mu}$ для $\mu \in \bigcup_{t \in \mathfrak{Z}} \mathbf{M}_{af(t)}^{f_j}$ и конечного множества j .

Наиболее полным решением подзадачи 4 является путь формального доказательства. При этом в наиболее общем случае необходимо осуществить символическое вычисление $\tilde{\Gamma}_{\mu M}$ одновременно для всех $\mu \in \bigcup_{t \in \mathfrak{Z}} \mathbf{M}_{af(t)}^f$, а полученное $\tilde{\Gamma}_{\mu M}$ сравнить с Γ_{μ} , заданным формальной спецификацией также символически. Отладочным тестом в этом случае являлось бы все множество $\bigcup_{t \in \mathfrak{Z}} \mathbf{M}_{af(t)}^f = \mathbf{M}_M^f \cap \mathbf{M}^f$.

При таком подходе, назовем его полным символическим тестированием, необходимо проводить одновременное символическое вычисление как моментов времени выходных событий с учетом диапазонов задержек блоков, так и логических значений сигналов. Теоретически это возможно путем символического моделирования логики работы цифровой системы с нарастающей неопределенностью моментов времени событий.

Заключение

Предложенная формализация дает возможность проводить дальнейшие теоретические исследования в рассматриваемой области, осуществлять декомпозицию задачи и разрабатывать методы решения каждой ее части.

Естественно, что для практического применения данная формализация должна быть исследована и представлена последовательностью, состоящей из реализуемых программным способом этапов.

Список литературы

1. **Иванников А. Д.** Моделирование как средство отладки микропроцессорных систем // Микроэлектроника и полупроводниковые приборы / Под ред. А. А. Васенкова и Я. А. Федотова. М.: Радио и связь, 1984. Вып. 9. С. 289–300.
2. **Калман Р., Фалб П., Арбиб М.** Очерки по математической теории систем. М.: Мир, 1971. 400 с.
3. **Заде Л.** Понятие состояния в теории систем. — В кн.: Общая теория систем. М.: Мир, 1966. С. 49–65.
4. **Стемпковский А. Л., Гаврилов С. В., Глебов А. Л.** Методы логического и логико-временного анализа цифровых КМОП СБИС. М.: Наука, 2007. 220 с.

A. D. Ivannikov, Deputy Director, e-mail: ADI@ippm.ru,
A. L. Stempkovsky, Academician of Russian Academy of Sciences, Director,
Institute for Design Problems in Microelectronics of Russian Academy of Science

Formal Mathematical Representation for the Task of Digital System Projects Debugging

For a digital system, defined by digital blocks connection and written software, a model of the desired external behavior is introduced. As such a model it is proposed to use the family of stationary dynamical systems, defined by the terminal variables set, the sets of values of these variables, as well as the set of permissible digital system interactions with the environment. It is assumed that the digital system being designed should fulfill a finite alphabet of functions, each function is characterized by a subset of permissible interactions. On the basis of such an assumption the structure of the digital system permissible interactions is analyzed. Structurally, the project for a digital system being designed is defined by circuit diagram of connected digital blocks and software text. Properly designed digital system project also defines its external behavior as a family of stationary dynamical systems. Mathematically, digital system project debugging task is formulated as the task of comparing two families of stationary dynamical systems, on the one hand the family, defined by given technical specifications for the design, and on the other hand the family, defined by the circuit diagram of connected digital blocks and software text of a digital system.

Keywords: digital system simulation, design debugging, digital system model, complex digital circuits and microsystems, stationary dynamical systems, permissible interactions

References

1. **Ivannikov A. D.** Modelirovanie kak sredstvo otladki mikroprocessornih sistem. *Mikroelektronika i poluprovodnikovye pribory*. Pod red. A. A. Vasenkova i Y. A. Fedotova. M., 1984. V. 9. P. 289–300.
2. **Kalman R. E., Falb P. L., Arbib M. F.** Topics in Mathematical System Theory. McGraw-Hill Book Co. 1969. 358 p.

3. **Zade L.** Ponjatie sostojanja v teorii sistem. V kn.: *Obshaya teoriya sistem*. M.: Mir. 1966. P. 49–65.
4. **Stempkovsky A. L., Gavrilov S. V., Glebov A. L.** Metodi logicheskogo i logiko-vremennogo analiza cifrovih KMOP SBIS. M.: Nauka. 2007. 220 p.

УДК 004.657

Н. С. Дудаков, инженер-программист, ОАО "Концерн "Системпром",
e-mail: Nikolay.Dudakov@gmail.com,

Ю. Ю. Шумилов, д-р техн. наук, проф., e-mail: shumilovyy@gmail.com
Национальный исследовательский ядерный университет "МИФИ"

Гибридная система управления хранением и оптимизации данных в АСУ

Решается задача повышения эффективности систем управления базами данных при разработке АСУ. Описаны современные подходы к построению систем управления базами данных. Показано, что для ряда практических задач применение существующих открытых СУБД недостаточно эффективно. Предложено разделение данных и использование нескольких программных решений для улучшения производительности системы управления хранением данных. Предложена математическая модель системы, процесс обработки запросов в базе данных описывается в терминах теории массового обслуживания. В соответствии с моделью системы предложены критерии эффективности работы системы, позволяющие количественно оценить разбиение данных. Также разработана методика оптимизации разбиения данных, позволяющая получить оптимальное разбиение и, тем самым, улучшить производительность СУБД.

Ключевые слова: СУБД, архитектура баз данных, система массового обслуживания, нелинейное псевдобулево программирование

Системы управления хранением данных АСУ

При проектировании автоматизированных систем управления (АСУ) одним из наиболее важных моментов является управление доступом, хранением

и использованием информации. Функциональная и эффективная работа современных АСУ невозможна без специализированных средств хранения и обработки информации. С увеличением объемов информации и сложности производимых вычисле-

ний возрастают требования к базам данных (системам управления хранением данных, СУХД), которые являются ядром современных АСУ [1].

При работе АСУ доступ к хранимой информации так или иначе получают все приложения и модули системы: источники информации, расчетные модули, средства визуализации и т. д. СУХД (далее, в том числе хранилище данных) является централизующим звеном в работе АСУ, от характеристик его работы и удобства доступа к данным зависит работа всего комплекса, т. е. при большом числе обращений к хранилищу именно оно может быть узким местом комплекса с точки зрения производительности. Эффективность работы хранилища данных зависит от используемых программных средств, осуществляющих доступ к данным, модели данных и производительности используемого оборудования.

На настоящий момент существует большое число систем управления хранением данных, удовлетворяющих различным запросам, и, что немаловажно, большинство программных решений открыто или основано на открытой разработке [2]. Проектирование баз данных — динамично развивающаяся область создания программного обеспечения, различие задач порождает разнообразные по характеристикам и целям применения хранилища данных.

Архитектурные решения современных СУБД

Большинство информационных систем хранения данных построено на клиент-серверной архитектуре, предполагающей централизованное хранение данных на сервере и доступ к данным со стороны клиентских приложений для чтения и записи. Достоинствами клиент-серверной архитектуры являются надежность, безопасность, простота в обслуживании, малая нагрузка на клиентские рабочие места. Недостатками — большая нагрузка на сервер и высокие требования к надежности и пропускной способности каналов связи между клиентами и сервером. Клиент-серверная архитектура более удобна для хранения больших объемов редко запрашиваемой информации, так как частая передача больших объемов данных по сети может быть затруднена.

В современной информационной системе большинство запросов к серверу базы данных являются читающими [3]. Поэтому именно запросы на чтение в основном определяют загрузку сервера баз данных, и именно они делают сервер и сеть "узким местом" клиент-серверной системы с точки зрения ее производительности. Учитывая, в ряде случаев, невысокие пропускные характеристики каналов передачи данных и загруженность сервера, недостатки "классической" клиент-серверной системы существенно сказываются на работоспособности АСУ.

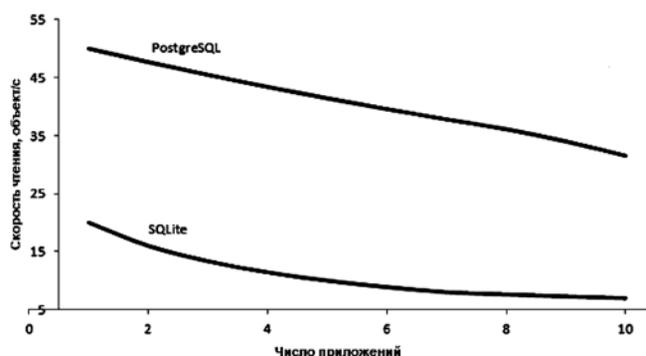
Подход, реализуемый в распределенных базах данных, — хранение копии базы данных на каждом рабочем месте — решает часть проблем, позволяя существенно снизить нагрузку на сервер за счет

однократной передачи изменений на подключенные рабочие места. Однако существенно труднее при этом решать задачу синхронизации клиентских и серверных копий базы данных. Также при подключении к серверу либо клиентское рабочее место запрашивает полную копию базы данных, либо необходимы механизмы документирования изменений и "частичных обновлений" для инициализации рабочих мест. В целом, в отличие от клиент-серверной архитектуры хранилища распределенные системы больше подходят для часто запрашиваемых данных сравнительно меньшего объема.

Иллюстрируя вышесказанное, можно отметить, что распределенные системы в значительной степени выигрывают у клиент-серверных по скорости чтения данных, особенно при повышении числа одновременно читающих приложений [4, 5]. Зависимость скорости чтения фиксированного числа объектов для базы данных PostgreSQL, сервер которой расположен удаленно, и локальной базы данных SQLite представлена на рисунке.

Ряд программных решений сочетают в себе возможности приведенных архитектур, осуществляя частичное разделение хранимых данных и располагая часть из них, в том числе, в локальных базах данных. В таких случаях фактором, определяющим производительность системы при фиксированных программных решениях, является алгоритм разделения данных (выделения локальной части). В большинстве подобных систем разделение данных осуществляется по принципу кэширования, т. е. динамического размещения часто запрашиваемых данных в локальных хранилищах [6]. Механизм кэширования позволяет оперировать только частотой запросов и не учитывает характеристики локальных хранилищ данных, при условии, что для локального хранения используется отличная от основной база данных. В ряде случаев осуществляется постоянное разделение данных исходя из предметной области и "примерных" свойств частей.

Также достаточно распространенным подходом является проектирование распределенных баз данных, при котором данные разделяются между серверами кластера таким образом, что каждый из серверов хранит часть данных и таблицу размеще-



Запросы на чтение для локальной и клиент-серверной баз данных

ния всех данных хранилища, осуществляя передачу запросов частям данных соответствующим серверам и последующую компиляцию результатов. Данный подход хорошо зарекомендовал себя, существует большое число как открытых [7], так и коммерческих [8] разработок по данному направлению. Основными достоинствами подхода являются снижение нагрузки на отдельные серверы и сравнительная легкость масштабирования системы до объемов данных порядка петабайт.

Разделение данных при проектировании АСУ

Основная сложность при разработке систем управления хранением данных состоит в разнородности хранимой информации и, как следствие, сложности построения универсального внутреннего механизма хранения данных (движка СУХД). Разнородность данных проявляется, например, в значительном различии темпов доступа к информации. В АСУ хранимые данные варьируются от сравнительно редко меняющихся объектов большого объема до меньших, запрашиваемых и изменяемых несколько раз в секунду. Также в ряде случаев могут потребоваться специализированные меры безопасности, вплоть до зашифрованного хранения части информации [9].

Рассматривая конкретные программные решения, можно отметить, что в ряде случаев более быстрая на простых операциях база данных существенно теряет в производительности при росте интенсивности нагрузки. В то же время программные решения, специально спроектированные для доступа из нескольких приложений, более устойчивы к конкурентным операциям.

В целом, для каждого готового программного решения в соответствии с его внутренней архитектурой существуют данные с "наиболее подходящим" набором свойств, обрабатываемые с наибольшей для данного решения эффективностью, и данные с "менее подходящим" набором свойств, обработка которых данным программным решением затруднена. В то же время вследствие значительной разнородности хранимых данных, жестких требований и высокой нагрузки современные АСУ не позволяют пренебречь характеристиками обрабатываемых данных за счет производительности того или иного программного решения.

Отсутствие универсальной системы управления базами данных с открытым кодом, ограничение на мощность и стоимость серверного оборудования приводят к тому, что при проектировании крупной АСУ, предназначенной для обработки разнородных данных, зачастую невозможно обеспечить все требования к системе одним готовым программным решением.

Возможным решением может быть повышение эффективности управления хранением данных за

счет разделения хранимых данных и использования нескольких взаимодействующих в той или иной степени доработанных программных решений. Соответственно, рассматривая хранимую информацию как набор классов данных (таблиц в реляционной модели данных), предлагается улучшить эффективность управления хранением данных за счет подбора распределения классов по базам данных. Выбор, в общем случае, нескольких программных решений позволяет совместить преимущества каждой из систем, избегая, по возможности, недостатков за счет наилучшего сопоставления части обрабатываемых данных и хранилища, отвечающего за эту часть. Неизменность разбиения при этом гарантирует постоянные характеристики системы при постоянных характеристиках нагрузки, что не достигается кэшированием.

Моделирование процесса обработки запросов СУХД

В процессе проектирования СУХД была создана математическая модель процесса обработки запросов к хранилищу (системе из нескольких хранилищ). Модель и критерии эффективности позволяют количественно оценить работу одного или нескольких хранилищ, что, в свою очередь, дает возможность оценки разбиения хранимых данных и, что не менее важно, пригодности и целесообразности использования того или иного программного решения.

Процесс обработки запросов СУХД адекватно может быть представлен с помощью математического аппарата теории массового обслуживания [10]. Так, исходя из работ [11], а также доступных экспериментальных данных запросы к хранилищу могут быть описаны как пуассоновский поток событий: согласно пуассоновскому распределению, число событий k на временном интервале t распределено с вероятностью

$$P(k, t) = \frac{(\lambda t)^k e^{-\lambda t}}{k!}, \quad (1)$$

где λ — интенсивность потока событий. При этом суперпозиция пуассоновских потоков представляет собой пуассоновский поток с суммарной интенсивностью.

Далее, для "характерных" запросов время обслуживания может быть представлено дискретной случайной величиной, т. е. вероятностью времени обслуживания b_j заявки типа j :

$$P(b_j) = \frac{\lambda_j}{\sum_k \lambda_k}, \quad (2)$$

где $\lambda_j(\lambda_k)$ — интенсивность потока заявок типа $j(k)$.

Рассматривая данную модель, в основе критерия эффективности можно использовать среднее время ожидания выполнения заявки. Для одного

хранилища критерием эффективности обработки группы классов данных может быть величина

$$W = \frac{\lambda b^2 (1 + \sigma_B^2 / M [b]^2)}{2(1 - \lambda b)}, \quad (3)$$

где $M[b]$ (далее — $b^{(1)}$) — математическое ожидание времени выполнения заявок для данного хранилища в соответствии с его нагрузкой; σ_B^2 — дисперсия времени выполнения заявок.

В качестве критерия эффективности совокупной системы из нескольких хранилищ (т. е. критерия эффективности СУХД) предлагается величина

$$J = \sum_i \sum_j (b_{ij} + w_i) x_{ij}, \quad (4)$$

где w_i — критерий эффективности хранилища типа i ; b_{ij} — время обслуживания заявки типа j в хранилище типа i (обозначим $j \rightarrow i$); x_{ij} — элемент принадлежности:

$$x_{ij} = \begin{cases} 1, & j \rightarrow i; \\ 0, & j \nrightarrow i. \end{cases} \quad (5)$$

Значение критерия эффективности СУХД при фиксированном наборе классов данных и используемых программных решений является, по сути, оценкой разбиения данных по хранилищам.

Далее, для определения оптимального, с точки зрения предложенных критериев, разбиения хранимых данных необходимо решить задачу оптимизации относительно переменных x_{ij} , в которой целевой функцией будет являться критерий J , ограничениями — выражения вида

$$\begin{cases} 1 - \lambda_1 b_1^{(1)} > 0; \\ 1 - \lambda_M b_M^{(1)} > 0, \end{cases} \quad (6)$$

где λ_i — суммарная интенсивность потока запросов для хранилища типа i ; $b_i^{(1)}$ — математическое ожидание времени выполнения заявок для хранилища типа i ; M — число хранилищ данных.

Представленные ограничения определяют отсутствие накопления очереди для каждого из хранилищ. Дополнительно могут быть добавлены любые линейные ограничения, например, для определения строгой принадлежности тех или иных классов соответствующим хранилищам.

Задача оптимизации критерия J принадлежит к классу нелинейных псевдобулевых задач [12]. При решении практической задачи для 100 классов данных и 2 хранилищ число возможных разбиений составляет величину порядка 10^{30} . Задачи псевдобулевой оптимизации часто встречаются на практике. Существует большое число подходов к их решению. Однако ввиду больших размерностей и высокой теоретической сложности вычислений не существует универсального алгоритма, дающего возможность получить точное решение за приемлемое время.

Одним из вариантов решения поставленной задачи оптимизации является частичная линеаризация

задачи, позволяющая перейти к дробно-линейной задаче оптимизации, и последующее сведение к задаче булевой выполнимости (*SAT*-задача). *SAT*-задача является широко известной NP-полной задачей теории алгоритмов [13]. Недостатком данного подхода является значительное увеличение числа переменных и ограничений: для приведенной размерности практической задачи число переменных может составить величину порядка 10^6 . В то же время, несмотря на высокую теоретическую вычислительную сложность, существует большое число программно реализованных алгоритмов, основанных, в свою очередь, на алгоритмах поиска с возвратом, позволяющих решить *SAT*-задачу за приемлемое время даже при столь значительных размерностях [14]. При этом ряд программных решений работают с линейными ограничениями задачи оптимизации, что позволяет избежать дополнительных преобразований [15].

Пример.

Рассмотрим разбиение 10 классов по двум базам данных, входными данными являются интенсивности и времена обработки запросов (табл. 1).

Таблица 1

Параметры запросов к базам данных

j	1	2	3	4	5	6	7	8	9	10
λ_j, c^{-1}	0,063	0,125	0,25	0,5	1	0,2	0,1	0,1	0,15	0,1
b_{1j}, c	5,5	1,5	1,2	0,18	0,125	0,3	0,8	0,1	0,1	0,21
b_{2j}, c	2	1	0,8	0,45	0,225	0,1	0,3	0,1	0,2	0,31

Начальное приближение $X_0 = \{x_{ij}\}$ определено в табл. 2.

Таблица 2

Начальное разбиение классов данных

$i \backslash j$	1	2	3	4	5	6	7	8	9	10
1	0	0	1	0	1	1	0	1	1	0
2	1	1	0	1	0	0	1	0	0	1

Значение критерия $J = 12,8$ с, среднее время ожидания выполнения заявок: 0,69 с и 0,66 с для первого и второго хранилища соответственно.

Оптимальное разбиение X_{opt} представлено в табл. 3.

Таблица 3

Оптимальное разбиение классов данных

$i \backslash j$	1	2	3	4	5	6	7	8	9	10
1	0	0	0	1	1	1	0	1	1	1
2	1	1	1	0	0	0	1	0	0	0

Значение критерия $J = 6,07$ с, среднее время ожидания выполнения заявок: 0,01 и 0,22 с для первого и второго хранилища соответственно.

Выбор программных решений в качестве основы для проектирования АСУ обусловлен, в первую очередь, характеристиками и предметной областью хранимых данных. Для решения практических задач наиболее целесообразно комбинировать программные решения с различной архитектурой для "наиболее широкого" покрытия характеристик обрабатываемых данных. Тем не менее, в качестве своеобразного предельного случая описанная методика может быть применена для одинаковых хранилищ. В таком случае данная конфигурация является прекрасно масштабируемой системой с постоянно разделенными классами данных. Также критерий оценки эффективности одного хранилища может быть применен при выборе подходящего программного решения безотносительно разделения данных.

Таким образом, в статье изложена методика оценки и повышения эффективности систем управления хранением данных. Данная методика, основываясь на математической модели, позволяет при ограниченных возможностях оборудования и каналов связи, тем не менее, функционально и эффективно решить задачу хранения разнородных данных при проектировании АСУ.

1. **Вигерс К.** Разработка требований к программному обеспечению. М.: Издательско-торговый дом "Русская редакция", 2004.
2. **Энсор Д., Стивенсон И.** Oracle. Проектирование баз данных. К.: Издательская группа BHV, 1999.
3. **Комплекс** программных средств повышения производительности баз данных "Mirror Database Tools". URL: www.mirrordatabase.com
4. **Owens M.** The Definitive Guide to SQLite. New York (USA): Apress publishing, 2006.
5. **Krosing H., Roybal K., Mlodgenski J.** PostgreSQL Server Programming. Birmingham (UK): Packt publishing, 2013.
6. **Soliman A.** Getting Started with Memcached. Birmingham (UK): Packt publishing, 2013.
7. **White T.** Hadoop: the Definitive Guide, 3rd Edition. Sebastopol (USA): O'Reilly Media, 2012.
8. **Распределенная система** The Google File System. URL: research.google.com
9. **Дудаков Н. С., Пирогов Н. Е., Шумилов Ю. Ю.** Кросс-платформенная система безопасного управления хранением динамических данных. М.: НИЯУ МИФИ, 2009. БИТ № 3.
10. **Клейнрок Л.** Теория массового обслуживания. М.: Машиностроение, 1979.
11. **Хинчин А. Я.** Работы по математической теории массового обслуживания. М.: Физматгиз, 1963.
12. **Boros E., Hammer P. L.** Pseudo-Boolean Optimization, RUTCOR. Piscataway (USA), 2001.
13. **Беллман Р., Дрейфус С.** Прикладные задачи динамического программирования. М.: Наука, 1965.
14. **Jarvisalo M., Le Berre D., Roussel O.** The International SAT Solver Competitions. URL: www.satcompetition.org
15. **Sheini H. M., Sakallah K. A.** Pueblo: A Hybrid Pseudo-Boolean SAT Solver // Journal on Satisfiability, Boolean Modeling and Computation. 2006.

N. S. Dudakov, Software Developer, JSC "Concern "Systemprom", e-mail: Nikolay.Dudakov@gmail.com,
Yu. Yu. Shumilov, Professor, National Research Nuclear University MEPhI, e-mail: shumilovyy@gmail.com

The Hybrid Data Management and Optimization System for the ACS

The paper presents a methodology for evaluating and improving the effectiveness of storage management during the development of automated control systems. The authors describe modern DBMSs, their architectures, advantages and disadvantages. It is shown, that performance of the open-source DBMSs is insufficient for some practical problems. The authors propose the division of stored data and the usage of several software solutions to improve the system performance. According to this proposition, each part of the data would be processed in a corresponding database. The model of the resulting system describing storage process in terms of queuing theory is also provided. According to this model, the authors propose criteria for efficient processing of requests to read and write the data. The optimization techniques that are needed to find an optimal division of stored data are also provided in this paper.

Keywords: DBMS, database architecture, queuing system, nonlinear pseudo-boolean optimization

References

1. **Wiegiers K.** Razrabotka trebovanij k programmnomu obespecheniju. M.: Izdatelsko-torgovy dom "Russkaya Redakciya", 2004.
2. **Ensor D., Stevenson I.** Oracle. Proektirovanie baz dannyh. K.: Izdatelskaya gruppa BHV, 1999.
3. **Комплекс** programnyh sredstv povysheniya proizvoditelnosti baz dannyh "Mirror Database Tools". URL: www.mirrordatabase.com
4. **Owens M.** The Definitive Guide to SQLite. New York (USA): Apress publishing, 2006.
5. **Krosing H., Roybal K., Mlodgenski J.** PostgreSQL Server Programming. Birmingham (UK): Packt publishing, 2013.
6. **Soliman A.** Getting Started with Memcached. Birmingham (UK): Packt publishing, 2013.
7. **White T.** Hadoop: the Definitive Guide, 3rd Edition. O'Reilly. Sebastopol (USA): Media, 2012.
8. **Distributed** file system, The Google File System. URL: research.google.com
9. **Dudakov N. S., Pirogov N. E., Shumilov Y. Y.** Krossplatformennaya sistema bezopasnogo upravleniya hraneniem dinamicheskikh dannyh. M.: NIYAU MIFI, 2009. BIT N 3.
10. **Kleinrock L.** Teoriya massovogo obsluzhivaniya. M.: Mashinostroenie, 1979.
11. **Hinchin A. Ya.** Raboty po matematicheskoy teorii massovogo obsluzhivaniya. M.: Fizmatgiz, 1963.
12. **Boros E., Hammer P. L.** Pseudo-Boolean Optimization. Piscataway (USA); RUTCOR, 2001.
13. **Bellman R., Dreyfus S.** Prikladnye zadachi dinamicheskogo programmirovaniya. M.: Nauka, 1965.
14. **Jarvisalo M., Le Berre D., Roussel O.** The International SAT Solver Competitions. URL: www.satcompetition.org
15. **Sheini H. M., Sakallah K. A.** Pueblo: A Hybrid Pseudo-Boolean SAT Solver. *Journal on Satisfiability, Boolean Modeling and Computation*. 2006.

В. А. Грушин, канд. техн. наук, доц., **Я. А. Архипова**, ст. преподаватель, e-mail: yand5833@yandex.ru
Нижегородский государственный лингвистический университет им. Н. А. Добролюбова

Динамическая модель краткосрочного прогнозирования социально-экономических процессов

Рассматривается метод анализа и прогнозирования социально-экономических данных на базе динамической модели в виде набора обеляющих фильтров порядка с 1 по Q включительно на основе построения динамической функции прогноза. Описан алгоритм обработки данных, его возможности и анализ эффективности применения. Приведены результаты практического применения предложенной модели при прогнозировании динамики цены акций ОАО "Газпром".

Ключевые слова: авторегрессионная модель наблюдений, обеляющий фильтр, обеляющий фильтр переменного порядка, динамическая функция прогноза, однородность данных, дерево прогнозов

Введение

Анализ и прогнозирование временного ряда данных социально-экономических процессов (СЭП), нестационарных по своей природе, по-прежнему актуальны и необходимы для принятия обоснованных и эффективных управленческих решений, а также стратегического и тактического планирования в экономической и социальной сфере деятельности людей.

В настоящее время широкое распространение для анализа параметров временных рядов СЭП получили методы регрессионного анализа: авторегрессии (АР) детерминированного малого порядка, скользящего среднего (СС) и авторегрессии — скользящего среднего (АРСС), а также некоторые их модификации. В их основе лежит предположение стационарности приращений или скорости приращений временного ряда [1]. Названные методы дают хорошие практические результаты при удачном выборе математической модели случайного процесса и ее параметров.

Применение классических методов статистического анализа оправдано лишь для относительно узкого круга прикладных задач с большими массивами однородных данных ретроспективных наблюдений [2].

Данные СЭП в принципе не являются стационарными процессами, в связи с этим требуется применение новой математической модели, основанной на однородности данных короткой выборки наблюдений на интервале прогнозирования и позволяющей отразить информационные свойства данных. Тогда задача прогнозирования динамических изменений процесса решается в рамках проверки статистических гипотез. Рассматривается гипотеза H_0 — увеличение значения данных последующего отсчета, против альтернативы H_1 — уменьшение или равенство последующего отсчета предыдущему. Решение в пользу той или иной гипотезы принимается по методу максимального правдоподобия.

Однако рассчитать вероятности гипотез для нестационарного случайного процесса довольно сложно.

В целях увеличения точности прогноза динамики временного ряда данных СЭП на основе определения вероятностей переходов вверх-вниз по короткой выборке наблюдений ставится задача получения соответствующей прогнозной модели на базе построения динамической функции прогноза (ДФП).

Метод обеляющего фильтра переменного порядка

В основу построения указанной модели был положен метод обеляющего фильтра (ОФ) определенного порядка, описанный в работе [3]. Однако выбор порядка применяемой прогнозной модели в виде обеляющего фильтра существенно влияет на значение самого прогноза. Порядок модели, в свою очередь, отражает интенсивность воздействия многомерного вектора социальных, экономических, политических, природных, технических и других факторов, действующих в совокупности на исследуемые данные на шаге прогноза. Количество этих факторов в общем случае на интервале наблюдения может постоянно меняться. Отсюда следует, что порядок математической модели должен быть переменным и соответствующим образом корректироваться.

Применяемый в качестве основы реализации метода обеляющий фильтр (рис. 1) имеет компенсационную основу, т. е. на выходе ОФ выделяется

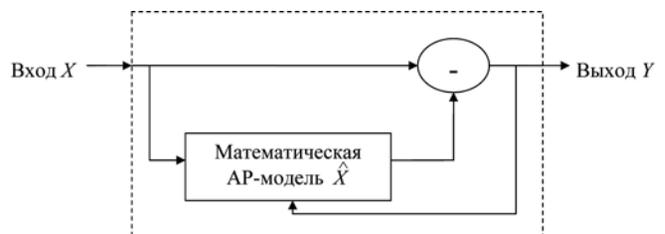


Рис. 1. Метод обеляющего фильтра

нескомпенсированный остаток Y разности между входным процессом X и его моделью \hat{X} , заданной вектором коэффициентов авторегрессии. При этом происходит адаптивная настройка коэффициентов авторегрессии по критерию минимума энтропии случайного процесса (СП) на выходе ОФ и фактическое разделение СП на детерминированную (АР-коэффициенты) и случайную (в виде "белого" шума) составляющие [4].

Выбор авторегрессионной модели наблюдений был обусловлен следующими причинами:

- АР-процесс большого порядка эквивалентен СС-процессу малого порядка [1], поэтому нет необходимости применять модель скользящего среднего в отдельности или в совокупности с АР-моделью наблюдений;
- фильтровая модель наблюдений в виде ОФ позволяет легко перейти к анализу данных в частотной области путем пересчета вектора АР-коэффициентов в оценку спектральной плотности мощности (СПМ) временного ряда [5];
- на выходе ОФ легко измерить дисперсию случайной составляющей входного процесса, которая соответствует ошибке прогноза при подаче на вход стационарного СП [1], в нашем случае — в виде первой разности отсчетов;
- при переходе к спектральным оценкам АР-процесса большое значение для последующего анализа имеют пики СПМ, дающие информацию о резонансах (доминирующих частотах) в анализируемой выборке данных.

При анализе нестационарных процессов и прогнозировании значений СП на шаге наблюдения важно правильно определить порядок математической модели (ОФ), так как с каждым последующим шагом наблюдения возможно изменение как случайной (дисперсии "белого шума" на выходе ОФ), так и детерминированной (вектора коэффициентов авторегрессии) составляющих прогноза. Для этого предлагается использовать метод ОФ переменного

порядка (ОФПП), блок-схема алгоритма которого изображена на рис. 2.

За основу метода ОФПП взята АР-модель наблюдений в классе линейных моделей в виде набора ОФ порядков q с 1 по Q , адаптивно настраиваемых по минимуму дисперсии порождающего шума на выходах ОФ. Параметры каждого ОФ пересчитываются в оценку прогноза на один шаг вперед с восстановлением постоянной составляющей данных по последнему отсчету. Это необходимо, чтобы не потерять информацию о нелинейных свойствах, содержащихся в данных в виде тренда. Оценки прогнозирования с выхода каждого канала, образованного q -м ОФ и вычислительным устройством, дают ДФП в виде оценки значения прогноза от порядка математической модели СП $F(q)$, лежащего в основе последующего анализа данных во временной области и получения собственно прогнозных значений. В данном случае речь идет о математической модели социально-экономических процессов в переменных состояниях, параметром которой служит число степеней свободы анализируемого входного процесса на шаге прогнозирования и наблюдения. При этом за основу при анализе СП выбираются ОФ с порядками, соответствующими текущему изменению данных на этом шаге.

На вход ОФ q , $q = 1...Q$ (блок 1 рис. 2) подаются данные приращений временного ряда (выборки) в виде первой разности $\Delta x_i = x_{i+1} - x_i$, где i — номер отсчета. При этом считаем ряд приращений стационарным СП с нулевым средним значением, а сами данные однородными в широком смысле.

С выхода каждого ОФ на соответствующие входы вычислительных устройств ВУ1...ВУ Q (блок 2 рис. 2) подаются векторы АР-коэффициентов и значения дисперсии нескомпенсированного остатка σ_q^2 , $q = 1...Q$, которые пересчитываются в оценки прогнозов на один шаг вперед.

Оптимальная в смысле минимума дисперсии ошибки прогнозирования оценка прогноза определяется по формуле регрессии n -го (не наблюдаемого) отсчета СП X_n относительно вектора его последовательных наблюдений [6]:

$$y_n^* = M\{x_n | x_1 \dots x_{n-1}\}, \quad (1)$$

где M — символ математического ожидания. При этом выполняются следующие операции над данными наблюдений:

а) определяются АР-модели наблюдений в классе линейных стохастических моделей типа "авторегрессия":

$$x_{iq} = a_1 x_{i-1} + \dots + a_q x_{i-q} + \eta_{iq} \quad (2)$$

где $a_1...a_q$ — векторы коэффициентов авторегрессии, $q = 1...Q$ — порядок мате-

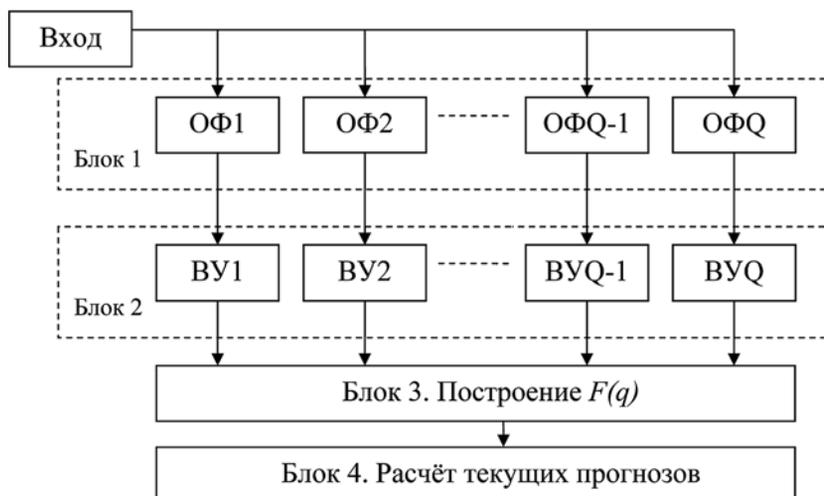


Рис. 2. Блок-схема алгоритма обеляющего фильтра переменного порядка

математической модели с порождающим "белым" шумом $\{\eta_{tq}\}$; $Q < n$, n — объем анализируемых данных, $t = 1, 2, \dots$; б) вычисляются оценки прогнозов СП X_t на один шаг вперед на выходе каждого канала обработки с использованием параметров моделей по формуле линейной оценки прогнозирования:

$$y_n = a_1 x_{n-1} + \dots + a_q x_{n-q} \quad (3)$$

Также фиксируется дисперсия ошибки прогнозирования на ОФ $_q$: $\sigma_{nq}^2 = \sigma_{\eta q}^2 = M\{\eta_t^2\}_{q}$. Результаты расчетов поступают на блок 3 (см. рис. 2) для построения ДФП $F(q)$. Расчеты проводятся в условиях нормировки математических АР-моделей каналов по уровню порождающего шума $\sigma_{nq}^2 = 1$; $q = 1, \dots, Q$. Это дает возможность корректно сравнивать результаты прогнозирования на одном графике, так как обеспечиваются одинаковые начальные условия математическим моделям ОФ разного порядка. Далее остальные действия по расчету прогнозов на один шаг вперед идут с использованием ДФП (блок 4 рис. 2).

ДФП цен закрытия акций ОАО "Газпром" на 27.03.13 представлена на рис. 3. Анализ проведен по массиву данных ежедневных приращений цен закрытия на торгах ММВБ с 16.01.12 по 26.03.13.

Исходя из принципа минимакса энтропии СП для расчета прогноза выбираем каналы, имеющие наибольшее отклонение прогнозного значения относительно последнего отсчета данных (уровень C). Это будут каналы, соответствующие уровню A — прогнозу с порядком q_A и уровню B — прогнозу с порядком q_B :

$$A = \sup F(q) = F(q_A); B = \inf F(q) = F(q_B). \quad (4)$$

Следовательно, требуется рассчитать два варианта прогнозов, соответствующих зонам AC и CB ДФП на шаге прогнозирования: прогноз вверх и прогноз вниз. Размах функции (AB) составляет зону устойчивой однородности анализируемого СП [7]. Таким образом, на шаге прогнозирования работают одновременно две математические модели авторегрессии (ОФ) порядков q_A и q_B .

Блок 4 (см. рис. 2) алгоритма ОФПП выполняет расчет прогнозных значений, который строится на предположении гауссовости распределения ошибки прогнозирования и линейности прогнозной модели. Кроме того, из двух равнозначных прогнозов — на увеличение временного ряда данных и на его уменьшение, следует выбрать наиболее правдоподобный прогноз. Выбор проводится в рамках проверки статистических гипотез. Рассматривается гипотеза H_0 : $y_t = y_t^+$ — рост значения прогнозируемого отсчета, против альтернативы H_1 : $y_t = y_t^-$ — прогнозируемое значение будет меньше последнего значения данных.

Согласно принципу максимального правдоподобия должна быть выбрана та гипотеза, вероятность наступления которой больше. Подсчет вероятностей проводим, используя построенную в блоке 3 (см. рис. 2) ДФП. Полагаем, что на шаге прогноза может быть равновероятно принята математическая модель любого порядка с 1 по Q включительно. Кроме того, необходимо ввести нормировку по порядку модели в диапазоне математического моделирования. Тогда гипотезе H_0 будет отвечать число ρ математических моделей (или каналов), дающих возрастающий прогноз, а альтернативе — $Q - \rho$ оставшихся вариантов. Следовательно, гипотеза H_0 может наступить с вероятностью

$$p^+ = \rho/Q = m,$$

а гипотеза H_1 — с вероятностью

$$p^- = 1 - \rho/Q = 1 - p^+ = 1 - m.$$

В условиях гауссового распределения прогнозов от порядка модели оценкой прогнозных значений y_t^+ и y_t^- можно считать середины значений уровней A и C и уровней C и B соответственно:

$$y_t^+ = (A + C)/2;$$

$$y_t^- = (C + B)/2.$$

Подставляя полученные значения прогнозов на 1 шаг вперед и повторяя описанные операции на последующие шаги, получаем в результате дерево прогнозов на K шагов вперед. С учетом полученных вероятностей можно выделить наиболее вероятную траекторию будущего изменения анализируемых данных (прогноз динамики временного ряда). Уровни A и B , получаемые на каждом шаге прогнозирования, определяют границы зон однородности данных. Те же уровни, построенные с помощью ОФ порядков, определенных на первом шаге прогнозирования, составят границы (зоны) устойчивой однородности продолжения входного процесса.

Анализ эффективности

Практическое применение метода ОФПП для проверки однородности данных временного ряда котировок акций при определении границ кластера [8] показало его высокую эффективность. Точность прогнозов на участках стационарности входного СП составила 0,5...1 % среднего значения ряда данных.

Граница применимости метода ОФПП определяется двумя основными факторами: возможной минимальной длиной выборки данных для анализа и максимальным значением порядка математической модели.

Первый фактор зависит от выбора математической модели используемых ОФ — как быстро ОФ могут настроиться на входной процесс. Вместо набора трансверсальных фильтров возрастающего по-

рядка, включенных в алгоритм предлагаемого метода (см. рис. 2), рациональнее воспользоваться ОФ с быстрым алгоритмом последовательной настройки решетчатой структуры, исследования адаптивных свойств и особенностей работы которого были проведены в работе [9]. Было показано, что для настройки вектора АР-коэффициентов достаточно 35...50 значений временного ряда. Соответственно, максимальный порядок математической модели должен быть, по крайней мере, на единицу меньше.

При выборе максимального порядка модели необходимо учесть два момента.

1. Возможность надежного обнаружения значимых отклонений в статистических характеристиках данных, получивших название разладки СП. Исследования в этом направлении [10] показали, что при применении ОФ решетчатой структуры выборка данных должна состоять из 200...300 значений.

2. ДФП должна содержать точки A (максимума) и B (минимума) границ зон оценок прогнозов. При анализе социально-экономических данных в большинстве случаев для этого достаточно ограничиться максимальным значением $Q = 250...350$.

Таким образом, по предварительным оценкам максимальный порядок математической модели социально-экономических процессов по методу ОФПП должен выбираться в пределах от $Q = 50$ до $Q = 300$.

Следует отметить, что предлагаемый метод имеет большую вычислительную сложность — время вычисления прогноза примерно в Q раз больше, чем в методе ОФ фиксированного порядка. Этот недостаток вполне может быть преодолен с повышением производительности вычислительной техники, а также рациональным выбором шага прогноза с учетом малой скорости и инерционности изменения социально-экономических данных.

Пример практического применения

В качестве практического эксперимента по методу ОФПП была рассчитана динамика изменения цены закрытия акций ОАО "Газпром" на три дня вперед с 27.03.2013 г. по 29.03.2013 г. Длина массива данных $n = 301$. Максимальное значение порядка модели ОФПП $Q = 300$. Полученная ДФП на первом шаге прогноза изображена на рис. 3. По ней были определены порядки математических моде-

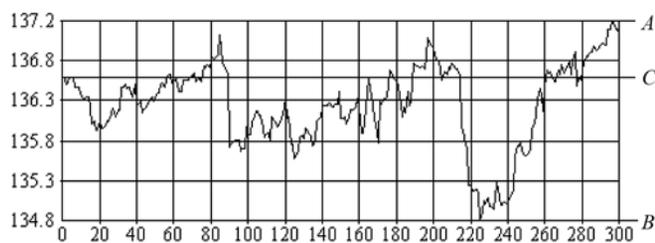


Рис. 3. Динамическая функция прогноза цен закрытия акций ОАО "Газпром" на 27.03.13

лей уровней A и B для максимальной и минимальной границ прогнозов: $q_A = 298$, $q_B = 225$. Математические модели данных порядков использовали для получения прогнозов верхней и нижней границ прогнозного коридора на три шага вперед.

Подставляя прогнозные значения y_t^+ и y_t^- для каждого шага прогноза, было получено дерево прогнозов (рис. 4, см. вторую сторону обложки) на три шага вперед и подсчитаны вероятности продолжения полученных ветвей p^+ и p^- , которые также указаны на рис. 4.

На основе дерева прогнозов было выбрано наиболее вероятное продолжение (прогноз) анализируемых данных на три шага вперед. На рис. 5 (см. вторую сторону обложки) изображены результаты прогнозирования динамики изменения временного ряда данных. Верхняя и нижняя линия (ряд 1 и ряд 3 соответственно) задают границы наиболее вероятного коридора однородности прогнозируемых значений. Ряд 2 отображает полученный прогноз цен акций ОАО "Газпром" на 27, 28 и 29 марта 2013 г., а ряд 4 (штриховая линия) отражает истинное продолжение динамики анализируемого инструмента фондового рынка России.

Заключение

Предложен и описан новый метод анализа и прогнозирования социально-экономических данных, в основу которого положен стохастический адаптивный алгоритм на базе фильтровой модели в виде набора ОФ переменного порядка $q = 1...Q$.

Коэффициенты авторегрессии каждого ОФ пересчитываются в оценки прогнозов на один шаг вперед и строится ДФП $F(q)$, параметры которой позволяют определить порядки ОФ для получения максимального и минимального прогнозов (верхней и нижней границ прогнозного коридора) для данного шага прогнозирования, вычислить прогнозные значения, построить дерево прогнозов и рассчитать вероятности реализации его ветвей на рассматриваемых шагах прогнозирования. Особенностью и несомненным достоинством метода ОФПП можно считать возможность его применения для анализа и прогнозирования данных в виде короткой выборки наблюдений нестационарного временного ряда. Анализ проводится во временной области без непосредственного расчета и перерасчета коэффициентов корреляции и с применением простых расчетных формул. Вся информация для прогнозирования при этом (статистические характеристики временного ряда) сосредоточена в ДФП — значения уровней A , B , и C , где значения уровней A и B ДФП задают границы устойчивой однородности ряда данных. Вероятности наступления дуальных прогнозов также легко подсчитываются и пересчитываются на несколько шагов вперед в наиболее вероятную траекторию развития временного ряда в рамках его однородности.

Проведен предварительный анализ эффективности предлагаемого метода с определением условий и границ его практической применимости. Применение метода ОФПП рассмотрено на примере прогнозирования динамики изменения цен акций ОАО "Газпром". Предложенный метод показал свою практическую эффективность при прогнозировании данных социально-экономических процессов, в частности, при определении однородности нестационарного временного ряда и при проведении типологической кластеризации данных социально-экономических процессов [3, 8].

Список литературы

1. **Савченко В. В.** Информационная теория колебаний биржевых котировок в динамике // Информационные технологии. 2011. № 3. С. 57–63.
2. **Лукашин Ю. П.** Адаптивные методы краткосрочного прогнозирования временных рядов: учеб. пособие. М.: Финансы и статистика, 2003. 416 с.
3. **Архипова Я. А.** Автоматическая периодизация динамики биржевых котировок на основе критерия минимума информа-

ционного рассогласования // Информационные технологии. № 6. 2012. С. 73–76.

4. **Савченко В. В.** Прогнозирование социально-экономических процессов на основе адаптивных методов спектрального оценивания // Автометрия. 1999. № 3. С. 99.

5. **Марпл С. Л.** Цифровой спектральный анализ и его приложения. М.: Мир, 1990.

6. **Программное** обеспечение биржевой игры на основе данных краткосрочного прогнозирования: учеб. пособие / Савченко В. В., Акатьев Д. Ю., Баринов А. В., Шкулев А. А. Под общ. ред. Савченко В. В. Н. Новгород: Изд. НГЛУ им. Н. А. Добролюбова. 1999. 87 с.

7. **Прохоров Ю. В., Розанов Ю. А.** Теория вероятностей. Изд. второе, перераб. М.: Наука, 1973.

8. **Архипова Я. А., Грушин В. А.** Определение границ однородности участков данных социально-экономических процессов по методу обеляющего фильтра переменного порядка // Проблемы управления в сложных экономических и социальных системах: Сборник научных статей. Вып. 2 / Под общей ред. В. А. Бородина. Н. Новгород: Изд. НГЛУ, 2011. С. 26–32.

9. **Савченко В. В., Шкулев А. А.** Адаптивная оценка прогнозирования случайного временного ряда на основе авторегрессионной модели наблюдений // Автометрия. 2002. № 2. С. 35–42.

10. **Грушин В. А.** Обнаружение и оценивание разладки случайного процесса по коротким выборкам наблюдений // Автометрия. 2000. № 6. С. 117–120.

V. A. Grushin, Associated, Professor, **Ya. A. Arkhipova**, Lecturer,

Department for Mathematics and Computer Science Nizhny Novgorod State Linguistic University

Short-Term Prediction Dynamic Model of Socio-Economic Processes

The social-economic processes analysis and forecasting technique on the basis of the dynamic model formed of whitening filters set with orders from 1 to Q and the forecast dynamical function definition, containing time series statistical characteristics, is examined. The model is based on homogeneity of the observation data short-run sample over the set forecasting interval and can show informational characteristics of the examined process. Data processing algorithm, its potentialities and application performance analysis are described. Using a filter model in the form of whitening filters set with a variable order gives a score in the final forecast accuracy and allows making a forecast tree with the probability value of its branches realizations calculation on the set forecasting step. The results of the offered model's practical application by the price movement predication of the Open Joint Stock Company Gazprom stocks are given.

Keywords: autoregressive observation model, whitening filter, whitening filter with variable order, forecast dynamical function, data homogeneity, forecast probability tree

References

1. **Savchenko V. V.** Informacionnaya teoriya kolebanij birzhevyyh kotirovok v dinamike. *Informacionnyye tehnologii*. 2011. N 3. P. 57–63.
2. **Lukashin Yu. P.** Adaptivnyye metody kratkosrochnogo prognozirovaniya vremennyh ryadov: Ucheb. posobie. M.: Financy i statistika, 2003. 416 p.
3. **Arkhipova Ya. A.** Avtomaticheskaya periodizatsiya dynamiki birzhevyyh kotirovok na osnove kriteriya minimuma informacionnogo rassoglasovaniya. *Informacionnyye tehnologii*. N 6. 2012. P. 73–76.
4. **Savchenko V. V.** Prognozirovanie socialno-ekonomicheskikh processov na osnove adaptivnyh metodov spektralnogo ocenivaniya. *Avtometriya*. 1999. N 3. С. 99.
5. **Marple S. L.** Digital Spectral Analysis with Applications, Prentice-Hall, Engelwood Cliffs, New Jersey, USA, 1987.
6. **Программное** обеспечение биржевой игры на основе данных краткосрочного прогнозирования: Учеб. пособие / Савченко В. В.,

Akatyev D. U., Barinov A. V., Shkulyov A. A. Pod obshei redakzhiej Savchenko V. V. N. Novgorod: NGLU. 1999. 87 с.

7. **Prokhorov Yu. V., Rozanov Yu. A.** Teoriya veroyatnostej. Izdanie vtoroje, pererabotannoe. M.: Nauka, 1973.

8. **Arkhipova Ya. A., Grushin V. A.** Opredeleniye granicz odnorodnosti uchastkov dannyh socialno-ekonomicheskikh processov po metodu obelyayushogo filtra peremennogo poryadka. *Problemy upravleniya v slozhnyh ekonomicheskikh i socialnyh sistemah: Sbornik nauchnyh statej*. Vol. 2. Pod obshei redakzhiej V. A. Borodina. N. Novgorod, NGLU, 2011. С. 26–32.

9. **Savchenko V. V., Shkulev A. A.** Adaptivnaya ozenka prognozirovaniya sluchajnogo vremennogo ryada na osnove avtoregreesionnoj modeli nablyudenij. *Avtometriya*. 2002. N 2. С. 35–42.

10. **Grushin V. A.** Obnaruzhenie i ocenivanie razladki sluchajnogo processa po korotkim vyborkam nabludenij. *Avtometriya*. 2000. N 6. С. 117–120.

СИСТЕМЫ АВТОМАТИЗИРОВАННОГО ПРОЕКТИРОВАНИЯ CAD-SYSTEMS

УДК 519.857

В. И. Струченков, д-р техн. наук, проф.,

Московский государственный технический университет радиотехники, электроники и автоматики
(МГТУ МИРЭА), e-mail: str1942@mail.ru

Алгоритмы нелинейного программирования для САПР железнодорожных трасс

При автоматизированном проектировании трасс новых дорог возникают задачи нелинейного программирования большой размерности. Стандартные методы решения таких задач требуют неприемлемо больших затрат машинного времени. В статье предложены алгоритмы, использующие специфические особенности проектных задач, которые позволяют преодолеть вычислительные трудности, не поступаясь математической корректностью.

Ключевые слова: трасса, план и продольный профиль, нелинейное программирование, целевая функция, градиент

Введение

Нелинейное программирование (далее НЛП) — это сравнительно новый раздел теории оптимизации, в котором рассматривается задача поиска экстремума (далее для определенности речь пойдет о минимуме) функции многих переменных на допустимой области, определяемой системой ограничений. Характерно, что вся совокупность переменных рассматривается как точка многомерного пространства, а процесс поиска решения можно представить как спуск по поверхности, соответствующей целевой функции, начиная от некоторой начальной точки, до попадания в малую окрестность точки минимума при обязательном соблюдении всех заданных ограничений [1, 2].

Применительно к проектированию трасс линейных сооружений (плана и продольного профиля) такой подход требует рассмотрения искомого проектного решения на проектируемом участке как единого целого, формализации проектной задачи путем построения адекватной модели целевой функции и системы ограничений. Этот подход принципиально отличается от попыток последовательного улучшения начального варианта на локальных участках, как это принято при использовании интерактивных методов проектирования [3, 4].

Особенность оптимизации трасс линейных сооружений состоит в том, что требуется найти не просто некоторую линию, удовлетворяющую условиям гладкости и другим ограничениям, для которой минимально значение целевой функции, но эта линия должна состоять из элементов заданного вида, на параметры которых накладываются ограничения. В качестве элементов в зависимости от типа сооружения используются отрезки прямых, круговых кри-

вых, парабол, клотоид и др. Число этих элементов, как правило, неизвестно, что существенно осложняет формализацию проектной задачи в виде задачи НЛП [5—7].

Отмеченная особенность проектирования оптимальных трасс, затрудняющая формализацию проектных задач, оказалась не единственной. Подробно этот вопрос рассмотрен в работах [6, 7], в которых показано, почему для поиска оптимальных трасс железных и автомобильных дорог как пространственных кривых пока не созданы приемлемые для практических целей математические модели и алгоритмы проектирования.

В связи с этим в качестве первого шага была решена задача проектирования продольного профиля по заданному варианту плана трассы.

Эта задача существенно проще, так как фиксируется сразу несколько составляющих строительных и эксплуатационных затрат и появляется возможность формализовать задачу в математических моделях и построить итерационный процесс совместного решения взаимосвязанных проектных задач: проектирование продольного профиля, проектирование поперечных профилей земляного полотна, проектирование водопропускных и других искусственных сооружений, распределение земляных масс и выбор способов производства работ [6, 7].

Задание вариантов плана трассы остается за проектировщиком, как и анализ компьютерных проектных решений, и принятие окончательного решения.

Задача проектирования продольного профиля имеет и самостоятельное значение, так как в ряде случаев, особенно при проектировании в обжитых районах, положение трассы в плане определяется условиями землепользования.

Математические модели целевой функции и формализация ограничений для этой задачи применительно к проектированию железных и автомобильных дорог рассмотрены в работах [6—8]. Там же показано, как и почему задачу приходится решать в несколько этапов, используя последовательно алгоритм НЛП для проектирования так называемого "цепочечного профиля" [5, 9], затем алгоритм динамического программирования [10, 11] для определения числа элементов и их примерного расположения, и снова практически тот же алгоритм НЛП для оптимизации параметров элементов [7, 12].

Профиль земли представляется в виде ломаной линии, а "цепочечный профиль" — это ломаная, удовлетворяющая всем ограничениям, кроме ограничения на минимальную длину элементов. Она имеет те же абсциссы вершин, что и профиль земли, и поэтому на первом этапе приходится рассматривать задачу НЛП большой размерности. Так, при проектировании перегона новой железной дороги число переменных до 1000 и число ограничений соответственно до 4000. Понятно, что время счета зависит от качества алгоритма и даже на современных компьютерах может достигать неприемлемых значений при использовании стандартных алгоритмов НЛП без учета специфических особенностей задачи. Повышение эффективности алгоритма НЛП особенно актуально ввиду необходимости многократного проектирования продольного профиля по каждому из рассматриваемых вариантов плана трассы из-за взаимосвязи с другими проектными задачами [6, 12].

Стандартные алгоритмы НЛП для решения задач с линейной системой ограничений разработаны для систем общего вида. Однако многие прикладные задачи имеют специфические особенности, что находит отражение в структуре систем ограничений. Использование этих особенностей при реализации методов НЛП позволяет создать эффективные математически корректные алгоритмы оптимизации.

Цель настоящей статьи состоит в том, чтобы показать, как это было сделано применительно к задаче проектирования продольного профиля новой железной дороги.

1. Постановка задачи. Математическая модель

Если обозначить профиль земли $H(s)$, а проектную линию $Z(s)$, то задача состоит в следующем (см. рисунок).

По заданной ломаной линии $H(s)$ найти другую ломаную $Z(s)$, такую, чтобы она удовлетворяла всем ограничениям, и был минимум критерия оптимальности, т. е.

$$\min \int_0^{S_0} F(Z(s), H(s), s) ds, \quad (1)$$

где S_0 — длина трассы в плане, а функция F моделирует строительные затраты на элементе длины [6, 12].

Различные модели целевой функции и соответствующие формулы для вычисления ее частных производных (градиента) рассмотрены в работах [5—7, 12].

Зная число (n) и длины элементов искомой ломаной (Δs_i), можно аналитически выразить все ограничения на $Z(s)$, если принять в качестве неизвестных z_i ($i = 1, 2, \dots, n$) ее ординаты в точках перелома. Эти ограничения делятся на три группы.

1. На уклоны элементов профиля

$$u_i \leq (z_{i+1} - z_i) / \Delta s_i \leq v_i \quad (i = 1, 2, \dots, n - 1),$$

где Δs_i — длины элементов. Это ограничение является дискретным аналогом ограничения на первую производную.

В силу малости проектных уклонов длина элемента и его проекции на ось абсцисс практически совпадают.

2. На разности уклонов смежных элементов:

$$w_i \leq (z_{i+2} - z_{i+1}) / \Delta s_{i+1} - (z_{i+1} - z_i) / \Delta s_i \leq t_i$$

Это ограничение является дискретным аналогом ограничения на угол поворота, а при фиксированных длинах элементов — на кривизну в профиле.

3. Высотные ограничения в отдельных точках

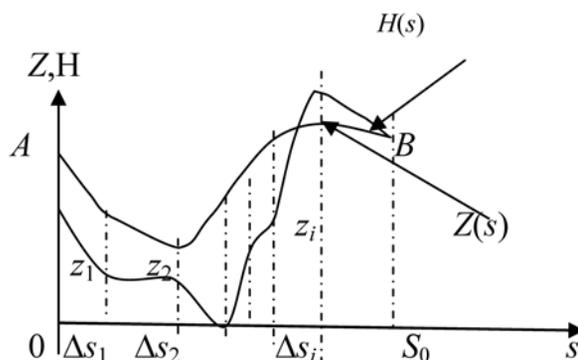
$$z_i \leq z_i^{\max} \quad \text{или} \quad z_i \geq z_i^{\min}.$$

В общем случае матрица линейных ограничений, возникающих из необходимости соблюдения технических норм, имеет вид

$$A = \begin{bmatrix} A_1 \\ -A_1 \\ A_{\Delta 1} \\ -A_{\Delta 1} \\ A_1 \\ A_2 \end{bmatrix}.$$

Система ограничений записывается в виде $Az \leq b$.

Блок A_1 соответствует ограничениям группы 1. Это двухдиагональная матрица. Все элементы первой диагонали равны минус 1, а все элементы второй диагонали равны 1.



Профиль земли $H(s)$ и "цепочечный профиль" $Z(s)$

Ограничениям второй группы соответствует блок $\mathbf{A}_{\Delta 1}$ — трехдиагональная матрица. Поскольку ограничения по уклонам и разностям уклонов двусторонние, в матрице \mathbf{A} блоки \mathbf{A}_1 и $\mathbf{A}_{\Delta 1}$ содержатся как со знаком плюс, так и со знаком минус.

Ограничениям третьей группы соответствуют блоки \mathbf{A}_1 и \mathbf{A}_2 . В каждой строке \mathbf{A}_1 все элементы нули, кроме одного, который равен минус 1 (ограничение снизу), а в каждой строке \mathbf{A}_2 — плюс 1 (ограничение сверху). В каждом столбце \mathbf{A}_1 и \mathbf{A}_2 может быть только один ненулевой элемент.

Ограничения не позволяют использовать методы локального варьирования, а взаимосвязь положения элементов проектной линии в насыпях и выемках, сооружаемых совместно, не позволяет применить динамическое программирование при поиске оптимального "цепочечного" профиля [6—8, 12]. Этим обусловлено особое внимание к методам НЛП, при использовании которых проектная линия рассматривается как единое целое.

2. Применение методов НЛП

2.1. Общая схема методов спуска

Алгоритмы нелинейного программирования при линейной системе ограничений состоят из следующих пунктов.

1. Построение допустимого начального приближения \mathbf{z}^0 .
2. Вычисление градиента целевой функции \mathbf{f}^0 .
3. Определение набора активных ограничений.
4. Построение направления спуска \mathbf{p}^0 в допустимой области.
5. Проверка условия прекращения счета и, если они не выполнены, то следующий пункт.
6. Вычисление шага λ и новой итерационной точки $\mathbf{z}^{k+1} = \mathbf{z}^k + \lambda \mathbf{p}^k$ и к п. 2.

Различные алгоритмы отличаются построением направления спуска [1, 2, 5]. При движении по вектору \mathbf{p}^k активные ограничения (т. е. выполненные как равенства) остаются активными. Если матрица \mathbf{A}^k составлена из строк исходной матрицы, соответствующих активным на данной итерации ограничениям, т. е. $\mathbf{A}^k \mathbf{z}^k = \mathbf{b}^k$ и требуется чтобы $\mathbf{A}^k \mathbf{z}^{k+1} = \mathbf{b}^k$, то очевидно, что $\mathbf{A}^k \mathbf{p}^k = 0$. Множество векторов, удовлетворяющих этому условию, называется нуль-пространством матрицы [2]. Далее индекс k (номер итерации) опускаем.

Если в качестве направления спуска принять проекцию \mathbf{p} антиградиента $-\mathbf{f}$ на нуль-пространство матрицы активных ограничений, то стандартный алгоритм требует решения систем линейных уравнений большой размерности. По формуле Розена [1, 2]

$$\mathbf{p} = -(\mathbf{E} - \mathbf{A}^T(\mathbf{A}\mathbf{A}^T)^{-1}\mathbf{A})\mathbf{f}, \quad (2)$$

где \mathbf{E} — единичная матрица, а индекс T означает транспонирование.

При большой размерности задачи для ее решения с требуемой точностью необходимо несколько тысяч итераций, что приводит к неприемлемым затратам машинного времени даже на современных персональных компьютерах. Поэтому пришлось искать возможности вычисления направления спуска без использования формулы Розена.

2.2. Использование особенностей системы ограничений

Если некоторая переменная z_j ни в одно из активных ограничений не входит, то $p_j = -f_j$. И если все ограничения неактивны, то направление спуска совпадает с антиградиентом.

Далее, если на q -й итерации матрица \mathbf{A}_q такова, что она может быть разбита на блоки вида

$$\mathbf{A}_q = \begin{bmatrix} n_1 & n_2 \\ \mathbf{A}_{q1} & 0 \\ 0 & \mathbf{A}_{q2} \end{bmatrix} \begin{bmatrix} m_1 \\ \vdots \\ m_2 \end{bmatrix},$$

то можно вычислять отдельно первые n_1 компонент вектора \mathbf{p} , используя только первые m_1 строк матрицы \mathbf{A}_q , а затем его остальные n_2 компонент, используя m_2 оставшихся строк. В этом легко убедиться, используя определение понятия проекции вектора на подпространство [2].

Будем считать, что такое разбиение на участки независимого построения проекции градиента выполнено и перейдем непосредственно к ее построению для каждого из них.

В случае отсутствия активных высотных ограничений правило построения проекции градиента зависит от комбинации активных ограничений по уклонам (строки матриц \mathbf{A}_1 или $-\mathbf{A}_1$) и ограничений по разности уклонов (строки матриц $\mathbf{A}_{\Delta 1}$ и/или $-\mathbf{A}_{\Delta 1}$).

Если матрица активных ограничений \mathbf{A}_q составлена только из строк \mathbf{A}_1 или $-\mathbf{A}_1$, т. е. имеется участок предельного уклона от некоторой точки с номером k до точки с номером $k+r$, то для вычисления компонент проекции градиента, соответствующих этому участку, нужно взять среднее арифметическое соответствующих компонент градиента

$$p_k = p_{k+1} = \dots = p_{k+r} = (f_k + f_{k+1} + \dots + f_{k+r}) / (r + 1). \quad (3)$$

Формула (3) получается следующим образом: все компоненты вектора \mathbf{p} с k -й по $(k+r)$ -ю должны быть равны между собой, так как \mathbf{p} принадлежит нуль-пространству двухдиагональной матрицы \mathbf{A}_q . Кроме того, нормаль $\mathbf{f} - \mathbf{p}$ должна быть ортогональна любому вектору из этого подпространства, например вектору, все компоненты которого равны единице. Следовательно,

$$f_k - p_k + f_{k+1} - p_{k+1} + \dots + f_{k+r} - p_{k+r} = 0,$$

что и дает формулу (3).

Геометрически равенство всех компонент вектора \mathbf{p} означает, что на рассматриваемом участке проектную линию как единое целое нужно сдвинуть по оси ординат. Формула (3) верна и в том случае, когда участок предельного уклона переходит в участок предельной кривизны, причем в общей точке этих участков разность уклонов имеет предельное значение. В этом случае только сдвиг по оси ординат сохраняет предельные значения уклонов и их разностей.

Если на участке предельной кривизны нет ни одного элемента предельного уклона, т. е. матрица \mathbf{A}_q составлена из строк матрицы $\mathbf{A}_{\Delta I}$ и/или $-\mathbf{A}_{\Delta I}$, то формула (3) дает вектор, принадлежащий нуль-пространству матрицы \mathbf{A}_q . Однако этот вектор не является проекцией градиента. Кроме сдвига существуют и другие преобразования участка предельной кривизны, которые не меняют разности уклонов смежных элементов профиля. Эти преобразования состоят в повороте участка как единого целого с центром в любой точке профиля, т. е. в увеличении (или уменьшении) уклонов всех элементов на одно и то же значение.

Если имеем участок предельной кривизны от некоторой точки с номером k до точки с номером $k+r$, то число переменных равно $r+1$, а число активных ограничений равно $r-1$. Число "степеней свободы", т. е. размерность нуль-пространства матрицы \mathbf{A}_q равна 2. Это означает, что любую вариацию проектной линии с сохранением разностей уклонов можно представить как комбинацию сдвига и поворота с центром в одной из вершин проектной линии. Другими словами, неизвестный вектор \mathbf{p} определяется двумя, а не $r+1$ параметрами. Один параметр задает сдвиг, а другой — поворот. Заметим, что можно было бы взять вместо сдвига поворот с центром в другой точке.

Обозначим параметр сдвига через a , а параметр поворота с центром в начале участка через δ . Тогда

$$p_k = a; p_{k+1} = a + \delta s_k; p_{k+2} = a + \delta(s_k + s_{k+1}); \\ \dots p_{k+r} = a + \delta(s_k + s_{k+1} + \dots + s_{k+r-1}).$$

Здесь, как и ранее, s_k — длина элемента от точки $k-1$ до k .

Такую же структуру имеет и произвольный вектор, принадлежащий нуль-пространству, на которое мы проектируем градиент \mathbf{f} . Этому произвольному вектору \mathbf{v} должен быть ортогонален вектор $\mathbf{f}-\mathbf{p}$ (нормаль).

Для определения неизвестных параметров a и δ используем ортогональности воспользуемся дважды.

1. Берем вектор \mathbf{v} , все компоненты которого равны 1. Его параметр $a = 1$, а параметр $\delta = 0$.

Условие $(\mathbf{f}-\mathbf{p}, \mathbf{v}) = 0$ дает

$$a(r+1) + \delta(rs_k + (r-1)s_{k+1} + \dots + s_{k+r-1}) = \\ = f_k + f_{k+1} + \dots + f_{k+r}.$$

2. Берем вектор \mathbf{v} , для которого параметр $a = 0$, а $\delta = 1$.

Условие $(\mathbf{f}-\mathbf{p}, \mathbf{v}) = 0$ дает

$$a(rs_k + (r-1)s_{k+1} + \dots + s_{k+r-1}) + \\ + \delta(s_k^2 + (s_k + s_{k+1})^2 + \dots + (s_k + s_{k+1} + \dots + s_{k+r-1})^2) = \\ = f_{k+1}s_k + f_{k+2}(s_k + s_{k+1}) + \dots + \\ + f_{k+r}(s_k + s_{k+1} + \dots + s_{k+r-1}).$$

Получили систему из двух линейных уравнений с двумя неизвестными a и δ . Матрица системы симметрическая и невырожденная. Система имеет единственное решение, что позволяет однозначно определить неизвестные a и δ , а затем все компоненты вектора \mathbf{p} .

Если на правом конце участка поворота разность уклонов не равна предельной, а затем вновь следует участок предельной кривизны (участок поворота), то необходимо уже не два, а три параметра для задания структуры вектора \mathbf{p} . Соответственно получается система трех уравнений с тремя неизвестными.

В общем случае, но при отсутствии активных высотных ограничений, участку, для которого компоненты вектора \mathbf{p} определяются независимо от остальных, может соответствовать произвольная комбинация более мелких участков сдвигов и поворотов. Недопустимы только два участка сдвига подряд, так как в данном случае это фактически один участок сдвига. Новый участок сдвига появляется, если в конце предыдущего участка разность уклонов не предельная, а далее все уклоны предельные. В этом случае новый параметр не нужен, так как на участке сдвига все компоненты вектора \mathbf{p} равны его компоненте, соответствующей концу предыдущего участка.

Если в двух соседних точках разности уклонов не предельные, то эти точки не могут принадлежать одному участку сдвига или поворота, но могут быть соответственно концом одного и началом другого участка независимого построения проекции градиента.

Размерность нуль-пространства матрицы активных ограничений на единицу больше числа участков поворота (в отсутствие активных высотных ограничений).

Если на участке сдвига обнаруживается активное высотное ограничение или два таких ограничения на участке поворота, то очевидно, что все соответствующие компоненты вектора \mathbf{p} равны нулю. Если на таких участках есть еще активные высотные ограничения, то это вырожденный случай. При его обнаружении в "лишних" высотных ограничениях изменяется правая часть неравенства на малую величину, например 0,0001 м, что практически не влияет на результат оптимизации, но позволяет преодолеть вырожденность.

Фактически определение структуры вектора проекции \mathbf{p} , как и любого вектора из нуль-простран-

ства матрицы активных ограничений, означает разложение этого вектора по базису нуль-пространства. А искомые параметры есть координаты неизвестного вектора в этом базисе. Участки предельного уклона и предельной кривизны, рассматриваемые совместно, содержат не по одному элементу, поэтому размерность нуль-пространства существенно меньше размерности (числа компонент) вектора \mathbf{p} . Поэтому и размерность систем линейных уравнений, которые приходится решать для определения структурных параметров, меньше числа активных ограничений на соответствующем участке. При использовании формулы Розена пришлось бы решать систему с матрицей $\mathbf{A}_g \mathbf{A}_g^T$, размерность которой равна числу активных ограничений.

Пусть базисные векторы нуль-пространства матрицы \mathbf{A}_g являются столбцами некоторой матрицы \mathbf{C} и координаты неизвестного вектора \mathbf{p} в этом базисе образуют вектор \mathbf{v} . Тогда $\mathbf{p} = \mathbf{Cv}$. Произвольный вектор \mathbf{s} в том же нуль-пространстве представляется в виде $\mathbf{s} = \mathbf{Cw}$, где \mathbf{w} — соответственно произвольный вектор его координат в том же базисе. Условие ортогональности $(\mathbf{f} - \mathbf{p}, \mathbf{s}) = 0$ позволяет построить систему уравнений для определения векторов \mathbf{v} , а затем и \mathbf{p} : $(\mathbf{f} - \mathbf{Cv}, \mathbf{Cw}) = 0$; далее $(\mathbf{C}^T \mathbf{f} - \mathbf{C}^T \mathbf{Cv}, \mathbf{w}) = 0$. И в силу произвольности вектора \mathbf{w} $\mathbf{C}^T \mathbf{f} - \mathbf{C}^T \mathbf{Cv} = 0$. Отсюда $\mathbf{v} = (\mathbf{C}^T \mathbf{C})^{-1} \mathbf{C}^T \mathbf{f}$, и для искомого вектора \mathbf{p} получаем

$$\mathbf{p} = \mathbf{Cv} = \mathbf{C}(\mathbf{C}^T \mathbf{C})^{-1} \mathbf{C}^T \mathbf{f}. \quad (4)$$

Таким образом, для вычисления проекции градиента приходится решать систему линейных уравнений, размерность которой равна числу базисных векторов.

Базисный вектор для участка сдвига имеет равные компоненты.

Для участка поворота в качестве базисных векторов можно взять векторы, соответствующие повороту проектной линии с центром в начальной и конечной точках соответствующего участка. При этом все уклоны получают равные приращения, а их разности не изменятся. Задав это приращение уклонов δ (например $\delta = 1$), последовательно вычисляем компоненты базисного вектора \mathbf{c} :

$$c_1 = 0; c_2 = \delta s_1; c_3 = \delta(s_1 + s_2); \dots; c_r = \delta(s_1 + s_2 + \dots + s_{r-1}).$$

Здесь нумерация условная, реально первый номер соответствует началу участка предельной кривизны. Аналогично выражаем через единственный параметр компоненты второго вектора, соответствующего повороту с центром в конце участка.

Как уже отмечалось, возможны сложные комбинации активных ограничений. Однако в любом случае базисные векторы можно построить, анализируя участки сдвига (активные ограничения по уклонам) или поворота (активные ограничения по разности уклонов). Наличие активного высотного

ограничения в некоторой точке на участке поворота оставляет только один базисный вектор, соответствующий повороту с центром в этой точке. Наличие двух таких точек фиксирует весь участок, равно как наличие одной такой точки на участке сдвига.

В использовавшейся ранее программе для вычисления проекции антиградиента на каждой итерации вместо систем с матрицей \mathbf{AA}^T решались системы линейных уравнений малой размерности, получаемые описанным способом.

Однако наличие структурных особенностей системы ограничений позволяет при любой комбинации активных ограничений построить базис в нуль-пространстве матрицы активных ограничений и определять направление спуска вообще без решения каких-либо систем линейных уравнений. На этом основан новый алгоритм.

Итак, имеем вектор начального приближения или текущую итерационную точку \mathbf{z}^0 в n -мерном евклидовом пространстве \mathbf{R}_n . Обозначим через \mathbf{M} нуль-пространство матрицы \mathbf{A} активных ограничений в точке \mathbf{z}^0 (индекс q — номер итерации, далее для простоты опускаем). Число строк матрицы \mathbf{A} обозначаем через m . Тогда размерность нуль-пространства матрицы \mathbf{A} равна $n - m$.

Предположим, что мы знаем базис в \mathbf{M} (базисную матрицу \mathbf{C}) и в дополнении к \mathbf{M} (базисную матрицу \mathbf{B}). В матрице \mathbf{C} $n - m$ столбцов, а в матрице \mathbf{B} соответственно m столбцов. Так, при $n = 3$ и $m = 1$ нуль-пространство — это плоскость, проходящая через нуль, и любые два вектора в ней, не направленные по одной прямой, образуют базис. Соответственно, дополнение — это прямая, проходящая через нуль и не лежащая в этой плоскости, в частности, это может быть прямая, ортогональная плоскости, тогда это ортогональное дополнение. Базис в дополнении — вектор на этой прямой.

При $n = 3$ и $m = 2$ нуль-пространство — это прямая, проходящая через нуль, а дополнение — это плоскость, проходящая через нуль и не содержащая эту прямую, ортогональность прямой и плоскости не обязательна. Соответственно, дополнение не единственно. Если обозначить дополнение к \mathbf{M} через \mathbf{M}_\perp , то $\mathbf{M} \cup \mathbf{M}_\perp = \mathbf{R}_n$. Базис в \mathbf{M} плюс базис в \mathbf{M}_\perp образуют полный базис в \mathbf{R}_n .

Перейдем к новому базису, составленному из столбцов матриц \mathbf{C} и \mathbf{B} , т. е. проведем замену переменных: вместо старых переменных (координат в исходном базисе) будем рассматривать координаты в новом базисе. Для произвольного вектора \mathbf{z} при такой замене имеем

$$\mathbf{z} - \mathbf{z}^0 = \mathbf{C}\mathbf{y} + \mathbf{B}\mathbf{x}. \quad (5)$$

Вектор $\mathbf{C}\mathbf{y}$ принадлежит \mathbf{M} , \mathbf{y} — вектор его координат в базисе \mathbf{C} . Вектор $\mathbf{B}\mathbf{x}$ принадлежит дополнению к \mathbf{M} , и вектор \mathbf{x} — его координаты в базисе \mathbf{B} . Вектор \mathbf{y} содержит $n - m$ компонент, а вектор \mathbf{x} имеет m компонент.

Произведенная замена может быть записана следующим образом:

$$\mathbf{z} - \mathbf{z}^0 = (\mathbf{C}|\mathbf{B}) \begin{pmatrix} \mathbf{y} \\ \mathbf{x} \end{pmatrix} = \mathbf{D}\mathbf{u}.$$

Матрица \mathbf{D} содержит $n-m$ столбцов матрицы \mathbf{C} и m столбцов матрицы \mathbf{B} .

Вектор \mathbf{u} содержит $n-m$ компонент вектора \mathbf{y} и m компонент вектора \mathbf{x} .

При $\mathbf{z} = \mathbf{z}^0$ имеем $\mathbf{x} = 0$ и $\mathbf{y} = 0$. Если $\mathbf{z} - \mathbf{z}^0$ принадлежит \mathbf{M} , то $\mathbf{x} = 0$. Поэтому поиск направления спуска в \mathbf{M} есть поиск вектора \mathbf{y} .

Обозначим целевую функцию, получаемую из формулы (1) при дискретном представлении искомой экстремали, через $K(\mathbf{z})$, а ее градиент через \mathbf{f} . Функцию, полученную при замене (5), обозначим $\Phi(\mathbf{x}, \mathbf{y})$. Первые $n-m$ компонент градиента функции $\Phi(\mathbf{x}, \mathbf{y})$ обозначим \mathbf{F}^y , а остальные m компонент — \mathbf{F}^x . Тогда преобразование градиента записывается в виде $\mathbf{F}^y = \mathbf{C}^T \mathbf{f}$ и $\mathbf{F}^x = \mathbf{B}^T \mathbf{f}$, что следует из формулы (4) и правила дифференцирования сложных функций. Это означает, что вычислив градиент в исходных координатах, его можно пересчитать в новых переменных, если известны базисные матрицы \mathbf{C} и \mathbf{B} . Если бы было известно, что в точке минимума активный набор тот же, что и в текущей точке \mathbf{z}^0 , то осталось бы решить задачу безусловной минимизации размерностью $n-m$ с начальным приближением $\mathbf{y} = 0$ и направлением спуска минус \mathbf{F}^y ($n-m$) компонент и $\mathbf{F}^x = 0$ (остальные m компонент). Другими словами, направление спуска выбиралось бы в одном и том же нуль-пространстве, а итерационная точка перемещалась бы в одном и том же граничном линейном многообразии. Однако есть еще и неактивные ограничения, которые при достаточном значении шага по вычисленному направлению могут стать активными. К тому же некоторые ограничения из активного набора должны быть исключены, так как в точке минимума они не обязательно активны. Поэтому приходится рассматривать направление в исходном пространстве, т. е. приведенный градиент $\mathbf{g} = \mathbf{C}\mathbf{F}^y + \mathbf{B}\mathbf{F}^x = \mathbf{C}\mathbf{C}^T \mathbf{f}$. Это направление не совпадает с проекцией градиента в исходном пространстве, если базис (столбцы матрицы \mathbf{C}) не ортонормальный. Действительно, как установлено ранее (см. формулу (4)), проекция градиента $\mathbf{p} = \mathbf{C}(\mathbf{C}^T \mathbf{C})^{-1} \mathbf{C}^T \mathbf{f}$. При ортонормальном базисе $\mathbf{C}^T \mathbf{C} = \mathbf{E}$ и $\mathbf{p} = \mathbf{C}\mathbf{C}^T \mathbf{f} = \mathbf{g}$.

Вектор \mathbf{g} при любом базисе в \mathbf{M} задает направление спуска в исходном пространстве, так как соответствующий ему вектор, у которого первые $n-m$ компонент образуют $-\mathbf{F}^y$, а следующие m компонент нулевые — это направление спуска в новом пространстве.

Базисная матрица \mathbf{C} остается неизменной до тех пор, пока не изменится набор активных ограничений. Заметим, что для поиска направления спуска не использовалась базисная матрица \mathbf{B} в дополнении.

2.3. Модификация набора активных ограничений

Рассмотрим вопрос об исключении ограничений из активного набора. Этот вопрос с необходимостью возникает при отсутствии направления спуска, например при попадании в вершину многогранника ограничений, что соответствует нуль-пространству матрицы активного набора, размерность которого равна нулю. Но эта вершина не обязана быть точкой минимума. Этот же вопрос о модификации активного набора может решаться и при наличии направления спуска в целях скорейшего избавления от "чужих" ограничений в активном наборе и выхода на нужное граничное многообразие для завершения процесса. Именно отсутствие возможности исключения ограничений из активного набора при нулевой проекции градиента на нуль-пространство матрицы, соответствующей этому набору, является условием достижения точки минимума.

В новых переменных система ограничений $\mathbf{A}\mathbf{z} \leq \mathbf{b}$ принимает вид

$$\mathbf{A}(\mathbf{C}\mathbf{y} + \mathbf{B}\mathbf{x}) + \mathbf{A}\mathbf{z}^0 \leq \mathbf{b}$$

и далее $\mathbf{A}\mathbf{B}\mathbf{x} \leq \mathbf{0}$, так как $\mathbf{A}\mathbf{z}^0 = \mathbf{b}$ и $\mathbf{A}\mathbf{C}\mathbf{y} = 0$. Здесь \mathbf{A} — по-прежнему матрица, соответствующая активному набору, а \mathbf{b} — соответствующий вектор правых частей активных неравенств. Поэтому $\mathbf{A}\mathbf{z}^0 = \mathbf{b}$, $\mathbf{A}\mathbf{C}\mathbf{y} = 0$, потому что $\mathbf{C}\mathbf{y}$ — вектор из нуль-пространства матрицы \mathbf{A} . Поскольку \mathbf{y} — произвольный вектор, то $\mathbf{A}\mathbf{C} = 0$.

Система ограничений $\mathbf{A}\mathbf{B}\mathbf{x} \leq 0$ резко упрощается, если в качестве базисной выбрать матрицу \mathbf{B} такую, что $\mathbf{A}\mathbf{B}$ — диагональная матрица. Тогда каждое неравенство содержит только одну переменную. Условие диагональности матрицы $\mathbf{A}\mathbf{B}$ означает, что в новом базисе, дополнительно к столбцам матрицы \mathbf{C} должны быть векторы (столбцы матрицы \mathbf{B}), каждый из которых имеет следующие свойства: при подстановке координат i -го вектора (в исходном базисе) в j -е ограничение при $i \neq j$ получаем нуль, а при $i = j$ — отличное от нуля число, т. е. соответствующий диагональный элемент матрицы $\mathbf{A}\mathbf{B}$. Это становится очевидным, если учесть, что произведение строки матрицы \mathbf{A} на любой вектор — это то же самое, что подстановка координат вектора в соответствующее неравенство. Каждый столбец матрицы \mathbf{B} задает такое смещение, что все ограничения, которые выполнялись как равенства, по-прежнему будут выполняться как равенства, кроме одного, соответствующего номеру столбца. Это ограничение будет нарушено, если соответствующий диагональный элемент матрицы $\mathbf{A}\mathbf{B}$ положителен, или перестает быть активным, если этот элемент отрицателен. Для определенности можно выбирать столбцы матрицы \mathbf{B} так, чтобы все диагональные элементы матрицы $\mathbf{A}\mathbf{B}$ были положительны. Далее $\mathbf{A}\mathbf{B}$ обозначается через \mathbf{D} , а ее диагональные эле-

менты через d_{ij} . Заметим, что ни один диагональный элемент не может быть равным нулю, так как иначе соответствующий ему столбец матрицы \mathbf{B} принадлежит нуль-пространству матрицы \mathbf{A} , а не дополнению к нему. Знак диагонального элемента легко изменить на противоположный, изменив знаки всех компонент соответствующего столбца матрицы \mathbf{B} .

При таком выборе базисных векторов в дополнении легко убедиться в их линейной независимости. Действительно, предположим противное. Тогда один базисный вектор \mathbf{V}_i является линейной комбинацией остальных. Но все остальные базисные векторы при подстановке в i -е неравенство дают нуль, значит и их линейная комбинация дает нуль, а вектор \mathbf{V}_i при подстановке в это же неравенство дает число, отличное от нуля. Полученное противоречие доказывает линейную независимость базисных векторов.

Если сохранять активный набор, т. е. не исключать ограничения, то должно быть $\mathbf{x} = 0$ и в новой итерационной точке. Мы хотим выяснить, можно ли исключить из активного набора некоторое ограничение с номером j , т. е. исключить из матрицы активных ограничений j -ю строку и тем самым увеличить размерность ее нуль-пространства на единицу. Все остальные ограничения остаются в активном наборе, что означает $x_i = 0$ при $i \neq j$, а x_j в новой точке будет меньше 0, чтобы j -е ограничение выполнялось, но не было активным. Мы имели направление спуска при старом активном наборе $-\mathbf{F}^y$ (первые $n - m$ компонент и m нулевых компонент). Исключение j -го ограничения из активного набора означает, что помимо неизменившихся первых $n - m$ координат у нового направления появится еще одна не нулевая координата, равная $-\mathbf{F}_j^x$. Все компоненты нового направления, кроме $-\mathbf{F}_j^x$ не меняют x_j . Чтобы ограничение $d_{jj}x_j \leq 0$ не нарушалось при новом направлении необходимо и достаточно $\mathbf{F}_j^x > 0$. Очевидно, что новое направление будет направлением спуска. Чтобы в этом убедиться, достаточно вычислить его скалярное произведение на антиградиент. Оно будет положительным:

$$(\mathbf{F}^y, \mathbf{F}^y) + (\mathbf{F}_j^x)^2 > 0.$$

Особо важно отметить, что в силу простой структуры системы ограничений в новом базисе, можно одновременно исключить из активного набора все ограничения, которым соответствуют положительные компоненты вектора \mathbf{F}^x . При этом исключение каждого ограничения соответствует присоединению к базисной матрице \mathbf{C} соответствующего базисного вектора матрицы \mathbf{B} . Новый полный базис при этом не изменяется. Направление спуска после исключения ограничений противоположно вектору, у которого первые $n - m$ компонент совпадают с \mathbf{F}^y , а остальные m компонент равны компонентам вектора \mathbf{F}^x , если соответствующее огра-

ничение исключается из активного набора, и нулю в противном случае.

Определив направление спуска в новом базисе, необходимо перейти к исходным переменным (старому базису), чтобы искать значение шага по направлению спуска, не нарушая неактивные ограничения, которые пока никакого значения не имели.

В исходных переменных направление спуска

$$\mathbf{p} = -(\mathbf{C}\mathbf{F}^y + \mathbf{B}\mathbf{F}_n^x),$$

где \mathbf{F}_n^x — содержит компоненты вектора \mathbf{F}^x , соответствующие исключаемым ограничениям, остальные его компоненты равны нулю. Поиск шага по направлению целесообразно осуществлять из условия минимума целевой функции по данному направлению, если при меньшем шаге не станет активным хотя бы одно новое ограничение.

Характерно, что если рассматривается вопрос о возможности исключения из активного набора только одного ограничения, и когда такое ограничение найдено, другие ограничения не анализируются, то не нужны и другие базисные векторы. Действительно, произведение матрицы на вектор — это линейная комбинация ее столбцов, коэффициенты которой есть соответствующие координаты вектора. Поэтому $\mathbf{B}\mathbf{F}_n^x = (\mathbf{F}^x)_j \mathbf{b}^j$, так как все прочие координаты \mathbf{F}_n^x , кроме j -й равны нулю. Здесь j — номер исключаемого ограничения, а \mathbf{b}^j — соответствующий базисный вектор. Фактически можно считать, что в матрицу \mathbf{C} добавлен столбец \mathbf{b}^j , а к вектору \mathbf{F}^y приписана еще одна координата, а именно $(\mathbf{F}^x)_j$.

Таким образом, умение строить базис в нуль-пространстве $\mathbf{M}\{\mathbf{z}/\mathbf{A}\mathbf{z} = 0\}$ (матрица \mathbf{C}) и в его дополнении (матрица \mathbf{B}), удовлетворяющий условиям $\mathbf{A}\mathbf{C} = 0$ и $\mathbf{A}\mathbf{B} = \mathbf{D}$, где \mathbf{D} — диагональная матрица, позволяет обойтись без решения систем линейных уравнений, как при поиске направления спуска, так и при исключении ограничений из активного набора.

Это дает существенные упрощения и позволяет решать задачи большой размерности в приемлемое время.

Заключение

Эффективность применения методов нелинейного программирования для проектирования продольного профиля железных дорог была установлена еще при проектировании БАМ на ЭВМ БЭСМ-4 [5]. В силу ограниченных возможностей этой ЭВМ проектировались короткие участки (до 5 км) и были использованы упрощенные модели целевой функции. Новые алгоритмы НЛП требуют больше памяти, но как показали расчеты, они позволяют проектировать продольный профиль на железнодорожном перегоне (до 30 км) на общедоступных ПЭВМ в приемлемое время [6, 7, 12].

Рассмотренные алгоритмы можно использовать и при решении других проектных задач. Так, про-

ектирование продольного профиля автодорог при использовании как круговых кривых, так и парабол сводится к поиску оптимального положения вершин углов поворота вертикальных кривых [6, 8], что приводит к рассмотренной выше системе ограничений. Более того, при изменении не только проектных отметок, но и длин элементов профиля, получаемых при преобразовании "цепочки" по алгоритму динамического программирования [10, 11], соответствующая система ограничений позволяет использовать изложенные приемы перехода к новому базису в итерационном процессе. Этот круг вопросов требует отдельного рассмотрения.

Список литературы

1. Гилл Ф., Мюррей У., Райт М. Практическая оптимизация: пер. с англ. М.: Мир, 1985.
2. Аоки М. Введение в методы оптимизации: пер. с англ. М.: Наука, 1977.
3. **CARD/1**. URL: <http://www.card-1.com/en/home/>
4. **Topomatic** Robur. URL: <http://www.topomatic.ru>
5. **Использование** математических методов оптимизации и ЭВМ при проектировании продольного профиля железных до-

рог. Труды Всесоюзного НИИ транспортного строительства. Вып. 101. М.: Транспорт, 1977.

6. **Струченков В. И.** Математические модели и методы оптимизации в системах проектирования трасс новых железных дорог // Информационные технологии. 2013. № 7. С. 10—17.

7. **Struchenkov V. I.** Mathematical Models and Optimization in Line Structure Routing: Survey and Advanced Results // International Journal Communication, Network and System Sciences. Special Issue: Models and Algorithms for Application. 2012. V. 5.

8. **Струченков В. И.** Математические модели и методы оптимизации в системах проектирования трасс автомобильных дорог // Информационные технологии. 2013. № 8. С. 2—8.

9. **Михалевич В. С., Шор Н. З.** Математические основы решения задачи выбора оптимального очертания продольного профиля. Труды Всесоюзного НИИ транспортного строительства. 1964. Вып. 51.

10. **Struchenkov V. I.** Piecewise Linear Approximation of Plane Curves with Restrictions in Computer-Aided Design of Railway Routes // World Journal of Computer Application & Technology. 2014. Vol. 2, is. 1.

11. **Struchenkov V. I.** Piecewise Parabolic Approximation of Plane Curves with Restrictions in Computer-Aided Design of Road Routes // Transactions on Machine Learning and Artificial Intelligence. 2013. Vol. 1, is. 1.

12. **Струченков В. И.** О различных подходах к автоматизации проектирования трасс линейных сооружений // Информационные технологии. 2013. № 12. С. 41—47.

V. I. Struchenkov, Professor,

Moscow State University of Radio Engineering, Electronics and Automation, e-mail: str1942@mail.ru

Nonlinear Programming Algorithms for CAD of Railways Routes

Nonlinear programming problems of large dimension arise with computer-aided design of new roads. Standard methods for solving such problems require unacceptably large computing time. The article proposes algorithms that use specific features of design tasks, which overcome the computational difficulties without sacrificing mathematical correctness.

These features include a system of restrictions on the desired design solution. So when designing the optimal longitudinal profile for a given variant route's plan the system of restrictions is linear and consists of blocks of three types, each of which contains only one, respectively, two and three non-zero elements in the corresponding row of the constraint matrix.

This feature allow for any set of active restrictions at each iteration of the standard process of nonlinear programming to construct a basis in the null space of the matrix of active restrictions and find the descent direction, i.e. reduced antigradient, without solutions auxiliary systems of linear equations. The article shows how to build a basis and modify active set, if the minimum is not reached, and a descent direction is absent. The algorithm is implemented in CAD Railway.

Keywords: route, horizontal and vertical alignment, nonlinear programming, objective function, gradient, system of restrictions

References

1. Gill F., Mjurrej U., Rajt M. Prakticheskaja optimizacija: per. s angl. M.: Mir, 1985.
2. Aoki M. Vvedenie v metody optimizacii: per. s angl. M.: Nauka, 1977.
3. **CARD/1**. URL: <http://www.card-1.com/en/home/>
4. **Topomatic** Robur. URL: <http://www.topomatic.ru>
5. **Ispol'zovanie** matematicheskikh metodov optimizacii i JeVM pri proektirovanii prodol'nogo profilja zheleznyh dorog. *Trudy Vsesojuznogo NII transportnogo stroitel'stva*. V. 101. M.: Transport, 1977.
6. **Struchenkov V. I.** Matematicheskie modeli i metody optimizacii v sistemah proektirovanii trass novyh zh. d. *Informacionnye tehnologii*. 2013, N. 7. P. 10—17.
7. **Struchenkov V. I.** Mathematical Models and Optimization in Line Structure Routing: Survey and Advanced Results. *International Journal Communication, Network and System Sciences. Special Issue: Models and Algorithms for Application*. 2012. N. 5.

8. **Struchenkov V. I.** Matematicheskie modeli i metody optimizacii v sistemah proektirovanija trass avtomobil'nyh dorog. *Informacionnye tehnologii*. 2013. N. 8. P. 2—8.

9. **Mihalevich V. S., Shor N. Z.** Matematicheskie osnovy reshenija zadachi vybora optimal'nogo ochertanija prodol'nogo profilja. *Trudy Vsesojuznogo NII transportnogo stroitel'stva*. 1964. V. 51.

10. **Struchenkov V. I.** Piecewise Linear Approximation of Plane Curves with Restrictions in Computer-Aided Design of Railway Routes. *World Journal of Computer Application & Technology*. 2014. V. 2, N. 1.

11. **Struchenkov V. I.** Piecewise Parabolic Approximation of Plane Curves with Restrictions in Computer-Aided Design of Road Routes. *Transactions on Machine Learning and Artificial Intelligence*. 2013. V. 1, N. 1.

12. **Struchenkov V. I.** O razlichnih podhodah k avtomatizacii proektirovanija trass lineynyh sooruzhenij. *Informacionnye tehnologii*. 2013. N. 12. P. 41—47.

Д. В. Салеев, аспирант, e-mail: saleevd@gmail.com, А. П. Преображенский, канд. физ.-мат. наук, доц., Воронежский институт высоких технологий

Анализ особенностей САПР для ПЛИС

Проводится анализ существующих систем автоматизированного проектирования для ПЛИС. Разработаны обобщенная структурная и функциональная схемы САПР для расширения возможностей существующих систем. Предложено введение дополнительного программного блока для оценки влияния технологических факторов на работоспособность с учетом требований заказчика к разрабатываемой ИС.

Ключевые слова: ПЛИС, САПР, проектирование микросхем

Опыт высокоразвитых стран доказывает, что микро- и нанoeлектроника в настоящее время является одной из ведущих отраслей промышленности и во многом определяет темпы развития страны и состояние ее экономики в целом [1].

Техническое исполнение современных микроэлектронных устройств, в том числе различных микросхем, достаточно сложное, поэтому эффективная их разработка невозможна без применения специального программного обеспечения — систем автоматического проектирования (САПР).

Одним из основных направлений в совершенствовании ИС является разработка программируемых логических интегральных схем (ПЛИС). Главным преимуществом ПЛИС перед другими семействами интегральных схем (СИС, БИС, СБИС) является свойство реконфигурируемости: на одной и той же интегральной схеме возможно выполнять различные устройства. Также требуется меньше времени для изготовления схемы. При этом стоимость ПЛИС многократно меньше, чем стоимость заказной интегральной схемы, а применение ПЛИС позволяет уменьшить конечные габаритные размеры аппаратуры в сравнении с аналогичными устройствами, реализованными на БИС [2—4].

Основными HDL-языками, используемыми в современных САПР при функционально-логическом проектировании (начиная с описания алгоритмов и заканчивая конечной логической схемой) являются языки VHDL и Verilog [5, 6].

Цель данной работы — анализ современных САПР для проектирования ПЛИС и выработка предложений по их модернизации.

Рассмотрим основные характеристики наиболее популярных САПР.

1. САПР фирмы *Altera*

Компания Altera, головной офис которой расположен в г. Сан-Хосе (Калифорния, США), — один из ведущих мировых производителей ПЛИС, специализированных интегральных схем для конкрет-

ных устройств — ASIC, "систем на кристалле" (SoC) [7].

1.1. MAX + PLUS II. Название системы MAX + PLUS является аббревиатурой от Multiple Array Matrix Programmable Logic User System (пользовательская система программирования логики упорядоченных структур). Пакеты программы позволяют работать не только на платформе IBM PC, но также на платформах IBM .RISC и HP9000. Системные требования программного пакета невелики (системе потребуется не более 500 Мбайт свободного места на жестком диске, для разработки достаточно больших кристаллов на ПЛИС будет вполне достаточно 128 Мбайт оперативной памяти и процессор Pentium II), что позволяет работать на всех современных компьютерах. MAX + PLUS II поддерживает работу с операционной системой Windows XP, однако достаточных сведений о совместимости с операционными системами Windows Vista и Windows 7 нет [8—10].

MAX + PLUS II — среда для разработки цифровых устройств на базе ПЛИС фирмы *Altera* семейств Classic, ACEX1K, MAX (MAX5000, MAX7000, MAX9000) и FLEX (FLEX 6000, FLEX8000, FLEX10K).

Возможности MAX + PLUS II весьма широки: система позволяет реализовать все стадии, необходимые для выпуска готовых изделий.

1. Создание проекта устройства — с использованием графического "Graphic Editor", текстового "Text Editor" и символьного "Symbol Editor" редакторов. Возможно описание проекта за счет внедрения файла, созданного во внешнем редакторе с использованием языков AHDL, VHDL и VerilogHDL. Система позволяет открывать и вносить изменения и сохранять схемные файлы системы OrCAD и файлы трассировки САПР серии ISE.

2. Трассировка связей, проверка корректности проекта, синтез структуры, выполняющиеся посредством компилятора MAX + PLUS II Compiler, являющимся основным элементом системы MAX + PLUS II. Компилятор позволяет выбирать режим его работы и создавать выходные файлы — файлы связей, анализа параметров и др. Основные при-

ложения компилятора — средство для проверки проекта на ошибки "Logic Synthesizer", приложение для извлечения реестра соединений из исходного файла проекта "Netlist Extractor", трассировщик связей "Fitter", утилита для построения базы данных проекта "Database Builder", предназначенная для дальнейшего программирования ПЛИС.

3. Верификация проекта средствами приложений для функционального моделирования проекта "Simulator" и временного анализа (анализ логической цепи, расчет задержек от каждого входа до выхода) "Timing Analyzer".

4. Конечное программирование микросхем — загрузка конечного проекта в ПЛИС или ПЗУ [8, 10–14].

Структурная схема MAX + PLUS II представлена на рис. 1.

Стоит отметить, что выполнение команд и задание параметров в различных компонентах MAX + PLUS II идентично, что существенно сокращает время реализации конечного проекта.

Особенностью MAX + PLUS II является уникальная файловая структура конечного проекта: помимо файлов проекта (графических — .gdf, текстовых — .tdf, сигнальных — .wdf, языковых файлов на языках описания аппаратуры и др.) пользователь или система в автоматическом режиме создает вспомогательные файлы, не содержащие логики проекта: файлы отчета — .rpt, символные — .sym, файлы конфигураций — .acf и другие [8, 11].

Также подчеркнем, что MAX + PLUS II позволяет редактировать файлы одного проекта в рамках использования их в другом проекте, даже если система выполняет тестирование, компиляцию или программирование первого проекта. Справочная система позволяет получить подробную информацию по каждой микросхеме фирмы *Altera*, что помогает выбрать базовую микросхему до начала функционального проектирования [8, 11].

Однако к серьезным недостаткам САПР MAX + PLUS II стоит отнести отсутствие обновлений и

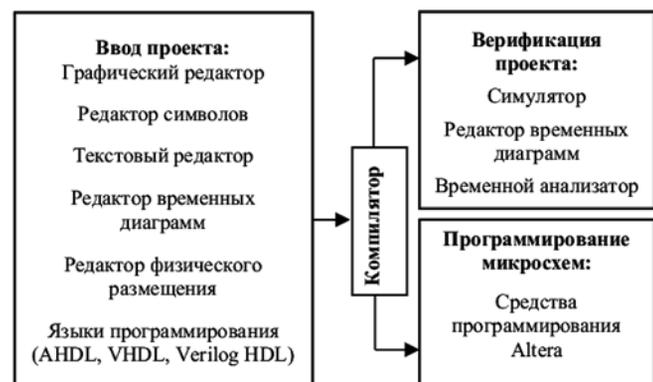


Рис. 1. Схема проектирования устройства в системе MAX + PLUS II [10]

полной технической поддержки пользователей, вследствие чего данный продукт проигрывает в быстрой реакции последним версиям аналогичных САПР. Прежде всего это связано с разработкой компанией *Altera* новой системы проектирования — Quartus II [7].

1.2. Quartus II. Quartus II включает средства для проектирования устройств с применением ПЛИС не только структур FPGA, но и CPLD. САПР поддерживает все этапы проектирования: начиная от создания проекта (логических и вспомогательных файлов), его дальнейшей компиляции, моделирования и заканчивая программированием ПЛИС (общая схема проектирования ПЛИС представлена на рис. 2) [8, 10, 15].

В Quartus II сохранены все программные средства и принципы проектирования, не изменилась и файловая структура проекта, в том числе и привычный интерфейс системы MAX + PLUS II. Однако САПР Quartus II более сложная в освоении, чем система MAX + PLUS II. Основные отличия Quartus II от MAX + PLUS II — поддержка всех новых семейств программируемых микросхем фирмы *Altera* (например, ПЛИС серий MAX II, Cyclone, СБИС программируемой логики серии Stratix и др.), однако некоторые микросхемы старых серий, доступные в MAX + PLUS II, не поддерживаются [10, 15].

Основные возможности САПР:

- поведенческое описание проекта;
- синтез проекта; размещение разводки ПЛИС;
- моделирование проекта; временной анализ, анализ потребляемой энергии устройства;
- программирование ПЛИС;
- проектирование (DSP) блоков цифровой обработки сигналов.

Quartus II не ограничивает стратегии проектирования и позволяет реализовывать проекты на основе как восходящего, так и нисходящего проектирования. Последний способ применяется в тех случаях, когда задан алгоритм работы конечного устрой-

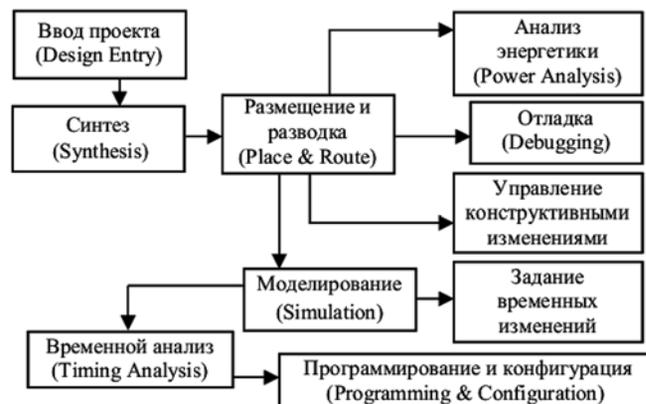


Рис. 2. Структурная схема проектирования ПЛИС с использованием САПР Quartus II [15]

ства. Quartus II поддерживает операционные системы Windows (версий XP, Vista, 7, 8) и Linux [10, 15].

В настоящее время самая последняя версия — Quartus II v12.0, выпущенная в августе 2012 г., стоимость — 2995 долларов, также существует бесплатная версия Quartus II Web Edition, которая функционально ограничена [7].

2. САПР фирмы Xilinx

Компания Xilinx — американский производитель ПЛИС и разработчик программного обеспечения для их проектирования.

Продукты серии ISE. Продукты серии ISE выпускают в различных модификациях: ISE WebPack, ISE Logic Edition, ISE Embedded Edition, ISE DSP Edition. Различаются версии ISE дополнительными программными модулями, входящими в состав системы [16].

Пакет программ ISE WebPack (рис. 3) является самым простым из этой серии и также, как и Quartus II, позволяет выполнять полный цикл проектирования — от начального описания до программирования готового кристалла. САПР предназначен для реализации цифровых систем на базе ПЛИС Xilinx семейств: Virtex и Spartan (до 7-й серии), CoolRunner-II, Kintex-7 и Zynq-7000 [8, 16].

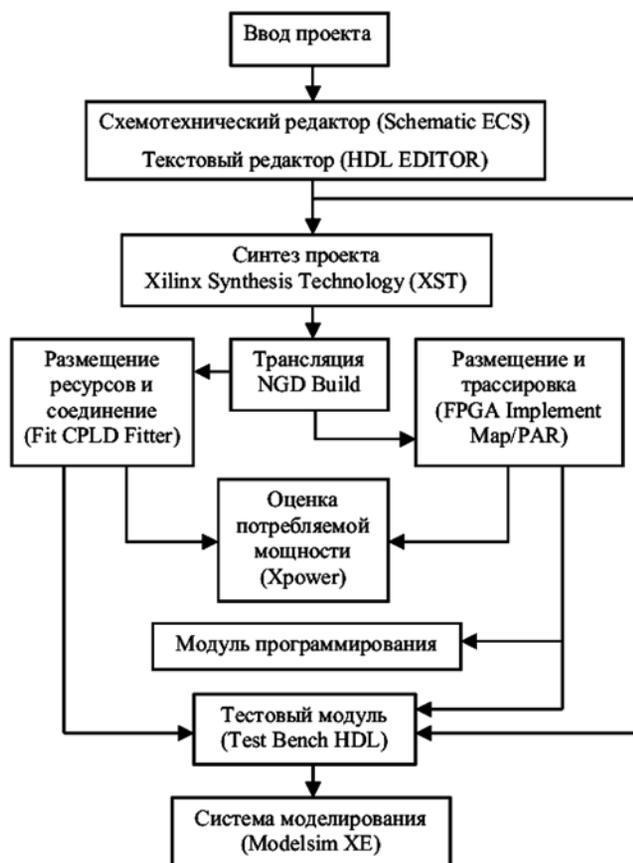


Рис. 3. Маршрут проектирования цифрового устройства с применением ISE WebPack [19]

ISE WebPack является открытой системой и позволяет использовать, изменять и объединять в одно целое компоненты проекта, разработанные в других САПР (например, Aldes или Synopsys). Проектирование не зависит от технологии изготовления ПЛИС: кристаллы на основе ПЛИС FPGA и CPLD программируются одинаково. Пользователям WebPack доступно специальное средство для создания описания модулей в интерактивном графическом режиме — LogiBLOX. Для поддержки работы над кристаллами, емкость которых составляет до 2 000 000 эквивалентов, разработана специальная технология — InternetTeamDesign (Ltd), позволяющая работать над одним проектом одновременно нескольким разработчикам с использованием локальной сети или с помощью Интернета [8, 17].

Помимо собственных разработок (редактор схем и топологий, HDL-описаний и пр.) в пакет САПР входят и программные продукты от других производителей: редактор диаграмм состояний StateCAD, средство для генерации тестовых сигналов HDL Benchner, утилита для функционального моделирования ModelSim и др. [5, 17].

ISE WebPack — это бесплатный продукт, единственное ограничение — поддержка ПЛИС емкостью до 1,5 млн системных вентилях. Другие продукты серии ISE — платные, однако все они имеют бесплатные 30-дневные триал-версии, доступные на официальном сайте компании [5, 18].

Фирма Xilinx предлагает дополнительное программное обеспечение для улучшения производительности системы: приложение для сквозной обработки программируемых процессорных систем на базе ПЛИС Xilinx — Embedded Development Kit (EDK), программное средство Platform Studio, IP-ядра, которые могут потребоваться для разработки встраиваемых систем на основе FPGA Xilinx с встроенными аппаратными ядрами процессора PowerPC и процессорами MicroBlaze [19]. EDK входит в состав пакетов ISE Embedded Edition и ISE System Edition [18].

Программа для анализа топологии кристалла PlanAhead упрощает переход между синтезом и трассировкой проекта и существенно уменьшает длительность итераций повторного проектирования. Использование данного приложения для сложных проектов позволяет увеличить производительность ISE на 30...50 % в сравнении с аналогичными продуктами других производителей [8, 18]. В настоящее время PlanAhead входит в состав всех пакетов программ ISE, в том числе и в состав ISE WebPack [16].

Программа для внутрисхемной логической отладки ПЛИС типа FPGA Chipscope Pro позволяет наблюдать за любым выбранным пользователем сигналом, блоком или узлом внутри интегральной схемы. Chipscope Pro входит в состав всех продуктов ISE, кроме бесплатной версии WebPack [16, 20].

3. САПР компании Altium Limited

Altium Designer (старое название Protel) — программный комплекс австралийского разработчика Altium Limited [21] — программного обеспечения для проектирования электронных устройств. Altium Designer — это первая САПР, разработчики которой особое внимание уделили программируемой логике в современных электронных устройствах.

Программа предназначена для проектирования электронных устройств на базе печатных плат и позволяет создавать проекты, начиная с принципиальной схемы и VHDL-описания ПЛИС и заканчивая подготовкой файлов для производства, включая моделирование, верификацию, трассировку схем.

Стоит отметить, что при использовании Altium Designer разработчику не требуется знания языка VHDL: возможна реализация проекта в виде принципиальной схемы с использованием библиотек готовых логических устройств благодаря особой технологии компании LiveDesign, позволяющей разрабатывать и отлаживать устройство непосредственно в кристалле без VHDL-описания и VHDL-моделирования. Другой особенностью Altium Designer является наличие русифицированного интерфейса и подсказок на русском языке [21, 22].

Компания Altium Limited активно продвигает свою продукцию на российский рынок — в начале 2012 г. на программный пакет Altium Designer произошло существенное снижение стоимости (более чем на 70 %). В настоящее время Altium Designer можно приобрести за 190 тыс. руб. [22].

В целом, рассмотренные САПР для проектирования ПЛИС достаточно похожи. Основное отличие — набор и типовой ряд микросхем, на основе которых планируется реализация проекта. Выбор САПР прежде всего зависит от задачи, поставленной перед разработчиком.

Безусловным преимуществом ISE WebPack перед продуктами фирмы Altera является бесплатное распространение, а благодаря применению дополнительных платных модулей можно обеспечить (практически вдвое) ускорение процесса отладки и трассировки проекта. Для получения первых навыков работы по проектированию ПЛИС подойдут более простые старые версии программных пакетов MAX + PLUS II.

Однако к недостаткам всех рассмотренных САПР можно отнести некоторую отдаленность от производства: реализация проектов заканчивается подготовкой файлов к производству. Разработка проекта, отвечающего требованиям технического задания, и технологический процесс производства происходит с некоторой возможной ошибкой при производстве интегральных схем. Таким образом, итоговая ошибка возрастает.

В модульной структуре САПР (рис. 4) предлагается ввести дополнительный модуль анализа,

расчета параметров технологического процесса и оптимизации, который выполняет построение математической модели при производстве интегральных схем с учетом причинно-следственных связей между показателями качества элементов (узлов, блоков) интегральной микросхемы и параметров физической структуры. Для установления взаимосвязей планируется использовать классические статистические методы обработки информации [22, 25]. Для уменьшения негативного влияния неконтролируемых параметров процесса для его оптимизации используется метод адаптивного управления с подстройкой параметров моделей: после выполнения каждой технологической операции проводится измерение выходных переменных технологического процесса производства интегральных схем, далее значения целевой функции [23] вычисляются и сравниваются с некоторым задаваемым параметром Δ [24].

При необходимости, значения корректируются путем подстройки параметров модели: вычисляются оптимальные значения воздействий $X_{\text{оптим}}$. Для одномерного случая и линейной модели значение

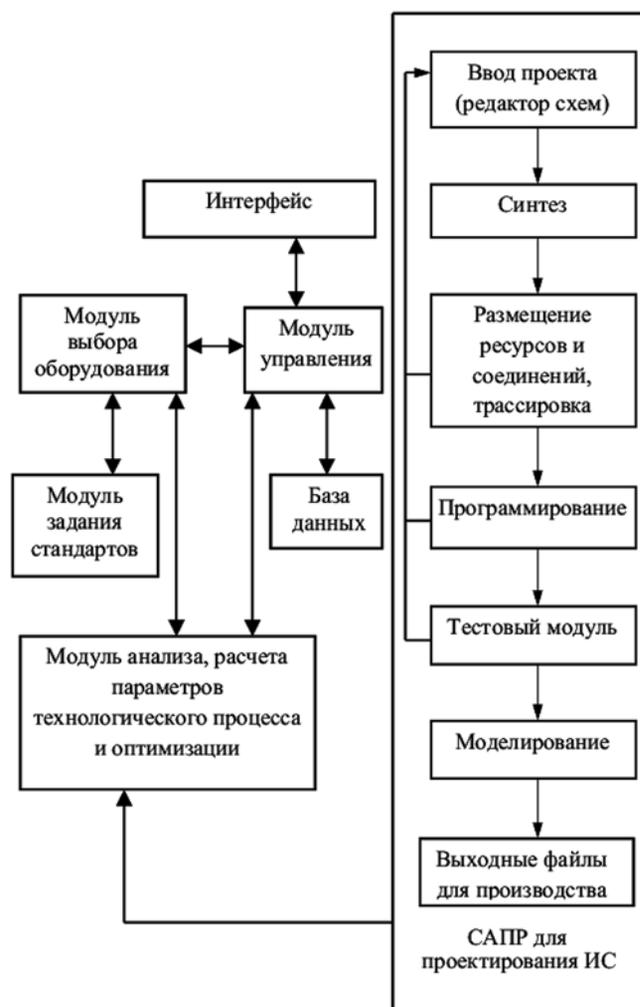


Рис. 4. Модульная схема САПР

целевой функции и процесса, состоящего из двух операций, рассчитывается как [23, 24]:

$$F = \phi_1(Y_{1\text{треб}} - Y_1)^2 + \phi_2(Y_{2\text{треб}} - Y_2)^2 + \phi_3(Z_{1\text{треб}} - Z_1)^2 + \phi_4(Z_{2\text{треб}} - Z_2)^2;$$

$$\frac{F(X_{\text{тек}})}{\partial X_{\text{тек}}} = 2(Y_{\text{треб}} - a_0 - a_1 X_{\text{тек}})(-a_1);$$

$$Y_{\text{треб}} - a_0 - a_1 X_{\text{тек}} = 0;$$

$$X_{\text{оптим}} = X_{\text{тек}} = \frac{Y_{\text{треб}} - a_0}{a_1},$$

где F — целевая функция; ϕ_1, ϕ_3 — векторы весовых коэффициентов выходных параметров, а ϕ_2, ϕ_4 — векторы весовых коэффициентов управляющих переменных процесса или технологической операции соответственно; $Y_{\text{треб}}$ — требуемые значения выходных переменных; $X_{\text{оптим}}$ — оптимальное значение входных параметров для следующей операции; a_0, a_1 — коэффициенты, определяемые экспериментально или из статистических данных; $Z_{\text{треб}}$ — требуемые значения воздействий на процесс; Z_1, Z_2 — значения воздействий для первых и второй технологических операций.

Впоследствии посредством анализа результатов, полученных в рамках математической модели, с учетом требуемых стандартов качества (согласно техническому заданию или ГОСТ), технический работник при реализации технологического процесса будет иметь данные о надежности изготавливаемых приборов и оценки влияния технологических факторов на работоспособность разрабатываемой ИС. На рис. 5 представлена функциональная схема, на которой продемонстрировано взаимодействие отдельных модулей в составе предлагаемого функционального блока.

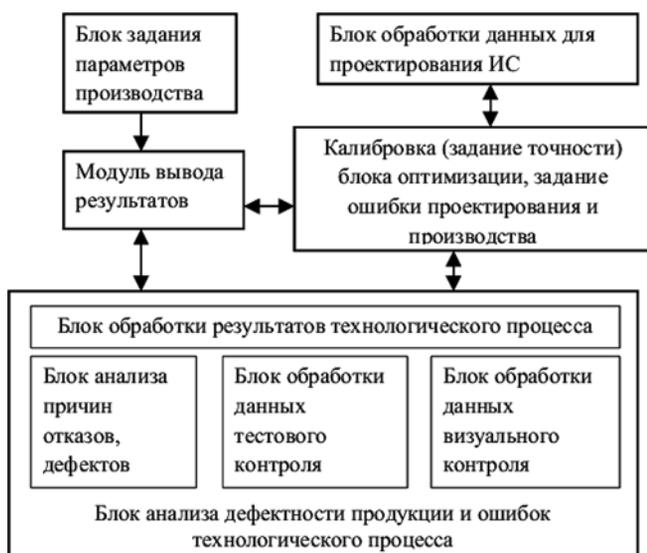


Рис. 5. Функциональная схема модуля анализа и оптимизации технологического процесса

1. Томас Ф., Иванов А. САПР микроэлектроники. Этапы большого пути // Электроника: Наука, Технология, Бизнес. 2006. № 3. С. 82—85.
2. Перкин П. В. Использование распределенных вычислений при разработке цифровых устройств управления // Информационные управляющие системы и компьютерный мониторинг: III Всеукраинская научно-техническая конференция студентов, аспирантов и молодых ученых: сборник материалов конференции. Донецк: ДонНТУ, 2012.
3. Нефедов А. В., Савченко А. М., Феоктистов Ю. Ф. Зарубежные интегральные микросхемы для промышленной электронной аппаратуры. М.: Энергоатомиздат, 1989. 226 с.
4. Негрбов П. Функционирование и синтез цифровых устройств // Принцип микропрограммного управления. URL: <http://schem.net/beginner/beginner106.php>.
5. Тарасов И. Е. Разработка цифровых устройств на основе ПЛИС Xilinx с применением языка VHDL. М.: Горячая линия-Телеком, 2005. 252 с.
6. Сайт Verilog Resources. URL: <http://www.verilog.com>
7. Сайт Altera Corporation. www.altera.com
8. Еркин А. Обзор современных САПР для ПЛИС. // Chip News: электронный научный журнал. 2008. № 10. URL: http://chip-news.ru/archive/chipnews/200810/Article_05.pdf
9. Дмитриенко А. А., Павлов В. М., Байструков К. И. Аппаратное и программное обеспечение лабораторного комплекса "Система многоканальной диагностики быстропротекающих процессов СМД". Томск: Изд-во Томского политехнического университета, 2011. 87 с.
10. Комолов Д. А., Мяльк Р. А., Зобенко А. А., Филиппов А. С. Системы автоматизированного проектирования фирмы Allera MAX + Plus I и Quartus II. М.: РадиоСофт, 2002. 256 с.
11. Стешенко В. Б. ПЛИС фирмы Altera: проектирование устройств обработки сигналов. М.: ДОДЭКА, 2000. 128 с.
12. Поречный В. Использование САПР "MAX + plus II" для разработки цифровых устройств на ПЛИС фирмы "Альтера" // ЭПОС. URL: http://www.epos.ua/view.php/pubs_39.
13. Прищепа С. Л., Ильина Е. А. Проектирование цифровых схем с помощью САПР MAX + plus II фирмы Altera. Минск: БГУИР, 2005. 52 с.
14. Бибило П. Н. Синтез логических схем с использованием языка VHDL. Минск: СОЛОН-Р, 2002. 304 с.
15. Опядчий Ю. Ф. Общая технология проектирования в среде Quartus II. Часть 1. "Общее описание проекта". М.: Изд-во МГТУ им. К. Э. Циолковского, 2005. 64 с.
16. ISE Design Suite. URL: <http://www.xilinx.com/products/design-tools/ise-design-suite/index.htm>
17. Зотов В. Ю. Проектирование цифровых устройств на основе ПЛИС фирмы Xilinx в САПР WebPack ISE. М.: Горячая линия-Телеком, 2003. 624 с.
18. Перекрест А. Обзор современных ПЛИС и САПР фирмы Xilinx // Компоненты и технологии. 2006. № 3.
19. Семилляк А. И. Реализация цифровых устройств на базе ПЛИС фирмы Xilinx с применением инструментального комплекса SPARTAN-3E STARTER KIT. URL: <http://meet.dstu.ru/files/sbornik2/6.doc>, свободный.
20. ПЛИС — программируемые логические интегральные схемы. URL: <http://www.fpga-cpld.ru/sapnew8.html>
21. Altium Limited. URL: <http://www.altium.com>
22. Система сквозного проектирования Altium Designer. URL: http://www.rodnik.ru/product/sapr/pp_i_plis/altium/altium_designer_soft_vhdl
23. Ли Т. Т., Адаме Т. Э., Гейнз У. М. Управление процессами с помощью ЭВМ. Моделирование и оптимизация. М.: Сов. радио, 1972. 256 с.
24. Салеев Д. В. Выделение главного параметра (показателя качества) для многокритериальной оптимизации технологического процесса // Техника и технология: новые перспективы развития: материалы IX Международной научно-практической конференции (24.05.13). М.: Спутник +, 2013. С. 36—40.
25. Салеев Д. В. Применение статистических методов для управления качеством интегральных схем // Новые технологии в научных исследованиях, проектировании, управлении, производстве: труды Всероссийской конференции. Воронеж: Изд-во Воронежского государственного технического университета, 2013. С. 110—111.

An Analysis of a Features of CAD for FPGA

In this paper an existing Current Aided Design (CAD) systems for FPGAs are analyzed. It is established that as a whole considered CAD are similar. The key difference of them is the typical kit of chips, on the basis of which a project implementation is planned. In all analyzed CAD systems, projects finalized by preparation of production files, and software tools, that allow to integrate received files into the process of production of chips, in order to decrease the technological errors in the production process are not developed. To extend the capabilities of the existing CAD systems, generalized structural and functional schemes of the optimization module are developed. Proposed the introduction of developed additional software unit to evaluate the influence of technological factors on the functionality, according to the customer requirements.

Keywords: chip, FPGA, chip design, CAD systems, VHDL, optimization, technological process, algorithm

References

1. **Tomas F.** SAPR mikroelektroniki. Jetapy bol'shogo puti. *Jelektronika: Nauka, Tehnologija, Biznes.* 2006. N. 3. P. 82–85.
2. **Perkin P. V.** Ispol'zovanie raspredelennyh vychislenij pri razrabotke cifrovych ustrojstv upravlenija. *Informacionnye upravljajushhie sistemy i komp'juternyj monitoring: III Vseukrainskaja nauchno-tehnicheskaja konferencija studentov, aspirantov i molodyh uchenyh: sbornik materialov konferencii.* Donetsk: DonNTU, 2012. P. 11–14.
3. **Nefedov A. V., Savchenko A. M., Feoktistov J. F.** Zarubezhnye integral'nye mikroshemy dlja promyshlennoj jelektronnoj apparatury, Moscow: *JeNERGOATOMIZDAT*, 1989. 226 p.
4. **Negrobov P.** Funkcionirovanie i sintez cifrovych ustrojstv, *Princip mikroprogrammnoho upravlenija.* URL: <http://cxem.net/beginner/beginner106.php>
5. **Tarasov I. E.** Razrabotka cifrovych ustrojstv na osnove PLIS Xilinx s primeneniem jazyka VHDL. Moscow: *Gorjachaja linija-Tel-ekom*, 2005. 252 p.
6. **Verilog Resources.** URL: <http://www.verilog.com>
7. **Altera Corporation.** URL: www.altera.com
8. **Erkin A.** Obzor sovremennyh SAPR dlja PLIS. *Chip News. jelektronnyj nauchnyj zhurnal.* 2008. N. 10. P. 23–27. URL: http://chip-news.ru/archive/chipnews/200810/Article_05.pdf
9. **Dmitrienko A. A., Pavlov V. M., Bajstrukov K. I.** Apparatoe i programmnoe obespechenie laboratornogo kompleksa "Sistema mnogokanal'noj diagnostiki bystroprotekajushhijh processov SMD", Tomsk: Izdatelstvo Tomskogo politehnicheskogo universiteta, 2011. 87 p.
10. **Komolov D. A., Mjal'k R. A., Zobenko A. A., Filippov A. S.** Sistemy avtomatizirovannogo proektirovanija firmy Altera MAX + Plus II i Quartus II., Moscow: RadioSoft, 2002. 256 p.
11. **Steshenko V. B.** PLIS firmy Altera: proektirovanie ustrojstv obrabotki signalov. Moscow: DODJeKA, 2000. 128 p.
12. **Porechnyj V.** Ispol'zovanie SAPR "MAH + plus II" dlja razrabotki cifrovych ustrojstv na PLIS firmy "Altera". *JePOS.* URL: http://www.epos.ua/view.php/pubs_39
13. **Prishhepa S. L., Il'ina E. A.** Proektirovanie cifrovych shem s pomoshh'ju SAPR MAH + plus II firmy Altera. Minsk: BGUIR, 2005. 52 p.
14. **Bibilo P. N.** Sintez logicheskijh shem s ispol'zovaniem jazyka VHDL. Minsk: SOLON-R, 2002. 304 p.
15. **Opadchij J. F.** Obshhaja tehnologija proektirovanija v srede Quartus II. Chast' 1. Obshhee opisanie proekta. M.: Izdatelstvo MGTU im. K. J. Ciolkovskogo, 2005. 64 p.
16. **ISE Design Suite.** URL: <http://www.xilinx.com/products/design-tools/ise-design-suite/index.htm>
17. **Zotov V. J.** Proektirovanie cifrovych ustrojstv na osnove PLIS firmy Xilinx v SAPR WebPack ISE, Moscow: Gorjachaja linija-Telekom, 2003. 624 p.
18. **Perekrest A.** Obzor sovremennyh PLIS i SAPR firmy Xilinx. *Komponenty i tehnologii.* 2006. N. 3.
19. **Semiljak A. I.** Realizacija cifrovych ustrojstv na baze PLIS firmy Xilinx s primeneniem instrumental'nogo kompleksa SPARTAN-3E STARTER. URL: <http://meet.dstu.ru/files/sbornik2/6.doc>
20. **PLIS** — programmiruemye logicheskie integral'nye shemy. URL: <http://www.fpga-cpld.ru/sapmew8.html>
21. **Altium Limited.** URL: <http://www.altium.com>
22. **Sistema skvoznogo proektirovanija Altium Designer.** URL: http://www.rodnik.ru/product/sapr/pp_i_plis/altium/altium_designer_soft_vhdl
23. **Li T. T., Adame T. J., Gejnz U. M.** Upravlenie processami s pomoshh'ju EVM. Modelirovanie i optimizacija. Moscow: Sovetskoe radio, 1972. 256 p.
24. **Saleev D. V.** Vydelenie glavnogo parametra (pokazatelja kachestva) dlja mnogokriterial'noj optimizacii tehnologicheskogo processa. *Tehnika i tehnologija: novye perspektivy razvitiya: materialy IX Mezhdunarodnoj nauchno-prakticheskoy konferencii (24.05.13)*, Moscow: Sputnik, 2013. P. 36–40.
25. **Saleev D. V.** Primenenie statisticheskijh metodov dlja upravlenija kachestvom integral'nyh shem. *Novye tehnologii v nauchnyh issledovanijah, proektirovanii, upravlenii, proizvodstve: trudy vsersijskoj konferencii.* Voronezh: Voronezhskij gosudarstvennyj tehniceskij universitet", 2013. P. 110–111.

БЕЗОПАСНОСТЬ ИНФОРМАЦИИ CRYPTOSAFETY INFORMATION

УДК 004.056.53

К. А. Щеглов, магистрант, А. Ю. Щеглов, д-р техн. наук, проф.,
Национальный исследовательский университет информационных технологий, механики и оптики,
Санкт-Петербург, e-mail: info@npp-itb.spb.ru

Защита от атак на уязвимости приложений

Исследованы вопросы защиты от атак на уязвимости приложений, изложен подход к защите, основанный на реализации разграничительной политики доступа к защищаемым ресурсам для критичных процессов (приложений), проиллюстрированный апробированными техническими решениями. Предложенный подход универсален в том смысле, что его эффективное применение возможно не только в целях защиты от атак, направленных на использование уязвимости приложений (что рассмотрено в данной работе), но и в целях защиты от любого рода атак, связанных с наделением приложений несанкционированными возможностями (за счет макровирусов, вредоносных скриптов и т. п.) и последующей эксплуатацией этих возможностей.

Ключевые слова: компьютерная безопасность, защита информации, атака, несанкционированный доступ, уязвимость приложений, контроль доступа, разграничительная политика доступа

Введение

Одной из ключевых задач защиты информации в современных условиях становится защита от атак на уязвимости приложений. Чтобы оценить актуальность этой задачи защиты, достаточно обратиться к соответствующей статистике, которая представлена, например, в работе [1] и проиллюстрирована на рис. 1.

Отметим, что в качестве критичных уязвимостей, атаки на которые необходимо предотвращать средством защиты, следует рассматривать, по крайней мере, уязвимости, характеризующиеся критической, высокой и средней степенью опасности. Классификация уязвимостей приведена в работе [1] (например, к средней степени опасности относятся уязвимости, которые позволяют реализовать удаленный отказ в обслуживании, неавторизованный доступ к данным или выполнение произвольного кода).

Оценка готовности приложений к безопасной эксплуатации с учетом статистики выявляемых в них уязвимостей и с использованием соответствующих математических моделей была нами получена в работе [2]. Например, зависимость изменения данной характеристики от изменения интенсивности выявляемых уязвимостей в приложениях (при средней продолжительности исправления уязвимости один месяц) представлена на рис. 2.

Чтобы оценить всю критичность сложившейся ситуации с безопасностью существующих приложений, достаточно вновь обратиться к рис. 1, но уже с учетом исследований, проиллюстрированных на

рис. 2. Как видим, значение вероятности готовности приложений к безопасной эксплуатации современных приложений в лучшем случае не превышает 0,3.

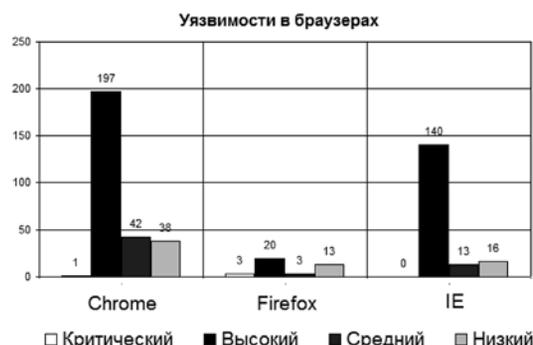


Рис. 1. Статистика обнаруженных уровней уязвимостей в браузерах за 2011 г.

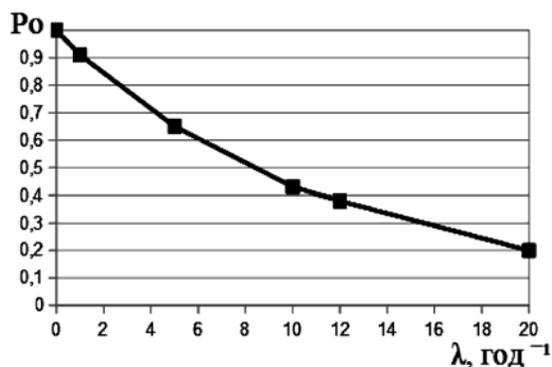


Рис. 2. Влияние интенсивности обнаружения уязвимостей на эксплуатационную безопасность приложений

Отметим фундаментальность исследуемой проблемы. Достаточно очевидно, что программные средства из года в год только усложняются, при этом сроки их разработки, диктуемые конкурентным рынком, сокращаются, и, как следствие, ошибок в программных средствах становится больше, а исправлять их становится все сложнее. Это объективная реальность. С учетом сказанного можно предположить, что из года в год актуальность проблемы защиты от атак на уязвимости приложений будет лишь возрастать, что и обуславливает фундаментальность данной проблемы.

Предлагаемый подход к защите

Предлагаемый подход основан на следующих предпосылках.

1. Естественно, что в первую очередь при реализации защиты информации возникает задача защиты (которую мы и будем рассматривать в работе) от так называемых целевых атак, атак на компьютерную систему, реализуемых с определенной конечной целью. Основными целями атак являются:

- цель 1 — нарушение конфиденциальности обрабатываемой информации (хищение информации);
- цель 2 — нарушение целостности обрабатываемой информации (модификация — подмена информации, дезинформация);
- цель 3 — нарушение доступности обрабатываемой информации (уничтожение информации либо вывод из строя информационной системы).

2. Основу защиты информации в компьютерной системе от несанкционированного доступа составляет реализация разграничительной политики доступа к ресурсам. Именно это решение мы и рассмотрим в работе. Однако в нашем случае в качестве субъекта доступа уже выступает сущность "процесс" (приложение). Необходимость реализации самостоятельной разграничительной политики доступа для процессов (приложений) обуславливается тем, что в современных операционных системах (ОС) любой запускаемый процесс наследует права доступа к ресурсам пользователя, запустившего данный процесс, т. е. все процессы (приложения), запускаемые одним и тем же пользователем, вне зависимости от статистики их уязвимости и критичности атак на уязвимости приложений будут обладать одними и теми же правами доступа к защищаемым ресурсам.

3. Защита должна строиться в предположении о том, что нам неизвестно, какая уязвимость присутствует и/или будет обнаружена в приложении в ближайшее время, к каким возможностям она приведет. То есть будем считать, что приложение, запущенное интерактивным пользователем, в результате эксплуатации обнаруженной в нем уязвимости может осуществить любое несанкционированное действие под управлением этого пользователя.

Таким образом, в работе излагается подход к защите от целевых атак на уязвимости приложений, основанный на реализации разграничительной политики доступа к ресурсам для субъекта доступа "процесс".

Замечание. Излагаемые в работе подходы к защите от атак на уязвимости приложений авторами апробированы и будут проиллюстрированы на примере технического решения компьютерной системы защиты информации "Панцирь+" для ОС Microsoft Windows.

Замечание. Реализацию разграничительной политики доступа процессов к ресурсам будем в работе иллюстрировать на примере решения одной из наиболее актуальных практических задач защиты — защиты от атак на уязвимости Интернет-браузеров (на примере широко используемого браузера IE). Понятно, что это лишь пример, все излагаемые в работе подходы могут аналогично использоваться при реализации защиты от атак на уязвимости иных приложений.

Определение субъекта доступа в разграничительной политике

Субъекты доступа в общем случае должны назначаться тремя сущностями:

- исходным именем пользователя;
- эффективным именем пользователя;
- именем процесса — полнопутевым именем исполняемого файла процесса из интерфейса, представленного на рис. 3 (в первую очередь, как отмечали, интерес здесь для нас представляет сущность "процесс" как основной элемент разграничительной политики доступа к ресурсам).

Однако не будем забывать, что одни и те же механизмы защиты должны использоваться для реализации разграничений прав доступа и для сущности "процесс" (в нашем случае), и для сущностей пользователя. Обоснование необходимости задания субъекта доступа в общем случае тремя сущностями приведено в работах [3—5]. При задании же сущности "процесс" в субъекте доступа могут использоваться маски и переменные среды окружения.

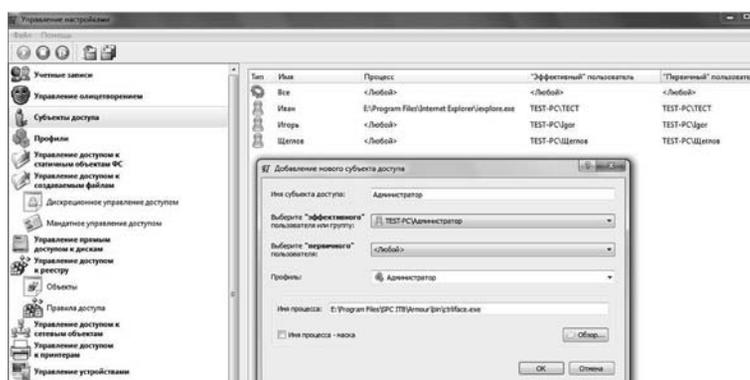


Рис. 3. Задание и отображение в интерфейсе субъектов доступа

Защита от атак на нарушение конфиденциальности, доступности и целостности обрабатываемой информации

В работах [6, 7] введена классификация файловых объектов, они подразделены на статичные и создаваемые. Статичные — это те объекты, которые присутствуют в системе на момент реализации администратором разграничительной политики доступа к ресурсам, в первую очередь — системные объекты. Обрабатываемая же в компьютерной системе информация (защита которой нас здесь интересует) сохраняется в файлах, создаваемых в процессе работы пользователя.

Принципы контроля доступа (разграничения прав доступа) к создаваемым файловым объектам предложены в работах [6, 7], реализация метода контроля доступа и сформулированные требования к корректности реализации разграничительной политики доступа рассмотрены, например, в статье [5].

Основная идея данного решения состоит в исключении сущности "объект доступа" из разграничительной политики как таковой (ввиду ее отсутствия на момент задания администратором правил доступа к ресурсам). Разграничение (контроль доступа) реализуется непосредственно между субъектами доступа к создаваемым ими файлам.

Замечание. Уточним, что разграничительная политика в любом случае заключается в реализации разграничений прав доступа субъектов к объектам

(в данном случае — к файлам), здесь же речь идет об исключении сущности "объект доступа" из назначаемых администратором правил доступа. В правилах указывается то, какие субъект имеет права доступа к файлам, созданным иными субъектами, — в правилах отсутствует сущность "объект доступа".

Реализуется контроль доступа следующим образом. При создании субъектом нового файла этот файл средством контроля доступа (диспетчером доступа) автоматически размечается — файлом наследуется учетная информация субъекта доступа (определяемая соответствующими тремя сущностями), создавшего этот файл. Данная информация размещается в атрибутах созданного файла.

При запросе же доступа к любому файлу средство контроля доступа анализирует наличие унаследованной файлом учетной информации создавшего его субъекта доступа и при наличии — ее содержимое. Это осуществляется посредством считывания и анализа атрибутов файла, к которому запрошен доступ.

В соответствии с заданными администратором правилами доступа, задаваемыми исключительно между субъектами доступа из интерфейса (рис. 4), диспетчер доступа предоставляет запрошенный субъектом доступ либо отказывает в нем, признавая тем самым запрос доступа несанкционированным.

В двух словах скажем о назначении правил доступа. В правом столбце интерфейса "Выберите субъекты создателей" (рис. 4) задаются субъекты, создающие файлы, последующий доступ к которым будет контролироваться (разграничиваться). Для каждого субъекта заданного в правом столбце интерфейса, в левом столбце интерфейса "Выберите субъекты, осуществляющие доступ" задаются субъекты, которым разрешается доступ к файлам, создаваемым заданным субъектом-создателем, и назначаются права доступа к этим файлам (чтение, запись, переименование, удаление), а также режимы аудита. Право доступа "исполнение" запрещено по умолчанию и, как следствие, не вынесено в интерфейс (рис. 4) (запрет исполнения создаваемых файлов является основой эффективной защиты от вредоносных программ [8]).

Рассмотрим решение задачи защиты, где в качестве уязвимого приложения будем рассматривать Интернет-браузер IE. Заведем из интерфейса (см. рис. 3) два субъекта доступа — "Все" и "IE" (рис. 5) и зададим из интерфейса (см. рис. 3) правила доступа (рис. 6).

Посмотрим, что мы получим в результате. Интернет-браузер вне зависимости от того, какими несанкционированными свойствами он будет наделен, не получит доступ к конфиденциальной информации, обрабатываемой на компьютере, — к создаваемым иными приложениями файлам (его работа в информационной системе в этой части изолирована).

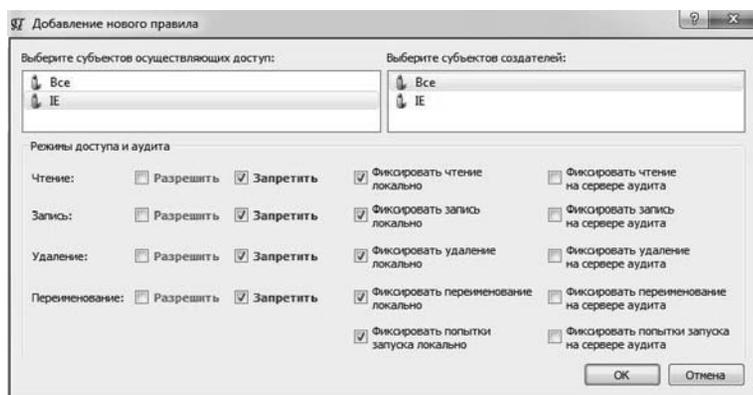


Рис. 4. Задание и отображение в интерфейсе правил доступа к создаваемым файлам

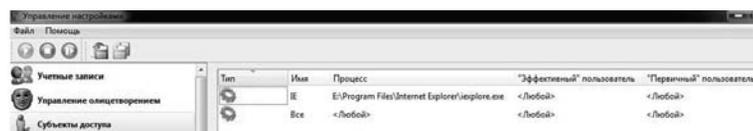


Рис. 5. Заданные субъекты доступа

Субъект осуществляющий доступ	Субъект-создатель файла	Режим доступа	Режим аудита
Все	IE	+Ч+З-И+У+П	ЧЗИУП:-----
IE	Все	-Ч-З-И-У-П	ЧЗИУП:-----

Рис. 6. Заданные правила доступа

Предотвращается возможность нарушения конфиденциальности, целостности и доступности (в части защиты от удаления) обрабатываемой на компьютере информации в результате реализации атаки на уязвимость Интернет-браузера.

Иные же приложения при данной разграничительной политике имеют доступ (кроме исполнения) к файлам, создаваемым IE. На этом моменте следует акцентировать внимание. Уязвимый интернет-браузер может создать вредоносный файл (например, содержащий макровирус), при чтении которого другим приложением данное приложение будет наделено вредоносными свойствами. Из этого следует, что целесообразно не только запретить полный доступ браузеру к файлам, создаваемым иными приложениями, но и полный доступ иных приложений к файлам, создаваемым браузером, по крайней мере, тех приложений, которые при прочтении вредоносного файла могут быть наделены вредоносными свойствами. Эти вопросы исследованы в работе [9], где в том числе представлена корректная модель контроля доступа, приведено ее обоснование.

Защита от атак на системные ресурсы

Если мы говорим о контроле (разграничении прав) доступа к системным ресурсам — к статичным объектам [4] (в общем случае — к системным файловым объектам, объектам реестра ОС, к устройствам, к сетевым ресурсам и т. д.), то при реализации разграничительной политики доступа субъектов к объектам необходимо задать объекту, доступ к которым будет разграничиваться, что, например, применительно к файловым объектам (включая внешние файловые накопители) реализуется из интерфейса, представленного на рис. 7. Соответственно требуется задать и правила доступа субъектов, заданных для системы с помощью интерфейса, представленного на рис. 3, к объектам, что реализуется с помощью интерфейса, представленного на рис. 8.

Замечание. Объект доступа может задаваться как своим полнопутьным именем, так и маской, а также переменными среды окружения.

Рассмотрим реализацию защиты системных ресурсов на примере системных файловых объектов и объектов реестра ОС от целевых атак на уязвимости Интернет-браузера (приложения). Соответствующие разграничительные политики для субъекта IE приведены на рис. 9 и на рис. 10.

Оценим получаемый результат. Разграничительной политикой доступа к статичным

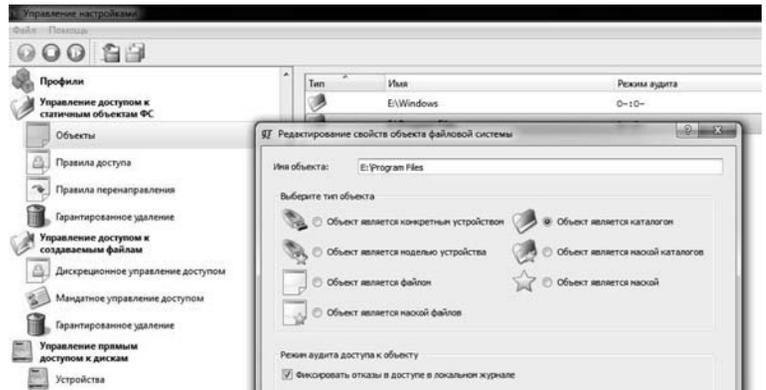


Рис. 7. Интерфейс задания файлового объекта доступа

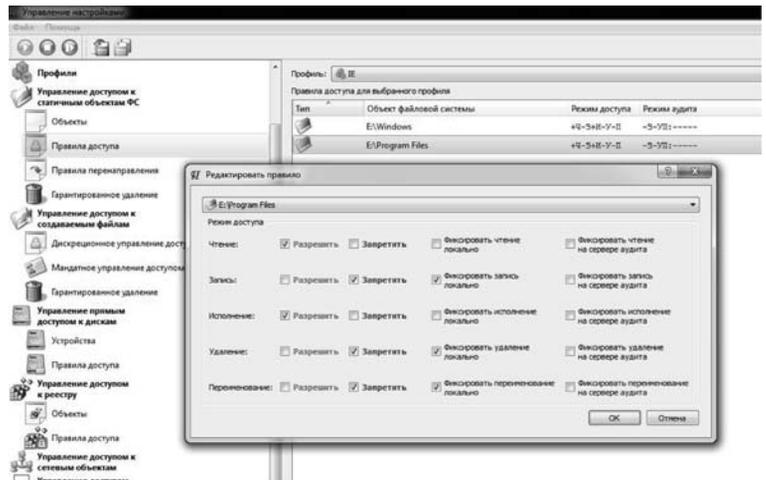


Рис. 8. Интерфейс задания прав доступа субъекта к объекту

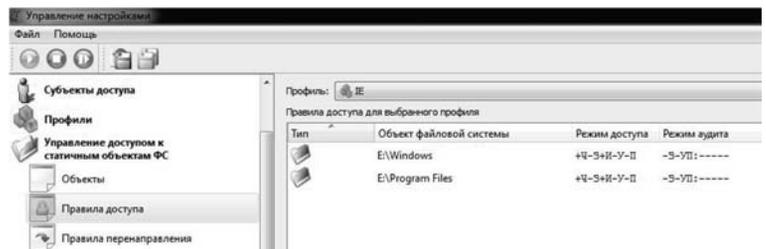


Рис. 9. Заданные правила доступа к системным файловым объектам

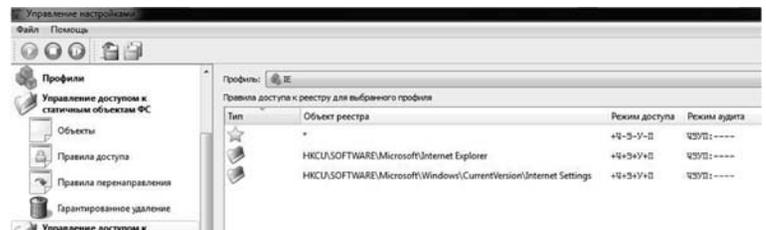


Рис. 10. Заданные правила доступа к объектам реестра ОС

(системным) файловым объектам (см. рис. 9) предотвращается возможность несанкционированной модификации приложением IE исполняемых файлов и файлов настройки ОС и приложений. Разграничительной же политикой доступа к объектам реестра ОС (см. рис. 10) IE разрешается доступ только к необходимым ему для корректного функционирования объектам реестра.

Отметим, что в данном случае решается задача защиты от атак на уязвимости приложения, реализуемых в целях нарушения доступности обрабатываемой информации за счет вывода из строя информационной системы в результате уничтожения/модификации системных ресурсов.

Дополнительные меры защиты

Напомним, что исходным посылом при реализации защиты от атак на уязвимости приложений исходно являлась реализация разграничительной политики доступа к ресурсам в предположении о том, что приложение, запущенное интерактивным пользователем, в результате эксплуатации обнаруженной в нем уязвимости может осуществить любое несанкционированное действие под управлением этого пользователя. Это обуславливает целесообразность реализации ряда дополнительных мер,

направленных на противодействие обходу злоумышленником реализованной разграничительной политики доступа к создаваемым и статичным файловым объектам. Основными способами обхода разграничительной политики доступа к ресурсам на практике сегодня являются повышение привилегий пользователя (в том числе получение прав администратора либо системного пользователя) за счет сопоставления (штатная возможность современных ОС [3]) уязвимого процесса (точнее потока, порожденного процессом) с правами другого пользователя и прямой доступ к диску (не к файловому объекту, к которому разграничиваются права доступа).

Соответствующая разграничительная политика доступа для субъекта IE в части предотвращения прямого доступа приложения к диску представлена на рис. 11, в части предотвращения возможности сопоставления приложения, запускаемого из-под одного пользователя, с иным пользователем (в том числе привилегированным) — на рис. 12.

Отметим, что с помощью интерфейса, приведенного на рис. 11, целесообразно настроить правила контроля доступа, противодействующие прямому доступу IE к любому диску (целесообразно запретить прямой доступ не только к жесткому диску, но и к любому внешнему накопителю, при этом объект доступа в разграничительной политике задается маской "Все" — обозначение "*"), а с помощью интерфейса, приведенного на рис. 12, целесообразно настроить правила контроля, противодействующие смене имени пользователя, от "лица" которого запущен IE, на любое иное имя пользователя, от "лица" которого далее IE будет функционировать (целесообразно запретить любую смену имени (SID) пользователя для приложения IE).

Заметим, что существуют и иные меры дополнительной защиты, правда, уже не столь актуальные, как изложенные, которые ввиду ограничения на объем рукописи не рассмотрены в данной работе.

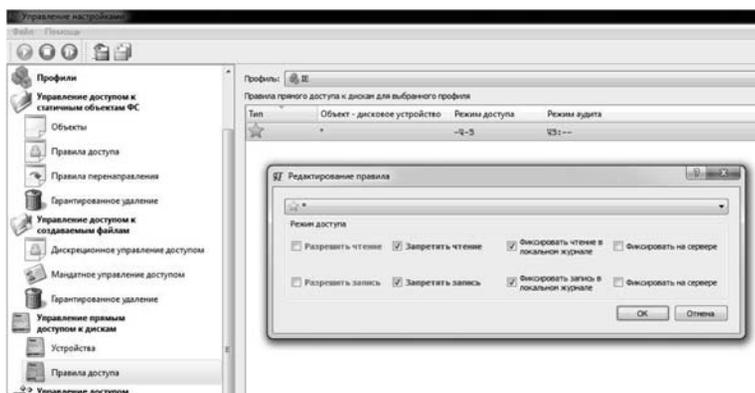


Рис. 11. Задание правил прямого доступа приложения к дискам

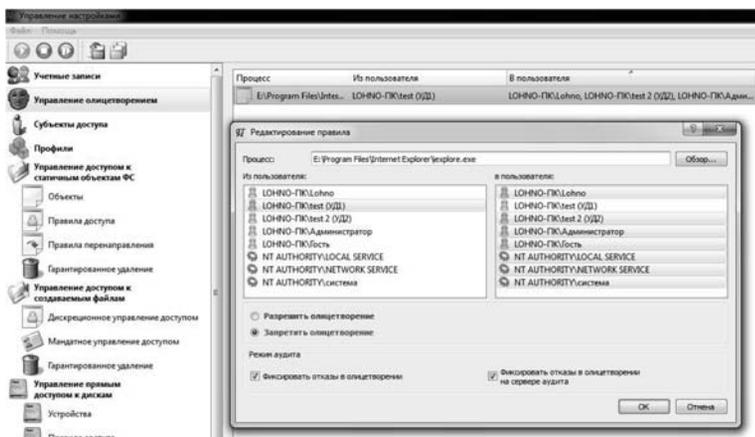


Рис. 12. Задание правил олицетворения приложения с иными пользователями

Заключение

- Отметим следующие важные моменты:
- ♦ Как видим, одна из актуальнейших современных задач защиты информации может быть решена, причем решена эффективно. Однако решение этой задачи требует пересмотра основных устоявшихся принципов реализации разграничительной политики доступа к ресурсам, требует реализации новых подходов к защите информации от несанкционированного доступа.
 - ♦ Изложенный подход к защите — это не некая теоретическая возможность, все технические решения реализованы и апробированы.

- ◆ Изложенный подход к защите универсален в том смысле, что его эффективное применение возможно не только в целях защиты от атак, направленных на эксплуатацию уязвимости приложений (что рассмотрено нами в данной работе), но и в целях защиты от любого рода атак, связанных с наделением приложений несанкционированными возможностями (за счет макровирусов, вредоносных скриптов и т. п. [9]) и последующей эксплуатацией этих возможностей.

Список литературы

1. **Статистика** уязвимостей за 2011 год от Лаборатории Касперского. URL: <http://it-sektor.ru/statistika-uyazvymosteyi-za-2011-god-ot-laboratorii-kasperskogo.html>
2. **Щеглов К. А., Щеглов А. Ю.** Защита от атак на уязвимости приложений. Модели контроля доступа // Вопросы защиты информации. 2013. Вып. 101, № 2. С. 36—43.
3. **ГОСТ Р 53114—2008.** Защита информации. Обеспечение информационной безопасности в организации, 2009.

4. **Малюк А. А.** Информационная безопасность: концептуальные и методологические основы защиты информации. М.: Горячая линия — Телеком. 2004. 280 с.
5. **Щеглов К. А., Щеглов А. Ю.** Методы идентификации и аутентификации пользователя при доступе к файловым объектам // Вестник компьютерных и информационных технологий. 2012. № 10. С. 47—51.
6. **Щеглов К. А., Щеглов А. Ю.** Принцип и методы контроля доступа к создаваемым файловым объектам // Вестник компьютерных и информационных технологий. 2012. № 7. С. 43—47.
7. **Щеглов К. А., Щеглов А. Ю.** Практическая реализация дискреционного метода контроля доступа к создаваемым файловым объектам // Вестник компьютерных и информационных технологий. 2013. № 4. С. 43—49.
8. **Щеглов К. А., Щеглов А. Ю.** Защита от вредоносных программ методом контроля доступа к создаваемым файловым объектам // Вестник компьютерных и информационных технологий. 2012. № 8. С. 46—51.
9. **Щеглов К. А., Щеглов А. Ю.** Защита от атак со стороны приложений, наделяемых вредоносными функциями. Модели контроля доступа // Вопросы защиты информации. 2012. Вып. 99, № 4. С. 31—36.
10. **Щеглов К. А., Щеглов А. Ю.** Контроль доступа к статичным файловым объектам // Вопросы защиты информации. 2012. Вып. 97, № 2. С. 12—20.

К. А. Shcheglov, Student, **A. Yu. Shcheglov**, Professor,

Saint-Petersburg National Research University of Information Technologies, Mechanics and Optics, Russia,
e-mail: info@npp-itb.spb.ru

Securing Against Applications Threats

In this article we research problem of securing against application threats, review suggested securing method, based on applying access policy for resources used by critical processes (applications) which is illustrated with tested technical solutions. Suggested method is universal in terms that its effective implementation is possible not only targeting security against attacks exploiting applications vulnerabilities (which is covered in this article), but also targeting security against any type of attacks related with applying unauthorized abilities to application (using macro-virus, malicious scripts, etc.) and following exploitation. Reviewed protection method is based on practical realization which was patented by authors of "File objects access control system based on auto-labeling" solution. This solution allows to rethink known realization of both access control methods including discretionary and mandate ones. This not only dramatically simplifies setting file objects access policy (by eliminating the "access object" essence from access control scheme), but also settings correct implementation in the general case is provided in the same time.

Keywords: computer security, information security, attack, unauthorized access, application threats, access control, access policy

References

1. **Статистика** уязвимостей за 2011 год от Лаборатории Касперского [Elektronnyj resurs]. URL: <http://it-sektor.ru/statistika-uyazvymosteyi-za-2011-god-ot-laboratorii-kasperskogo.html>
2. **Shcheglov K. A., Shcheglov A. Yu.** Zashhita ot atak na uязvymosti prilozhenij. Modeli kontrolya dostupa. *Voprosy zashhity informacii*. 2013. Iss. 101, N. 2. P. 36—43.
3. **ГОСТ Р 53114—2008.** Zashhita informacii. Obespechenie informacionnoj bezopasnosti v organizacii, 2009.
4. **Maljuk A. A.** Informacionnaja bezopasnost: konceptualnye i metodologicheskie osnovy zashhity informacii. M.: Gorjachaja linija Telekom, 2004. 280 p.
5. **Shcheglov K. A., Shcheglov A. Yu.** Metody identifikacii i autentifikacii polzovatelja pri dostupe k fajlovym objektam. *Vestnik kompjuternyh i informacionnyh tehnologij*. 2012. N. 10. P. 47—51.

6. **Shcheglov K. A., Shcheglov A. Yu.** Princip i metody kontrolya dostupa k sozdavaemym fajlovym objektam. *Vestnik kompjuternyh i informacionnyh tehnologij*. 2012. N. 7. P. 43—47.
7. **Shcheglov K. A., Shcheglov A. Yu.** Prakticheskaja realizacija diskrecionnogo metoda kontrolya dostupa k sozdavaemym fajlovym objektam. *Vestnik kompjuternyh i informacionnyh tehnologij*. 2013. N. 4. P. 43—49.
8. **Shcheglov K. A., Shcheglov A. Yu.** Zashhita ot vredonosnyh programm metodom kontrolya dostupa k sozdavaemym fajlovym objektam. *Vestnik kompjuternyh i informacionnyh tehnologij*. 2012. N. 8. P. 46—51.
9. **Shcheglov K. A., Shcheglov A. Yu.** Zashhita ot atak so storony prilozhenij, nadeljaemyh vredonosnymi funkccijami. Modeli kontrolya dostupa. *Voprosy zashhity informacii*. 2012. Iss. 99, N. 4. P. 31—36.
10. **Shcheglov K. A., Shcheglov A. Yu.** Kontrol dostupa k statichnym fajlovym objektam. *Voprosy zashhity informacii*. 2012. Iss. 97, N. 2. P. 12—20.

УДК 004, 378.1

Т. Н. Пастушак, канд. техн. наук, доц., зав. каф., проректор,
С. С. Соколов, канд. техн. наук, доц., нач. управления, **Н. М. Ковальногова**, аспирант,
А. В. Маньковская, специалист по дистанционным образовательным технологиям,
ФГБОУ ВПО "ГУМРФ имени адмирала С. О. Макарова"

Приоритетные направления развития электронной информационно-образовательной среды отраслевого университетского комплекса

Рассмотрен опыт вузов в построении электронной информационно-образовательной среды, использовании электронного обучения и применения дистанционных образовательных технологий. Приведено описание и методологическая взаимосвязь основных компонентов электронной информационно-образовательной среды отраслевого вуза. Представлено описание структуры и программного комплекса, служащего основой для различных образовательных систем. Рассмотрены уровни взаимодействия преподавателей и обучающихся в электронном обучении. Перечислены возможности и преимущества компонентов электронной информационно-образовательной среды. Приведены примеры использования и развития дистанционных образовательных технологий в учебном процессе, а также способы внедрения инновационных технологий в учебный процесс.

Ключевые слова: электронная информационно-образовательная среда, образовательный портал, система дистанционного обучения, электронное обучение, комплексная информационно-аналитическая система, комплексная электронно-библиотечная система

Введение

В настоящее время одной из актуальных задач и направлений инновационной деятельности образовательного учреждения высшего образования является построение, обеспечение функционирования и активное развитие информационно-образовательной среды вуза.

Внедрение такой среды позволяет создать единое информационное пространство, повысить качество образовательного процесса, эффективно использовать информационные, научно-образовательные ресурсы, обеспечить электронное взаимодействие и информационную поддержку деятельности всех участников образовательного и научного процессов.

Несомненным лидером в области внедрения инновационных образовательных технологий является Московский государственный университет экономики, статистики и информатики. В течение многих лет и в Петрозаводском государственном университете системно ведутся работы по созданию информационно-образовательной среды университетского комплекса. Ряд вузов, таких как Воронежский государственный университет, Пензенская государственная технологическая академия, Томский политехнический университет и Перм-

ский государственный технический университет и другие добились значительных успехов в этой деятельности. Изучение опыта других вузов является обязательным условием успешной и эффективной работы по созданию информационно-образовательной среды вуза.

В то же время отраслевое образование, например обеспечивающее подготовку членов экипажей морских судов, имеет свои специфические особенности. При подготовке плавсостава отраслевой вуз должен обеспечивать выполнение требований не только Министерства образования и науки Российской Федерации, но и требования Международной конвенции о подготовке и дипломировании моряков и несении вахты. Обучение специалистов плавсостава необходимо вести в условиях, близких к условиям работы на море, сочетать фундаментальную теоретическую подготовку с практически и лабораторными занятиями, производственными практиками. Практическая составляющая подготовки плавсостава обязательно должна включать использование различных тренажеров.

Государственный университет морского и речного флота имени адмирала С. О. Макарова (Университет) — современный научно-образовательный

комплекс, ведущий подготовку специалистов для морского и речного флота.

Представляя собой крупный отраслевой территориально распределенный комплекс, имеющий в своем составе сеть учебных городков на территории Санкт-Петербурга и региональную сеть филиалов, университет создал, активно развивает и использует электронную информационно-образовательную среду (ЭИОС).

В соответствии с требованиями законодательства в сфере образования электронная информационно-образовательная среда вуза включает в себя электронные информационные ресурсы, электронные образовательные ресурсы, совокупность информационных технологий, телекоммуникационных технологий, соответствующих технологических средств и обеспечивает освоение обучающимися образовательных программ в полном объеме независимо от местонахождения обучающихся [1].

Многокомпонентная система ЭИОС университета включает в себя комплексную информационно-аналитическую систему управления (КИАСУ), систему дистанционного предоставления контента (образовательный портал и видео-портал) и дистанционного взаимодействия (система дистанционного обучения СДО "ФАРВАТЕР"), комплексную электронную библиотечную систему (КЭБС). Перечисленные компоненты ЭИОС являются приоритетными направлениями ее развития [2].

Комплексная информационно-аналитическая система управления

КИАСУ ГУМРФ является автоматизированной основой для функционирования электронной информационно-образовательной среды. В качестве программной платформы системы была выбрана United University. Первым успешно внедренным модулем этой веб-ориентированной системы стал модуль "Приемная комиссия". Работа в системе осуществляется посредством веб-браузера. United University — это современная технологическая платформа — серверная часть написана на языке Java, в ней используются широко распространенные OpenSource-компоненты (Hibernate, Spring, Tapestry и т. д.), поддерживаются все популярные операционные системы и системы управления базами данных (СУБД). В настоящее время активно развиваются модули "Движение контингента" и "Договора".

Образовательный портал и система дистанционного обучения "ФАРВАТЕР"

Развитие дистанционных образовательных технологий (ДОТ) и электронного обучения осуществляется с использованием двух ресурсов: образовательного портала университета и СДО "ФАРВАТЕР".

Под электронным обучением в соответствии с Федеральным Законом "Об образовании в РФ"

(273-ФЗ от 29 декабря 2012 года) понимается организация образовательной деятельности с применением содержащейся в базах данных и используемой при реализации образовательных программ информации и обеспечивающих ее обработку информационных технологий, технических средств, а также информационно-телекоммуникационных сетей, выполняющих передачу по линиям связи указанной информации, взаимодействие обучающихся и педагогических работников [1].

Под дистанционными образовательными технологиями понимаются образовательные технологии, реализуемые в основном с применением информационно-телекоммуникационных сетей при опосредованном (на расстоянии) взаимодействии обучающихся и педагогических работников [1].

Образовательный портал разработан на платформе 1С-Битрикс — типовом решении для образовательных учреждений и позволяет выполнять требования ФГОС по обеспечению доступа студентов и курсантов к содержанию учебных дисциплин через Интернет. Портал представляет собой комплексную систему, позволяющую, с одной стороны, создать внутренний информационно-коммуникационный ресурс для сотрудников, а с другой стороны, построить закрытую социально-образовательную сеть для обучающихся (студентов и курсантов) и преподавателей, интегрированную в учебный процесс.

Образовательный портал выступает в роли хранилища учебно-методических материалов (УММ), а также позволяет организовать эффективное дистанционное взаимодействие обучающихся и преподавателей посредством технологии Университетской социальной сети.

Структура хранения материалов представлена на рис. 1.

Портал для обучающихся является закрытой системой учебного заведения. Доступ в нее могут получить только обучающиеся (в настоящее время и выпускники), уполномоченные сотрудники и преподаватели учебного заведения, либо привлеченные внешние персоны, которых требуется вовлекать в учебный процесс и жизнь учебного заведения. Зарегистрированному пользователю на главной странице портала предоставлен интересно оформленный



Рис. 1. Структура хранения файлов

удобный интерфейс, который помогает ориентироваться среди набора ссылок и образовательных материалов. Портал оснащен мощным инструментарием для поддержания динамики развития. Систематическое обновление платформы отвечает за эволюцию данного информационного ресурса, что способствует заинтересованности курсантов и студентов, использующих портал в образовательном процессе. Допускается два режима работы портала: пользовательский и административный.

Пользовательский режим образовательного портала обеспечивает:

- удаленный доступ к образовательному контенту 24 часа в сутки;
- систему обмена информацией между преподавателями и обучающимися;
- организацию совместной работы обучающихся;
- работу сервиса новостей и объявлений;
- онлайн-расписание;
- создание персональных файловых хранилищ для каждого пользователя с гибкой настройкой прав доступа;
- видеопортал.

Пользовательский режим портала подразумевает работу портала для определенных групп, таких как "Преподаватели", "Студенты и курсанты", "Редакторы расписаний" и др. Каждой группе присвоены различные права доступа и соответствующий интерфейс, посредством которого организована работа с информационным наполнением.

Группа пользователей "Студенты и курсанты" имеют доступ к учебно-методическим материалам по дисциплинам, могут отчитываться за выполнение заданий дистанционно, смотреть оценки и получать комментарии преподавателей, обмениваться сообщениями и файлами с другими обучающимися, совместно работать над документами, а также смотреть расписание занятий и экзаменационно-лабораторных сессий.

"Преподаватели" с помощью портала могут организовывать как индивидуальное, так и групповое общение с обучающимися, пополнять базу знаний по учебным дисциплинам, сохранять свои наработки, иметь быстрый доступ к спискам обучающихся и их контактам, старостам групп, создавать опросы.

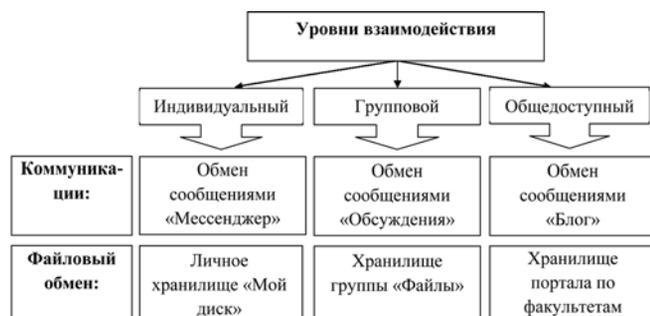


Рис. 2. Структура документооборота

Существуют три уровня обмена файлами между преподавателями и обучающимися: индивидуальный, групповой и общедоступный. Для каждого из уровней организованы разграниченные сервисы для организации общения и передачи файлов. На рис. 2 изображена структура уровней документооборота.

В режиме "Пользователь" организована информационно-поисковая система, которая упрощает навигацию по базе учебно-методических материалов. Портал предоставляет полную информацию о структуре вуза, нормативных документах и образовательных программах университета. Помимо этого на портале размещены ссылки на другие образовательные справочные и электронно-библиотечные ресурсы Интернет, например видеотека [elementy.ru], проект "Лекториум ТВ", "Знания NalandaU".

Одной из особенностей образовательного портала является компонент Видеопортал. На Видеопортале представлены образовательные видеоресурсы университета (видеолекции, видеолaborаторные работы, учебные фильмы и т. д.) и общедоступные видеоресурсы сети Интернет, включающие ссылки на лекции и видео основных образовательных видеопорталов мира.

Как упоминалось ранее, портал отвечает всем требованиям эволюционного развития в соответствии с поддержанием заинтересованности основных групп пользователей. В связи с этим существует модульность, которая позволяет разработчикам модифицировать и расширять функциональность портала. В настоящее время в промышленной эксплуатации находится модуль "Онлайн-расписание" Образовательного портала университета. Модуль позволяет на основе заданных параметров в режиме онлайн формировать таблицу расписаний. Удобство пользователя модуля обеспечено фильтром с широким набором параметров: факультет, группа, преподаватель, аудитория.

Модуль оснащен двумя дополнительными функциями:

- направление на печать выбранного расписания;
- выгрузка выбранного расписания в отдельный файл.

Большая часть функционала Образовательный портал скрыта от пользователей. Поддержкой работоспособности портала занимаются администраторы. В администраторском режиме доступно:

— управление структурой учебного заведения и основными сущностями (факультеты/кафедры, специальности, группы, предметы);

— создание отдельных разделов портала по факультетам/кафедрам, делегирование прав на управление ими и размещение УММ;

— размещение необходимой документации для обучающихся и преподавателей в общее хранилище документов, управление правами доступа;

— обработка заявок обучающихся и преподавателей на регистрацию, импортирование новых спи-

сков обучающихся в портал, формирование учебных групп;

- осуществление обратной связи, управление потоком обращений и контроль их исполнения сотрудниками учебного заведения;

- обновление и модификация модулей под нужды пользователей.

Структурой программной системы Образовательного портала является Bitrix Framework (англ. *framework* — каркас, структура) — это созданная на основе PHP платформа для разработки веб-приложений. Платформа Bitrix Framework — технологическое ядро для управления порталом. Продукт может эффективно работать на любой UNIX или Windows-системе. В качестве базы данных используется MySQL — свободная реляционная система управления базами данных. Она обладает полным функционалом, необходимым для поддержки и развития образовательного портала. Для размещения проектов, созданных на основе этой базы данных, можно использовать любой классический UNIX-хостинг с поддержкой PHP и MySQL [4].

Конфигурация образовательного портала состоит из следующих компонентов: "Контент", "Сервисы", "Аналитика" и "Настройки".

"Контент" описывает скрытую для пользователя структуру построения портала. Здесь содержится основное наполнение страниц портала и происходит прямое управление файловым хранилищем, организованным посредством СУБД.

Раздел "Сервисы" реализует прикладные возможности образовательного портала — модули. Каждый модуль отвечает за управление определенными элементами и параметрами портала: информационным наполнением и структурой, форумами, рассылкой, распределением прав между группами пользователей, сбором статистики посещений, учебными курсами. Здесь администраторам доступна модификация модулей, организация и настройка их работы.

Аналитический компонент "Аналитика" позволяет накапливать и анализировать все основные показатели работы портала, получать объективную численную статистику для анализа, вести детальный учет эффективности внешних посещений ресурса. "Аналитика" дает администраторам возможность:

- вести учет хитов, хостов, сессий, посетителей сайта;
- регистрировать все ссылающиеся сайты;
- анализировать посещения разделов и документов сайта;
- вести учет посетителей по дням с выделением постоянной аудитории сайта;
- отслеживать все сессии одного посетителя между посещениями сайта;
- регистрировать события, происходящие на сайте и за его пределами;
- автоматически чистить базу данных статистики в соответствии с настройками.

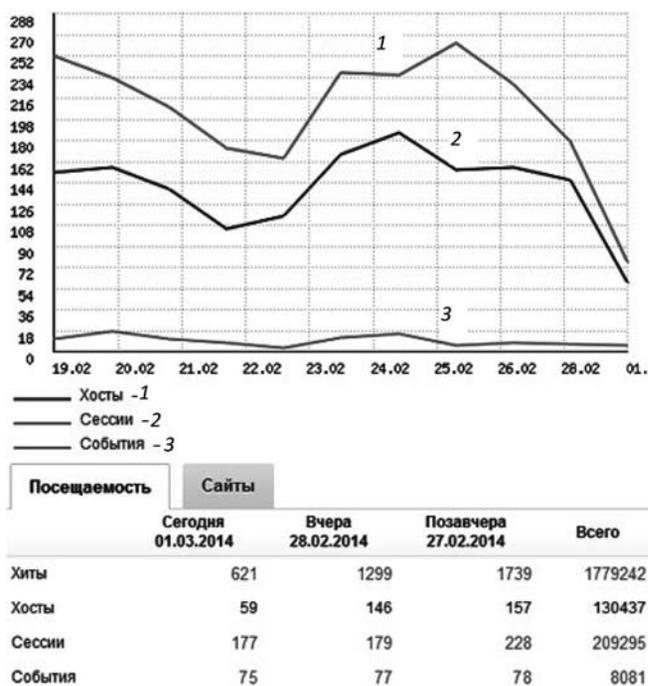


Рис. 3. Статистика хостов, сессий и событий

На рис. 3 представлен пример сервиса "Аналитика" — статистика хостов, событий и сессий. Сутью событий являются переходы по ссылкам на веб-ресурсе, под термином хост понимается доменное имя или IP-адрес, посредством которого можно обратиться к portalу, сессия открывается в момент захода на сайт и заканчивается при закрытии браузера.

В разделе "Настройки" доступно управление безопасностью портала, пользователями и поисковой системой, формирование логической структуры, разработка шаблонов отображения. Здесь осуществляется мониторинг производительности, резервное копирование и непосредственная корректировка работы системы с помощью командной PHP-строки.

Основное назначение другого образовательного ресурса СДО "ФАРВАТЕР" — учебная внеаудиторная самостоятельная работа обучающихся (абитуриентов, студентов, курсантов, аспирантов и т. д.).

СДО "ФАРВАТЕР" разработана на платформе Moodle (*Modular Object-oriented dynamic learning environment*), которая соответствует международному стандарту SCORM, обеспечивающему совместимость компонентов и возможность их многократного использования.

СДО "Фарватер" — модульная объектно-ориентированная динамическая среда обучения представляет собой LMS (*learning management system*) — систему управления обучением, основанную на Интернет-технологиях.

Как элемент электронной информационно-образовательной среды университета, СДО "ФАРВАТЕР" предоставляет широкий спектр возможностей для преподавателей по созданию электронных курсов:



Рис. 4. Схема сервисов СДО "ФАРВАТЕР"

разработка лекций и банка тестовых заданий; дистанционный контроль знаний обучающихся; организация тестирования и анкетирования, размещение учебно-методической документации.

Схема сервисов системы дистанционного обучения представлена на рис. 4.

Курсанты и студенты в СДО могут изучать дисциплины по электронным учебникам и пособиям, выполнять задания, проходить обучающее и контрольное тестирование. При создании вопросов преподавателям доступен обширный инструментарий: выбор типа вопроса из двенадцати существующих, гибкая система настроек тестов, ограничение времени и попыток тестирования. Важной особенностью является настройка доступа к тесту определенным IP-адресам, что позволяет провести контрольное тестирование для студентов и курсантов в определенной аудитории. Результаты тестирования хранятся в удобной форме с указанием данных

обучающегося и подробной отчетностью прохождения теста (рис. 5).

Образовательный портал и СДО "ФАРВАТЕР" — сложные и постоянно развивающиеся системы. Доступ к ним осуществляется посредством единой универсальной программы — веб-браузера. Заинтересованность целевой аудитории (студентов, курсантов, преподавателей) постоянно растет. На рис. 6 (см. третью сторону обложки) приведен график статистических данных посетителей портала и СДО за период сентябрь 2013 — февраль 2014, на котором явно прослеживается увеличение численности постоянных и новых пользователей.

В целях более эффективного использования и развития ресурсов ЭИОС университета дважды в год проводятся курсы повышения квалификации профессорско-преподавательского состава университета, его филиалов и колледжа по программе "Использование инновационных образовательных технологий в учебном процессе".

Участники курсов выступают в роли дистанционно обучающихся. Для информационной поддержки создаются учебные группы на портале, курсы и опросы в СДО и размещаются методические материалы. Слушатели курсов, погружаясь в электронно-образовательную среду, могут оценить достоинства работы с веб-ресурсами со стороны преподавателя и студента/курсанта.

Обучение преподавателей проходит по основным пяти темам: использование образовательного портала для взаимодействия с обучающимися; создание электронного курса в СДО "ФАРВАТЕР";

организация работы тьютора со студентами в университете; использование интерактивных средств в образовательном процессе; разработка видеолaborаторных работ и т. п. Также в процессе занятий слушатели курсов знакомятся с технологиями разработки видеопрезентаций и видеоинструкций, организации телемостов со студентами посредством онлайн связи, работы в электронных библиотечных системах и с КЭБС. В процессе обучения на курсах преподаватели создают новые учебные мультимедийные материалы, которые пополняют базы образовательных ресурсов университета.

Ежегодно среди профессорско-преподавательского состава университета, его филиалов и колледжа про-

		Скачать данные таблицы как: текстовый файл, разделенный запятыми									
	Имя / Фамилия	Адрес электронной почты	Тест начат	Завершено	Затраченное время	В. 1 /20,00	В. 2 /20,00	В. 3 /20,00	В. 4 /20,00	В. 5 /20,00	Оценка/100
	Имя Фамилия Просмотр попытки	_____@mail.ru	19 сентября 2013 09:56	20 сентября 2013 09:32	6 дн. 23 час.	20,00 ✓	20,00 ✓	20,00 ✓	20,00 ✓	7,50 ✓	87,50
	Имя Фамилия Просмотр попытки	_____@mail.ru	25 сентября 2013 12:15	25 сентября 2013 12:38	23 мин. 6 сек.	6,67 ✓	20,00 ✓	0,00	4,67 ✓	2,50 ✓	33,83
	Имя Фамилия Просмотр попытки	_____@mail.ru	25 сентября 2013 12:16	25 сентября 2013 12:55	38 мин. 40 сек.	20,00 ✓	13,33 ✓	20,00 ✓	6,67 ✓	5,00 ✓	65,00
	Имя Фамилия Просмотр попытки	_____@yandex.ru	25 сентября 2013 12:16	25 сентября 2013 12:48	31 мин. 45 сек.	20,00 ✓	13,33 ✓	20,00 ✓	13,33 ✓	10,00 ✓	76,67
	Имя Фамилия Просмотр попытки	_____@mail.ru	25 сентября 2013 12:16	1 октября 2013 23:11	6 дн. 10 час.	20,00 ✓	20,00 ✓	20,00 ✓	20,00 ✓	10,00 ✓	90,00
	Имя Фамилия Просмотр попытки	_____@mail.ru	25 сентября 2013 12:16	25 сентября 2013 12:50	33 мин. 40 сек.	20,00 ✓	13,33 ✓	20,00 ✓	13,33 ✓	2,50 ✓	69,17
	Имя Фамилия Просмотр попытки	_____@mail.ru	25 сентября 2013 12:17	25 сентября 2013 12:58	41 мин. 35 сек.	20,00 ✓	20,00 ✓	20,00 ✓	20,00 ✓	2,50 ✓	82,50
	Имя Фамилия Просмотр попытки	_____@gmail.com	25 сентября 2013 12:21	-	не закончен	20,00	20,00	-	13,33	-	-
Общее среднее						18,21 (56)	17,55 (56)	18,21 (56)	16,89 (56)	5,49 (56)	76,23 (57)

Рис. 5. Таблица результатов тестирования в СДО "ФАРВАТЕР"

водится конкурс на лучший электронный курс, созданный в СДО «ФАРВАТЕР». Целью конкурса является развитие инновационных образовательных технологий в образовательном процессе университета и использование дистанционных технологий для повышения качества подготовки специалистов. Задачи конкурса: вовлечение преподавателей университета в работу по проектированию и разработке качественно новых мультимедийных интерактивных учебных материалов; внедрение передовых технологий и стандартов обучения.

Комплексная электронная библиотечная система

В соответствии с требованиями Федерального закона № 273-ФЗ: "В организациях, осуществляющих образовательную деятельность, в целях обеспечения реализации образовательных программ, формируются библиотеки, в том числе цифровые (электронные) библиотеки, обеспечивающие доступ к профессиональным базам данных, информационным справочным и поисковым системам, а также иным информационным ресурсам" [1].

КЭБС состоит из двух контуров: Интернет и интранет. В рамках Интернет-контура на официаль-

ном сайте университета имеется свободный доступ к электронному каталогу библиотечного комплекса. В рамках интранет-контура реализована внутренняя электронная библиотека университета, в которой есть возможность доступа к полнотекстовым версиям всех учебно-методических материалов, издаваемых в университете [3].

На рис. 7 представлена архитектура электронно-образовательной среды университета.

Одной из перспектив развития ЭИОС является обеспечение сквозной идентификации пользователей, которая заключается в обеспечении автоматического доступа пользователя к ресурсам ЭИОС при идентификации на Образовательном портале.

В этом случае необходимо дописать модуль управления безопасностью, включающий в себя возможности доступа к перечисленным выше ресурсам, основанным на разных платформах, что является темой отдельной исследовательской и программно-технологической деятельности. В методологическую основу модуля должна входить классическая матрица прав доступа, имеющая в первом столбце категории ресурсов, в первой строке — группы пользователей, а на пересечении — уровень доступа: чтение/запись/изменение.

Заключение

Таким образом, в условиях реформирования профессионального образования и смены основной парадигмы образования в сторону новых методов обучения, развитие описанных в статье инновационных образовательных технологий, несомненно, обеспечит повышение качества подготовки специалистов в современном университетском комплексе.

Список литературы

1. **Федеральный** закон от 29.12.2012 № 273-ФЗ (ред. от 25.11.2013) "Об образовании в Российской Федерации".
2. **Пастушак Т. Н., Соколов С. С., Ковальникова Н. М.** Инновационные образовательные технологии современного университетского комплекса // Сборник трудов VII Санкт-Петербургского конгресса "Профессиональное образование, наука, инновации в XXI веке". Санкт-Петербург, 2013. С. 166—167.
3. **Пастушак Т. Н., Соколов С. С.** Основные направления формирования единого библиотечно-информационного пространства отраслевого университетского комплекса // Сборник трудов Юбилейной XIII Санкт-Петербургской международной конференции "Региональная информатика (РИ—2012)". Санкт-Петербург, 24—26 октября 2012 г. СПб.: СПОИСУ, 2013, Санкт-Петербург, 2012. С. 242—247.
4. **Онлайн-документация** пользователей 1С-битрикс. URL: <http://www.1c-bitrix.ru/products/> (дата обращения: 04.03.2014).
5. **Образовательный портал** ФГБОУ ВПО "ГУМРФ имени адмирала С. О. Макарова". URL: <http://edu.gumrf.ru/> (дата обращения: 04.03.2014).
6. **СДО "ФАРВАТЕР"** ФГБОУ ВПО "ГУМРФ имени адмирала С. О. Макарова". URL: sdo.gumrf.ru/ (дата обращения: 04.03.2014).
7. **Комплексная электронно-библиотечная система.** Электронный каталог. URL: <http://www.e-library.gumrf.ru/> (дата обращения: 04.03.2014).

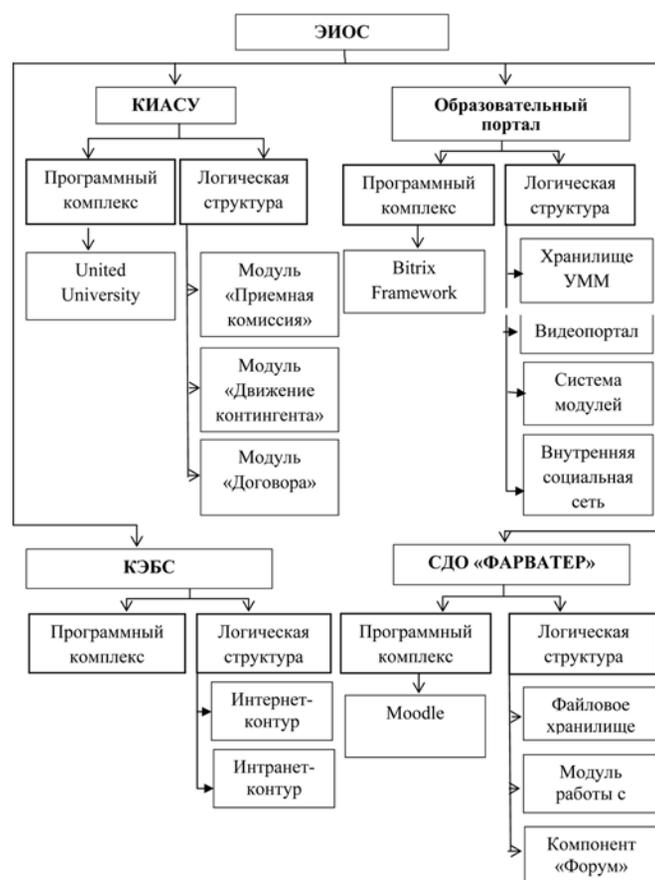


Рис. 7. Архитектура электронно-образовательной среды университета

T. N. Pastuschak, Associated Professor, Head of department, Provost,
S. S. Sokolov, Associated Professor, Head of Department, **N. M. Kovalnogova**, Graduate Student,
A. V. Mankovskaya, Expert on Distance Learning Technologies
Admiral Makarov State University of Maritime and Inland Shipping, e-mail: Sokolov@gumrf.ru

Priority Areas for the Development of Electronic Informational and Educational Environment of the University Complex

The article describes the experience of various universities in building electronic information-educational environment, using e-learning and the use of distance learning technologies. The description and methodological relationship major components of the electronic information industry and educational environment of the university. The description of the structure and program complex that serves as a basis for various educational systems. Examined the levels of interaction of teachers and students in e-learning. Lists the features and advantages of the components of the electronic educational environment. Examples of the use and development of distance learning technologies in the learning process, as well as ways to introduce innovative technologies into the learning process.

Keywords: *electronic information and educational environment, educational portal system of distance learning, e-learning, integrated information analysis system, integrated electronic library system*

References

1. **Federal'nyj zakon** ot 29.12.2012 N 273-FZ (red. ot 25.11.2013) "Ob obrazovanii v Rossijskoj Federacii".
2. **Pastushhak T. N., Sokolov S. S., Koval'nogova K. M.** Innovacionnye obrazovatel'nye tehnologii sovremenogo universitetskogo kompleksa. *Sbornik trudov. VII Sankt-Peterburgskij kongress "Professional'noe obrazovanie, nauka, innovacii v XXI veke"*, Sankt-Peterburg, 2013. P. 166—167.
3. **Pastushhak T. N., Sokolov S. S.** Osnovnye napravlenija formirovanija edinogo bibliotечно-informacionnogo prostranstva otraslevogo universitetskogo kompleksa. *Jubilejnaja XIII Sankt-Peterburgskaja mezhdunarodnaja konferencija "Regional'naja informatika.*

(*RI—2012*)". Sankt-Peterburg, 24—26 okt. 2012 g.: Trudy konferencii \ SPOISU. SPb, 2013. Sankt-Peterburg, 2012. P. 242—247.

4. **Onlajn dokumentacija pol'zovatelej IS-bitriks.** URL: <http://www.1c-bitrix.ru/products/> (data obrashhenija: 04.03.2014).

5. **Obrazovatel'nyj portal FGBOU VPO "GUMRF imeni admirala S.O. Makarova"**. URL: <http://edu.gumrf.ru/> (data obrashhenija: 04.03.2014).

6. **SDO "FARVATER" FGBOU VPO "GUMRF imeni admirala S.O. Makarova"** URL: sdo.gumrf.ru/ (data obrashhenija: 04.03.2014).

7. **Kompleksnaja jelektronno-bibliotечноnaja sistema.** Jelektronnyj katalog. URL: <http://www.e-library.gumrf.ru/> (data obrashhenija: 04.03.2014).

УДК 004.02:378

А. А. Смагин, д-р техн. наук, проф., заф. каф., e-mail: AlSmagin@ulsu.ru,
С. В. Липатова, канд. техн. наук, доцент, e-mail: dassegel@mail.ru,
О. Л. Курилова, ст. преподаватель, e-mail: oxana197208@rambler.ru,
Ульяновский государственный университет, г. Ульяновск

Метод численной оценки профессиональных и общекультурных компетенций студента и выпускника вуза

Описан разработанный метод количественной оценки общекультурных и профессиональных компетенций. Представлен способ формализации и измерения компетенций и их элементов студента или выпускника вуза. Продемонстрирована взаимосвязь дисциплин учебного плана на основе входящих и исходящих компетенций в рамках компетентностного подхода ФГОС ВПО.

Ключевые слова: *компетенция, учебный план, оценка общекультурных компетенций, оценка профессиональных компетенций*

Введение

Одним из основных путей модернизации российского образования является компетентностный подход, который определяет цели и результаты образования посредством ключевых компетенций.

ФГОС ВПО [1] разделяет компетенции на общекультурные (ОК) и профессиональные (ПК), причем ОК формируются в результате активной общественной работы, различных тренингов, тестов, моделирующих и ролевых игр. ПК получают в про-

цессе усвоения практических и теоретических знаний по специальным дисциплинам. Вопросам преподавания дисциплин посвящено множество работ, в которых детально рассмотрен данный процесс, а вот методам количественной оценки, позволяющим автоматизировать процесс контроля усвоения компетенций и соответствия уровня компетенций требованиям работодателей, уделено значительно меньше внимания и на данный момент нет единой системы количественной оценки общекультурных и профессиональных компетенций выпускника [2].

Несмотря на достаточно полную разработанность теоретической базы компетентного подхода в образовательном процессе, существует проблема создания и внедрения автоматизированных систем мониторинга и оценки компетенций выпускников вуза [3]. Например, в работе [4] описан инновационный метод оценки компетенций с использованием уровней освоения компетенции, но в данном методе не предлагается практическая реализация, т. е. построение автоматизированной системы.

Для выбора способа оценки уровня сформированности компетенций были проанализированы работы различных авторов (Б. Блума, Д. Бокка, Дж. Гилфорда, В. П. Беспалько, В. П. Симонова [5], С. Торпа [6], Дж. Клиффорда [6] и др.). В работах [7–9] используются теории В. П. Симонова, С. Торпа, Дж. Клиффорда для оценки сформированности различных видов компетенций в общеобразовательных и профессиональных учреждениях, но в этих работах не учитывается последовательность изучения дисциплин на основе входящих и исходящих компетенций, не предлагается точная количественная оценка компетенций, а только определенный диапазон, не учитывается разбиение компетенций на ОК и ПК согласно ФГОС.

Проблема измерения уровня компетентности студентов должным образом не стандартизована. Поэтому целью данной работы является разработка метода оценки компетенций выпускника вуза, который позволит обеспечить: мониторинг элементов компетенций (знаний, умений и навыков), мониторинг оценки компетенций для выпускников и студентов вуза, возможность управления процессом обучения (изменение последовательности изучаемых дисциплин с учетом исходящих и входящих элементов компетенций), обобщение результатов.

Для достижения этой цели необходимо решить ряд следующих задач: формализация компетенций и их элементов, взаимосвязей компетенций и учебного плана, разработка метода количественной оценки ОК, ПК.

Взаимосвязь компетенций и дисциплин

Взаимосвязь дисциплин и компетенций прописывается в учебном плане. Учебный план определяет перечень изучаемых дисциплин, порядок их изучения и объем. Каждая дисциплина для начала изучения требует определенной подготовки от студента и уровня владения компетенциями. После успешного изучения дисциплины уровень повышается, студент осваивает дополнительные компетенции (взаимосвязь элементов компетенций и дисциплины на примере дисциплины "Метрология, стандартизация и сертификация информационных технологий" представлена на рис. 1, слева требуемые перед началом обучения элементы компетенций, справа — сформировавшиеся после обучения).

Не все дисциплины связаны между собой явно, порядок усвоения знаний по дисциплинам, а следовательно, и элементов компетенций, определяется кафедрами вуза при составлении рабочего учебного плана. Взаимосвязь дисциплин можно отследить через входящие (требуемые перед обучением) и исходящие (сформированные в процессе обучения) компетенции. На рис. 2 показан фрагмент ориентированного графа (для специальности "Информационные системы"), на котором для каждой дисциплины отображены предшествующие и следующие за ней дисциплины.

Данную взаимосвязь можно описать, используя аппарат теории множеств. Пусть имеются множество дисциплин D и множество компетенций K .



Рис. 1. Взаимосвязь компетенций и дисциплин на примере дисциплины "Метрология, стандартизация и сертификация информационных технологий"

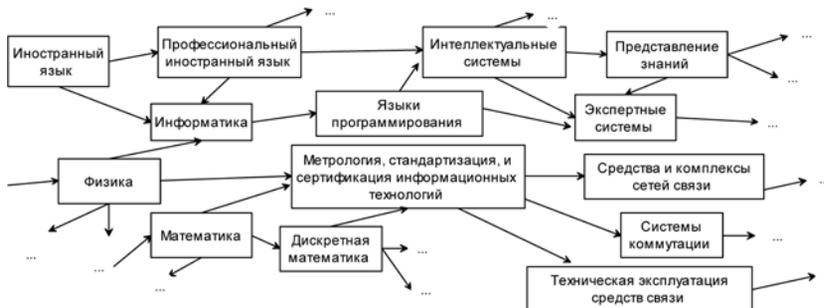


Рис. 2. Фрагмент ориентированного графа взаимосвязи дисциплин на примере специальности "Информационные системы и технологии"

Множество $D = \{D_1, D_2, \dots, D_m\}$, где $D_i = \{Out_{D_i}, Inp_{D_i}\}$ — дисциплина, $i = 1 \dots m$, $Out_{D_i} = \{Th_{O_i}, Pr_{O_i}\}$ — множество теоретических и практических элементов знания, формируемых в процессе изучения дисциплины; $Inp_{D_i} = \{Th_{I_i}, Pr_{I_i}\}$ — множество теоретических и практических элементов знания, необходимых для усвоения дисциплины.

Множество $K = \{K_1, K_2, \dots, K_n\}$, где $K_j = \{Th_j, Pr_j, C_j\}$ — компетенция $j, j = 1 \dots n$, Th_j — множество элементов теоретического знания, Pr_j — множество практических навыков, C_j — множество профессиональных качеств личности. Так как составляющими компетенций являются знания, умения и навыки, например, для ПК-26 — "Готовность использовать математические методы обработки, анализа и синтеза результатов профессиональных исследований", характерны следующие компоненты:

ПК-26 =

= { $\left. \begin{array}{l} \text{знание основ теории вероятности и математической статистики;} \\ \text{знание классических методов решения нелинейных уравнений;} \\ \text{знание структуры погрешностей решения вычислительных задач;} \\ \text{знание различных методов решения систем линейных уравнений;} \\ \text{и т. д.} \end{array} \right\}$

Каждая дисциплина D_i связана с дисциплиной D_y из множества D . Имеется множество связей $S^y = \{S_1, S_2\}$, через которые отыскиваются наиболее близкие друг другу дисциплины, где связь S_1 связывает дисциплины через знания, а S_2 связывает дисциплины через умения и навыки.

Связь между D_i и K_j существует, если $Th_{O_i} \cap Th_j \neq 0$ или $Th_{I_i} \cap Th_j \neq 0$ или $Pr_{O_i} \cap Pr_j \neq 0$ или $Pr_{I_i} \cap Pr_j \neq 0$.

Связь между D_i и D_y определяется следующим образом:

- если $Inp_{D_y} \cap Out_{D_i} \neq 0$ и $Inp_{D_i} \cap Out_{D_y} = 0$, то D_i предшествует D_y (и наоборот);
- если $Inp_{D_y} \cap Out_{D_i} \neq 0$ и $Inp_{D_i} \cap Out_{D_y} \neq 0$, то D_i изучается в одном семестре с D_y ;
- если $Inp_{D_y} \cap Out_{D_i} = 0$ и $Inp_{D_i} \cap Out_{D_y} = 0$, то D_i не связаны с D_y .

Аналогично можно определить взаимосвязь компетенции и дисциплины.

Метод количественной оценки компетенций

Существует много методов, позволяющих провести оценку таких составляющих компетенций, как знания, умения, навыки, сформированность компетенций студентов. Эти методы отличаются не только способом взаимодействия с оцениваемым, но и трудовыми и временными затратами, имеющейся информационной поддержкой и т. д. (табл. 1).

Перечисленные методы оценивают элементы компетенции, но ни один не позволяет всесторонне, полно и комплексно оценить все компетенции.

Количественная оценка общекультурных компетенций

Выпускники по направлению подготовки "Информационные системы и технологии" с квалификацией "бакалавр", согласно Федеральному государственному образовательному стандарту [1] должны обладать определенными ОК и ПК. Для определения уровня компетенций конкретного выпускника необходимо подобрать наиболее оптимальный способ оценки каждой из компетенций (табл. 2).

Алгоритм оценки компетенций заключается в том, что первоначально студент должен пройти тестирование для оценки ОК, которые формировались при обучении в школе. Владение такими компетенциями как ОК-1, ОК-2, ОК-3 и т. д. позволяет более эффективно приобретать ПК и пользоваться ими в профессиональной деятельности. На втором шаге оцениваются ПК посредством анализа средней и итоговой оценки по предметам, причем для каждой дисциплины также учитываются отдельные элементы ОК и ПК, определенные экспертами (преподавателями) и коэффициенты уровня освоения компетенций.

Суть оценки ОК заключается в определении уровня владения студентами определенными личностными качествами и социальными навыками, такими как уровень интеллекта, работа в коллективе и т. п.

Таблица 1

Классификация методов оценки компетенций [10, 11]

Метод	Групповой/ Индивидуальный	Участие тестируемого	Реализация в виде ПО	Временные ресурсы	Экономические затраты	Участие эксперта
Оценочно-рейтинговая система	Индивидуальный	—	+	Долгосрочные	+	+
Биографическое анкетирование	Индивидуальный	+	—	Краткосрочные	—	—
Интервью	Индивидуальный	+	—	Краткосрочные	+	+
Тесты	Индивидуальный	+	+	Краткосрочные	+	—
Кейс-метод	Индивидуальный	+	—	Краткосрочные	+	+
Деловые игры	Групповой	+	—	Краткосрочные	+	+
Моделирующие игры	Групповой	+	+	Краткосрочные	+	+
Рольевые игры	Групповой	+	—	Краткосрочные	+	+
Групповая дискуссия	Групповой	+	—	Краткосрочные	+	+
Портфолио	Индивидуальный	—	+	Долгосрочные	—	—
Метод развивающейся кооперации	Групповой	+	—	Краткосрочные	+	+
Проектный метод	Индивидуальный	+	+	Краткосрочные	+	—
Мозговой штурм	Групповой	+	—	Краткосрочные	+	+

Компетенции, методы и шкалы оценки

Компетенции	Описание элементов компетенций	Метод оценки	Шкалы	Мин/Макс	Шаг
Общекультурные					
ОК-1	Владение культурой мышления Способность к обобщению, анализу, восприятию информации Способность к постановке цели и выбору путей ее достижения	Тест структуры интеллекта Амтхауэра [1] Методика Кэттелла [2], [3] Тест смысловых ориентаций Леонтьева [4]	Интервальная	Мин=20 Макс=80 Мин=1 Макс=10 Мин=1 Макс=7	
ОК-2	Умение логически верно и аргументированно строить устную и письменную речь	Методика Кэттелла [2], [3]		Мин=1 Макс=10	
ОК-3 ...	Готовность к кооперации с коллегами, работе в коллективе	Методика Е. П. Ильина и П. А. Ковалева "Личностная агрессивность, и конфликтность" [6], [7] Уровень мотивации достижения (Мехрабиан) [5]		Мин=0 Макс=10 Мин=30 Макс=210	
Профессиональные					
ПК-1	Способность проводить предпроектное обследование (инжиниринг) объекта проектирования, системный анализ предметной области, их взаимосвязей	Суммарная оценка по предметам	1. Порядковая "2, 3, 4, 5" 2. Номинальная "зачет/ незачет"	Мин=2 Макс=5	1
ПК-2	Способность проводить техническое проектирование (реинжиниринг)		1. Порядковая "2, 3, 4, 5" 2. Номинальная "зачет/незачет"	Мин=2 Макс=5	1
ПК-3 ...	Способность проводить рабочее проектирование		1. Порядковая "2, 3, 4, 5" 2. Номинальная "зачет/незачет"	Мин=2 Макс=5	1

Методики оценки личностных качеств и социальных навыков разработаны специалистами в области психологии и социологии [12]. Существует множество психодиагностических методик и тестов [13]—[19], поэтому среди них отбирают наиболее формализованные, имеющие готовые инструменты компьютерной диагностики, которые можно внедрить в информационную систему, оценивающую все компетенции.

Существующие методики были разбиты на классы, соответствующие определенным компетенциям, в этих классах были определены методы, позволяющие дать количественную оценку компетенций. Результат анализа представлен в табл. 2, откуда следует, что, например для ОК-1 существуют три метода оценки, для ОК-3 — два метода и т. д.

Для оценки ОК представляется целесообразным использовать методики и тесты (см. табл. 2), которые автоматизированы и представлены множеством программных средств, например Экспериментально-диагностический комплекс (ЭДК) [20], Psychometric Expert [21], Maintest 4 [22]. Результатом выполнения тестов являются численные оценки элементов (составляющих) компетенций.

Результатом применения перечисленных методов оценки является набор параметров, характеризующих отдельные свойства (знания, навыки) выпускника. Например, для ОК-3 можно применить методики, описанные в табл. 2, со следующими выходными параметрами: вспыльчивость, агрессивность, бескомпромиссность и т. д. (рис. 3).

Для оценки ОК используют разные тесты, методы и методики с различными интервальными шкалами. Необходимо привести их к единой общепринятой шкале оценок от 1 до 5. Для каждой оценки

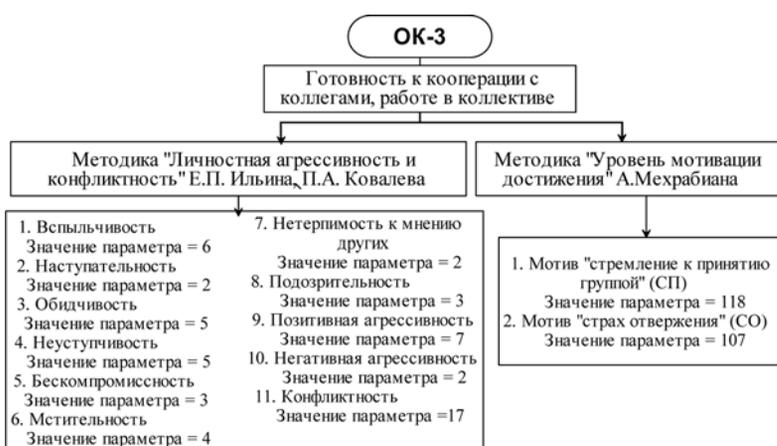


Рис. 3. Методики и выходные параметры ОК-3

Таблица 3

Принадлежность оценок определенным отрезкам

g	a_g	b_g	Итоговая оценка
1	$a_1 = 30$	$b_1 = 30 + 36 = 66$	"1" [30; 66]
2	$a_2 = 67$	$b_2 = 66 + 36 = 102$	"2" [67; 102]
3	$a_3 = 103$	$b_3 = 102 + 36 = 138$	"3" [103; 138]
4	$a_4 = 139$	$b_4 = 174$	"4" [139; 174]
5	$a_5 = 175$	$b_5 = 210$	"5" [175; 210]

по пятибалльной шкале поставим в соответствие определенный интервал из произвольной интервальной шкалы с минимальным значением, равным a , и максимальным значением, равным b . Оценки могут принимать значения $g = 1, \dots, 5$, где $g \in Z$.

Пусть произвольная шкала задана на отрезке $[a; b]$, который разбивается на пять отрезков, где

$$g \in [a_g; b_g]; \Delta = \frac{b-a}{5} \in R;$$

$$g \in \begin{cases} a_g = a; b_g = a + \Delta, \text{ если } g = 1, \\ a_g = b_g + 1; b_g = b_{g-1} + \Delta, \text{ если } \Delta \in Z, \\ 2 \leq g \leq 5, \\ a_g = b_g + 0,1; b_g = b_{g-1} + \Delta, \text{ если } \Delta \in R, \\ \Delta \notin Z, 2 \leq g \leq 5. \end{cases} \quad (1)$$

Например, для оценки ОК-3 используется методика Мехрабиан, где $a = 30$ и $b = 210$. Согласно формулам (1), оценки пятибалльной шкалы принадлежат отрезкам, представленным в табл. 3.

Представим результаты оценки в следующем виде: пусть Rez_{jk} — множество результатов оценки элемента k общекультурной компетенции j , где $j = 1 \dots n, k = 1 \dots h$.

$Rez_{jk} = \{Name_k, OK_j, Nmetod_k, CH_{kr}, Znach_{kr}, Date, Student\}$, где $Name_k$ — название результата оценки элемента k общекультурной компетенции j ; OK_j — название общекультурной компетенции; $Nmetod_k$ — название метода (методики, теста); CH_{kr} — множество выходных параметров метода k ; $Znach_{kr}$ — множество оценок выходных параметров r метода k ; $Date$ — дата проведения оценки общекультурных компетенций, $Student$ — студент, чья компетенция оценивается, $k = 1 \dots h, r = 1 \dots l$.

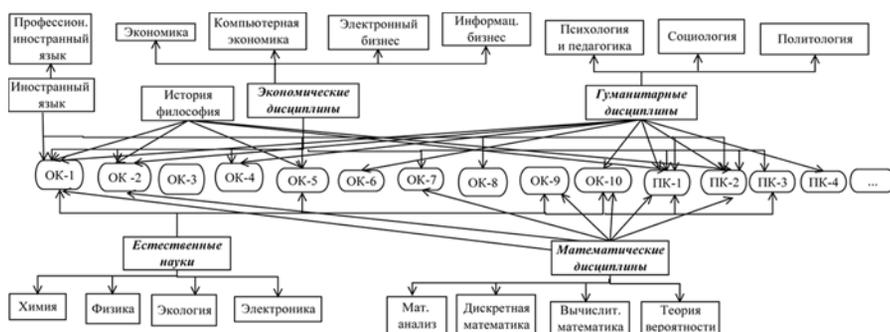


Рис. 4. Фрагмент графа компетенций и дисциплин

После применения метода (методики, теста) $Nmetod_k$ для оценки OK_j получается множество оценок $Znach_{kr}$ определенных выходных параметров (рис. 3). Эти оценки следует привести к единой взвешенной оценке.

Если выходной параметр r метода k точно совпадает с названием элемента компетенции j (см. табл. 2), то $Rez_{jk} = Znach_{kr}$

Если названию элемента k компетенции j соответствует несколько выходных параметров метода k , то

$$Rez_{jk} = \frac{r=1}{l} \sum Znach_{kr} \quad (2)$$

Далее полученному результату Rez_{jk} оценки элемента k компетенции j следует поставить в соответствие определенный интервал по формуле (1) и, следовательно, определенную оценку по пятибалльной шкале.

Поскольку для получения оценки OK_j можно использовать несколько методов (методик, тестов), каждый из которых дает определенный числовой результат Rez_{jk} оценки элемента k компетенции j , то для получения итоговой оценки компетенции j следует использовать формулу

$$OK_j = \frac{k=1}{h} \sum Rez_{jr} \quad (3)$$

Количественная оценка профессиональных компетенций

ПК в учебных заведениях оценивают в процессе обучения. Поэтому можно использовать сведения об успеваемости студента (выпускника) по конкретным дисциплинам для оценки составляющих компетенций. На примере специальности 230400 Информационные системы и технологии (квалификация "Бакалавр"), для выявления связи между дисциплинами и компетенциями был разработан граф, показывающий формирование компетенций в ходе изучения дисциплин. Фрагмент графа представлен на рис. 4.

Для получения общей оценки по ПК необходимо учитывать элементы компетенций, которые изучаются и измеряются в разных дисциплинах. Для формализации связей между элементами компетенций и их оценками предлагается использовать табличное представление (см. ниже табл. 5).

В табл. 4 приведен фрагмент матрицы компетенций $S = (c_{ij})$, размера $m \times n$ [23], использующейся для оценки компетенций, которыми обладает выпускник. Такая матрица соответствует ФГОС ВПО [1] по направлению ИС.

Фрагмент матрицы компетенций для специальности ИС

Дисциплина	ПК-1	ПК-2	ПК-3	ПК-4	ПК-5	ПК-6	ПК-7	ПК-8	ПК-9	ПК-10	ПК-11	ПК-12	...	
Интеллектуальные системы и технологии	$O_{пк1}$	$O_{пк2}$												O
Инструментальные средства ИС	$O_{пк1}$	$O_{пк2}$												O
Инфокоммуникационные системы и сети	$O_{пк1}$	$O_{пк2}$				$O_{пк6}$					$O_{пк11}$			
Методы и средства проектирования ИС	$O_{пк1}$	$O_{пк2}$												
Метрология, стандартизация и сертификация ИТ	$O_{пк1}$	$O_{пк2}$	$O_{пк3}$	$O_{пк4}$										
Безопасность жизнедеятельности	$O_{пк1}$	$O_{пк2}$	$O_{пк3}$		$O_{пк5}$									
Администрирование в ИС и сетях	$O_{пк1}$	$O_{пк2}$				$O_{пк6}$					$O_{пк11}$			
Компьютерная геометрия и графика	$O_{пк1}$	$O_{пк2}$												O
Надежность ИС	$O_{пк1}$	$O_{пк2}$	$O_{пк3}$						$O_{пк9}$	$O_{пк10}$				
Аппаратные средства ЭВМ	$O_{пк1}$	$O_{пк2}$												O
Прикладное ПО ЭВМ и сетей	$O_{пк1}$	$O_{пк2}$												O
...														
Учебная практика	$O_{пк1}$	$O_{пк2}$	$O_{пк3}$		$O_{пк5}$	$O_{пк6}$						$O_{пк12}$		O
Научно-исследовательская работа	$O_{пк1}$	$O_{пк2}$				$O_{пк6}$		$O_{пк8}$	$O_{пк9}$	$O_{пк10}$	$O_{пк11}$	$O_{пк12}$		O
Производственная практика	$O_{пк1}$	$O_{пк2}$	$O_{пк3}$	$O_{пк4}$	$O_{пк5}$			$O_{пк8}$		$O_{пк10}$	$O_{пк11}$			O
Итоговая государственная аттестация	$O_{пк1}$	$O_{пк2}$	$O_{пк3}$	$O_{пк4}$	$O_{пк5}$	$O_{пк6}$	$O_{пк7}$	$O_{пк8}$	$O_{пк9}$	$O_{пк10}$	$O_{пк11}$	$O_{пк12}$		O
Итоговая оценка компетенции	O_1	O_2	O_3	O_4	O_5	O_6	O_7	O_8	O_9	O_{10}	O_{11}	O_{12}	...	$O_{общ}$

Формула численной оценки компетенций, предложенная в работе [4], в данной статье значительно модифицирована следующим образом.

При оценке компетенции j учитываются значения матрицы компетенций, где каждой ячейке c_{ij} соответствует оценка компетенции j по дисциплине i , которая вычисляется по формуле

$$O_{пкij} = \frac{\sum_{t=1}^s \omega_{ij_t} P_{ij_t} A_{ij_t} \lambda_{ij_t}}{\sum_{t=1}^s \omega_{ij_t} \lambda_{ij_t}}, \text{ где } O_{пкij} \in [0; 25]; \tag{4}$$

$$\omega_{ij} = L_{ij} \left(\sum_{i=1}^m L_{ij_i} \right)^{-1}; P_{ij} = \frac{\sum_{v=1}^q P_{ij_v}}{q},$$

где s — число семестров; t — номер семестра; m — число дисциплин за все время обучения; q — число оценок студента в течение одного семестра; v — номер оценки; ω_0 — вес дисциплины i для компетенции j вычисляется делением общего числа часов

$$L_{ij} \text{ для дисциплины } i \text{ на общее число часов } \sum_{i=1}^m L_{ij_i},$$

согласно учебному плану (введение веса необходимо, чтобы придать большую значимость дисциплинам, на изучение которых отводится большее число часов по сравнению с другими дисциплинами); P_{ij} — средняя оценка студента по конкретному предмету за семестр, здесь могут учитываться оценки за лабораторные и семинарские занятия, тестирование остаточных знаний, рефераты и прочие контрольные мероприятия; A_{ij} — итоговая оценка по дисциплине i

за семестр t , причем $A_{ij} = \begin{cases} 0, & \text{если незачет,} \\ 5, & \text{если зачет,} \\ \text{оценка,} & \text{если экзамен} \end{cases}$; λ_{ij} — коэффициент уровня освоения компетенций.

В работе [25] авторы рекомендуют задание уровнем сформированности компетенций проводить на основе классификации Блума [26]. Уровни освоения компетенций можно представить в виде табл. 5.

В различных учебных планах дисциплины могут оказаться в разных семестрах, и не всегда напрямую связаны с уровнем усвоения компетенций, поэтому для определения уровня компетенций после изучения дисциплин предлагается использовать графовую модель взаимодействия компетенций и дисциплин (см. рис. 2) и каркасную модель, подробно описанную в статье [24]. Для определения уровня освоения компетенций λ для всех дисциплин находим пути формирования компетенций. Для каждой компетенции находим последнюю дисциплину в пути, в котором она формировалась и первую дисциплину, в которой эта компетенция начинает формироваться (во всех предыдущих вершинах графа данная компетенция не встречается). Используя

Таблица 5

Уровни освоения компетенций

Глубина знаний	Уровень λ
Формирование знания	1
Формирование понимания	2
Способность применения	3
Способность осуществлять анализ	4
Способность осуществлять синтез	5
Способность оценивать	6

полученные цепочки для каждой дисциплины, делим их на равные интервалы и каждому последовательно присваиваем определенный уровень освоения компетенций λ .

Оценка компетенций по всем дисциплинам

Чтобы получить оценку O_j компетенции j по всем дисциплинам надо выполнить следующую последовательность действий.

1. Оценить компетенцию j по каждой дисциплине i отдельно. Для этого следует воспользоваться формулой (4) и получить $O_{пк,ij}$, причем $O_{пк,ij} \in [0; 25]$.

2. Сопоставить общепринятые оценки от 1 до 5 с полученным результатом от 1 до 25. Для этого, согласно формуле (1), предлагается ввести соответствие в виде табл. 6.

3. Оценка O_j рассчитывается как среднее арифметическое $\bar{O}_{пк,ij}$ по формуле

$$O_j = \frac{\sum_{i=1}^m \bar{O}_{пк,ij}}{m}, \quad (5)$$

где m — число дисциплин, формирующих компетенцию j ; i — номер дисциплины.

4. Общая оценка компетентности студента по всем дисциплинам и компетенциям вычисляется по формуле

$$O_{общ} = \frac{\sum_{j=1}^n O_j}{n}. \quad (6)$$

На рис. 5 (см. третью сторону обложки) дано графическое представление зависимости развития компетенции ПК-1 относительно дисциплин. Имея наглядно представленную зависимость развития компетенции ПК-1 от дисциплин, читаемых в разных семестрах, можно проанализировать развитие ПК-1 конкретного студента на протяжении обучения и какие предметы давали большее или меньшее приращение для этой компетенции.

Таблица 6

Таблица соответствия оценок

Оценки $\bar{O}_{пк,ij}$	$O_{пк,ij}$
1	0—5
2	6—10
3	11—15
4	16—20
5	21—25

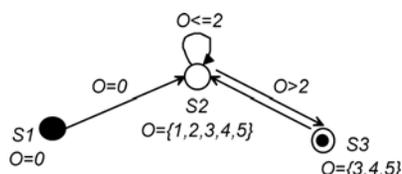


Рис. 7. Модель состояния студента в результате изучения одной дисциплины

В итоге, оценив ПК выпускника, можно построить их профили (рис. 6, см. третью сторону обложки) и провести анализ успеваемости студентов, выявив средние, наилучшие и наихудшие результаты, отобразив их графически.

Предлагается представить профили ПК студентов в следующем виде. Пусть $K = \{K_f\}$ — множество профилей компетенций студентов, где каждый профиль компетенций f -го студента представляется в виде: $K_f = \{Name_f, K_j, O_j\}$, где $Name_f$ — фамилия студента, K_j — название компетенции, O_j — численная оценка компетенции j , $j = 1...n$, $f = 1...w$.

Студент наиболее компетентен, т. е. имеет максимально высокие оценки по компетенциям, если график приближается к границе окружности, где значения оценок равны 5 (рис. 6). Чтобы представить численную оценку компетентности студента предлагается использовать формулы вычисления площадей секторов на лепестковой диаграмме.

Студенты максимально компетентны, если по всем компетенциям имеют оценку 5 и, следовательно, площадь профиля компетенций достигает максимального значения:

$$S_{max} = \frac{1}{2} 5^2 n \sin \frac{360^\circ}{n} = 12,5 n \sin \frac{360^\circ}{n},$$

где n — число компетенций.

Площадь профиля студента f вычисляется по формуле

$$S_f = \sum_{j=2}^n \frac{1}{2} O_{j-1} O_j \sin \frac{360^\circ}{n},$$

где j — компетенции.

Определим функцию \bar{h}_f относительной компетентности студента f следующим образом: $\bar{h}_f = \frac{S_f}{S_{max}}$.

Студент f наиболее компетентен, если функция \bar{h}_f достигает минимального значения, т. е. $\bar{h}_f \rightarrow \min$.

Модель компетенций выпускника

Модель компетенций студента в процессе обучения можно представить в виде диаграммы состояний UML (рис. 7), выделив состояние студента (выпускника) до обучения, в процессе обучения и после изучения дисциплины, а условия перехода, поставив в зависимость от оценки O за предмет.

Модель изменения компетенций студента за все время обучения в вузе тогда представляется в виде модели состояний с вложенными элементами. Фрагмент такой модели представлен на рис. 8.

Каждое состояние студента в процессе изучения дисциплины i можно представить в следующем виде: $S_{gi} = \{(K_1, f_{gi_1}), \dots, (K_n, f_{gi_n})\}$, где $g = \{1, 2, 3\}$ (S_1 — состояние до процесса изучения определенной дисциплины в вузе; S_2 — состояние во время изучения дисциплины в вузе; S_3 — состояние после изучения дисциплины); K_j — название компетенции.

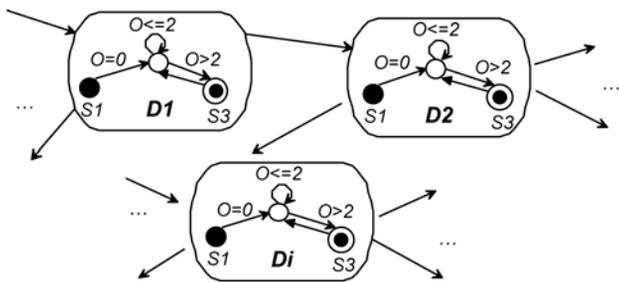


Рис. 8. Модель состояния студента в результате изучения всех дисциплин

Таблица 7

Матрица перехода состояний

Состояние	S_1	S_2	S_3
S_1	0	1	0
S_2	0	1	1
S_3	0	1	0

ции j , $j = 1, \dots, n$, где n — число компетенций определенной дисциплины i , $i = 1, \dots, m$, где m — число дисциплин в процессе обучения, f_{gij} — функция оценки компетенции j для дисциплины i :

$$f_{gij} = z_{gij} + \Delta_{gij}, \text{ где } z_{gij} = \begin{cases} O1_{пкij}, \text{ где } g = 1 \\ \frac{P_{ij} + A_{ij}}{2}, \text{ где } g = 2, \\ \bar{O}_{пкij}, \text{ где } g = 3 \end{cases}$$

где z_{gij} — значение оценки компетенции j для дисциплины i ; Δ_{gij} — приращение оценки компетенции j для дисциплины i ; P_{ij} , A_{ij} , $\bar{O}_{пкij}$ вычисляются по формуле (4) и соответствуют табл. 6; $O1_{пкij}$ — оценка компетенции на начальном этапе изучения дисциплины, получается при проведении входного тестирования.

Причем, если $g \leq 2$, то $f_{gij} \in [1; 5]$, если $g = 3$, то $f_{gij} \in [3; 5]$.

Пусть $\mathbf{B} = (b_{xy})$ — матрица переходов из одного состояния в другое (табл. 7), где

$$b_{xy} = \begin{cases} 1, \text{ если есть переход из состояния } x \text{ в состояние } y, \\ 0, \text{ если нет перехода из состояния } x \text{ в состояние } y. \end{cases}$$

Например, если $b_{22} = 1$, то студент учится неудовлетворительно и ему необходимо пересдать экзамены или зачеты, если $b_{32} = 1$, то студент получает дополнительное образование в вузе и т. д.

Заключение

Описанная численная методика позволит оценивать компетентность учащихся, выпускников, специалистов как профессиональную, так и общекуль-

турную. Получаемая оценка базируется на имеющихся образовательных стандартах и имеет практическую значимость. Предлагаемый метод оценки компетенций можно использовать при разработке:

- национальных стандартов в сфере профессиональной деятельности для создания унифицированной системы сертификации в различных регионах;
- учебных программ для повышения квалификации и переквалификации, развития карьеры;
- учебных курсов для повышения квалификационного уровня без отрыва от работы.

Список литературы

1. **Федеральный** государственный образовательный стандарт высшего профессионального образования по направлению подготовки 230400 "Информационные системы и технологии" (квалификация (степень) "бакалавр") [Электронный ресурс]: утв. приказом М-ва обр. и науки РФ от 14 янв. 2010 г. № 25. URL: http://www.edu.ru/db-mon/mo/Data/d_10/prm25-1.pdf (дата обращения: 25.02.2013).
2. **Пахаренко Н. В., Зольникова И. Н.** Модель определения уровня сформированности общекультурных и профессиональных компетенций // *Современные проблемы науки и образования*. 2013. № 5. [Электронный журнал]. URL: <http://www.science-education.ru/106-7502> (дата обращения: 20.10.2013).
3. **Байденко В. И.** Компетентностный подход к проектированию государственных образовательных стандартов высшего профессионального образования (методологические и методические вопросы): метод. пособие. М.: Исследовательский центр проблем качества подготовки специалистов, 2005. 114 с.
4. **Бочагов О. В.** Один из инновационных методов оценки компетентностей студентов // *Инновации в образовании*. 2010. № 2. С. 63–66.
5. **Симонов В. П.** Педагогический менеджмент: 50 НОУ-ХАУ в области управления образовательным процессом: учеб. пособие. 2-е изд. М.: Высшее образование, 1997. 264 с.
6. **Торп С., Клиффорд Дж.** Коучинг: руководство для тренера и менеджера. СПб.: Питер, 2004. С. 26–27.
7. **Сергеева Л. В.** Формирование информационно-технологической компетентности учащихся гимназии Дис. канд. пед. наук. Самара, 2005. 191 с.
8. **Осокина О. М., Ростовцев А. Н.** Оценка уровня сформированности ИТ-компетенций учащихся школы менеджмента и маркетинга по ступеням научения // *Социальная политика и социология*. 2009. № 10. С. 208–215.
9. **Касагиков А. Д., Лейбов А. М., Осокина О. М.** Оценка уровня сформированности ИТ-компетенций студентов-технологов // *Современное машиностроение. Наука и образование*. Санкт-Петербург: Изд-во Политехн. ун-та, 2013. С. 100–108.
10. **Лайл М. Спенсер, Сайн М. Спенсер.** Компетенции at work. Модели максимальной эффективности работы. М.: НИРО, 2005. 372 с.
11. **Уидет С., Холфорд С.** Руководство по компетенциям. М.: НИРО, 2008. 228 с.
12. **Шевандрин Н. И.** Психодиагностика, коррекция и развитие личности: учебник для вуза. М.: Вильямс, 2001. 510 с.
13. **Елисеев О. П., Амтхауэра Р.** Тест структуры интеллекта (TSI). Практикум по психологии личности. СПб.: Питер, 2003. С. 342–370.
14. **Капустина А. Н.** Многофакторная личностная методика Р. Кэттелла. — СПб.: Речь, 2001. 112 с.
15. **Практическая психодиагностика. Методики и тесты:** учеб. пособие. Самара: Бахрах, 1998.
16. **Леонтьев Д. А.** Тест смысло-жизненных ориентаций (СЖО). 2-е изд. М.: Смысл, 2000. 18 с.
17. **Фетискин Н. П., Козлов В. В., Мануйлов Г. М.** Диагностика мотивации достижения (А. Мехрабиан) // *Социально-психологическая диагностика развития личности и малых групп*. М.: Изд. Института психотерапии, 2002. С. 98–102.
18. **Ильин Е. П.** Мотивация и мотивы. СПб.: Питер, 2000. С. 401–405.

19. **Дерманова И. Б.** Диагностика эмоционально-нравственного развития. СПб.: Питер, 2002. С. 142–146.

20. **Экспериментально-диагностический** комплекс (ЭДК) [Электронный ресурс]. URL: <http://testpsy.net/ru> (дата обращения: 12.09.2013).

21. **Psychometric Expert**. URL: <http://www.psychometrica.ru> (дата обращения: 12.09.2013).

22. **Система** тестирования Maintest 4, HR-Лаборатория "Human Technologies". [Электронный ресурс]. URL: <http://www.ht.ru/tests/maintest4/#tability> (дата обращения: 12.09.2013).

23. **Боюр Р. В.** Практическое применение матрицы компетенций для мониторинга соответствия компетенций обучающихся в вузе внешним требованиям. URL: <http://edu.tltsu.ru/>

sites/sites_content/site117/html/media2286/practical_matrix.doc (дата обращения: 12.09.2013).

24. **Курилова О. Л., Липатова С. В.** Применение каркасной модели работы с ориентированным графом для оптимизации учебного плана // Вестник компьютерных и информационных технологий. 2014. № 4. С. 48–55.

25. **Тельнов Ю. Ф., Капустина О. А., Гаспарян М. С.** Концептуальные основы проектирования основной образовательной программы по направлению "Прикладная информатика". М.: МГУЭСИ, 2009. С. 193–198.

26. **Таксономия** Блума. URL: http://cniit.mpei.ac.ru/textbook/01_03_01_04.htm (дата обращения: 25.10.2013).

A. A. Smagin, Professor, Head of the Chair, **S. V. Lipatova**, Assistant Professor,
O. L. Kurilova, Senior Lecturer, e-mail: oxana197208@rambler.ru

Method of Numerical Evaluation of Professional and Common Cultural Competences in Undergraduates and Graduates

In the article the authors present a method of scoring of common cultural and professional competences. They suggest the way of formalizing and measuring of competences and their components in undergraduates and graduates. Presented algorithm consists of two parts: the first performs scoring of common cultural competence (CC) based on processing of test results provided by student, while the second one analyzes average and final grades in disciplines considering factors defined by experts to obtain the professional competence's score. CC scoring is used to determine the student's level of personal qualities and social skills like intelligence, teamwork, etc.

The authors also show the correlation of curriculum subjects based on incoming and outgoing competences within competence approach of Federal State Education Standard of Higher Vocational Education (FSES HVE) and present the competence model of graduates.

Keywords: competence, curriculum, evaluation of common cultural competences, evaluation of professional competences

References

1. **Federalnyy** gosudarstvennyy obrazovatelnyy standart vysshogo professionalnogo obrazovaniya po napravleniyu podgotovki 230400 "Informatsionnyye sistemy i tehnologii" (kvalifikatsiya (stepen) "bakalavr") [Elektronnyy resurs]: utv. prikazom M-va obr. i nauki RF ot 14 yanv. 2010 g. № 25. URL: http://www.edu.ru/db-mon/mo/Data/d_10/prm25-1.pdf (data obrascheniya: 25.02.2014).

2. **Paharenko N. V., Zolnikova I. N.** Model opredeleniya urovnya sformirovannosti obschekulturnykh i professionalnykh kompetentsiy // Sovremennyye problemy nauki i obrazovaniya. 2013. № 5. [Elektronnyy resurs]. URL: <http://www.science-education.ru/106-7502> (data obrascheniya: 20.10.2013).

3. **Baydenko V. I.** Kompetentnostnyy podhod k proektirovaniyu gosudarstvennykh obrazovatelnykh standartov vysshogo professionalnogo obrazovaniya (metodologicheskie i metodicheskie voprosy): Metodicheskoe posobie. M.: Issledovatel'skiy tsentr problem kachestva podgotovki spetsialistov, 2005. 114 s.

4. **Bochagov O. V.** Odin iz innovatsionnykh metodov otsenki kompetentsiy studentov. *Innovatsii v obrazovanii*. 2010, № 2. S. 63–66.

5. **Simonov V. P.** Pedagogicheskiy menedzhment: 50 NOU-HAU v oblasti upravleniya obrazovatelnyim protsessom: uchebnoe posobie. M.: Vysshее obrazovanie, 2009. 354 p.

6. **Torp S., Klifford Dzh.** Kouching: rukovodstvo dlya trenera i menedzhera. SPb.: Piter, 2004. S. 26–27.

7. **Sergeeva L. V.** Formirovanie informatsionno-tehnologicheskoy kompetentnosti uchashchihsya gimnazii: Dis. kand. ped. nauk. Samara, 2005. 191 p.

8. **Osokina O. M., Rostovtsev A. N.** Otsenka urovnya sformirovannosti IT-kompetentsiy uchashchihsya shkoly menedzhmenta i marketinga po stupenyam naucheniya // Sotsialnaya politika i sotsiologiya. 2009. № 10. S. 208–215.

9. **Kasatkov A. D., Leybov A. M., Osokina O. M.** Otsenka urovnya sformirovannosti IT-kompetentsiy studentov-tehnologov // *Sovremennoe mashinostroenie. Nauka i obrazovanie*. 2013. P. 100–108.

10. **Layl M. Spenser, Sayn M. Spenser.** Kompetentsii at work. Modeli maksimalnoy effektivnosti raboty. M.: HIPPO, 2005. 372 p.

11. **Uiddet S., Holford S.** Rukovodstvo po kompetentsiyam. M.: HIPPO, 2008. 228 p.

12. **Shevandrin N. I.** Psihodiagnostika, korrektsiya i razvitie lichnosti: uchebnik dlya studentov vuza. M.: Vlyyaos, 2001. 510 p.

13. **Eliseev O. P., Amthauera R.** Test strukturyi intellekta (TSI): Praktikum po psikhologii lichnosti. SPb.: Piter, 2003. 560 p.

14. **Kapustina A. N.** Mnogofaktomaya lichnostnaya metodika R. Kettella. SPb.: Rech, 2001. 112 p.

15. **Prakticheskaya** psihodiagnostika. Metodiki i testyi: uchebnoe posobie. Samara: Izdatelskiy Dom "Bahrah", 2001. 672 p.

16. **Leont'ev D. A.** Test smyslozhiznennykh orientatsii (SZhO). 2-e izd. M.: Smyisl, 2000. 18 p.

17. **Fetiskin N. P., Kozlov V. V., Manuylov G. M.** Sotsialno-psihologicheskaya diagnostika razvitiya lichnosti i malyykh grupp: uchebnoe posobie. M.: Izdatelstvo instituta psihoterapii, 2002. 339 p.

18. **Il'in E. P.** Motivatsiya i motivyi. SPb.: Piter, 2000. P. 401–405.

19. **Dermanova I. B.** Diagnostika emotsionalno-nravstvennogo razvitiya. SPb.: Piter, 2002. S. 142–146.

20. **Экспериментально-диагностический** комплекс (EDK) [Elektronnyy resurs]. URL: <http://testpsy.net/ru> (data obrascheniya: 12.03.2014).

21. **Psychometric Expert**. [Elektronnyy resurs]. URL: <http://www.psychometrica.ru> (data obrascheniya: 12.03.2014).

22. **Система** testirovaniya Maintest 4, HR-Laboratoriya "Human Technologies". URL: <http://www.ht.ru/tests/maintest4/#tability> (data obrascheniya: 12.03.2014).

23. **Bbyur R. V.** Prakticheskoe primenenie matritsy kompetentsiy dlya monitoringa sootvetstviya kompetentsiy obuchayushchihsya v vuze vnesnim trebovaniyam. [Elektronnyy resurs]. URL: http://edu.tltsu.ru/sites/sites_content/site117/html/media2286/practical_matrix.doc (data obrascheniya: 12.03.2014).

24. **Kurilova O. L., Lipatova S. V.** Primenenie karkasnoy modeli raboty s orientirovannym grafom dlya optimizatsii uchebnogo plana. // Vestnik kompyuternyykh i informatsionnykh tehnologiy. 2014. № 4. S. 48–55.

25. **Telnov Yu. F., Kapustina O. A., Gasparian M. S.** Kontseptualnyye osnovyy proektirovaniya osnovnoy obrazovatelnoy programmy po napravleniyu "Prikladnaya informatika". M.: MGUESI, 2009. S. 193–198.

26. **Таксономия** Блума. [Elektronnyy resurs]. URL: http://cniit.mpei.ac.ru/textbook/01_03_01_04.htm (data obrascheniya: 12.03.2014).

УДК 004.7

В. Н. Тарасов, д-р техн. наук, проф., зав. каф., **Н. Ф. Бахарева**, д-р техн. наук, проф., зав. каф. ИВТ,
Г. А. Горелов, аспирант, **С. В. Малахов**, аспирант,

Поволжский государственный университет телекоммуникаций и информатики, г. Самара,
e-mail: vt@ist.psati.ru, bahareva-nf@psuti.ru, gleb_fox@bk.ru, malakhov-sv@psuti.ru

Анализ входящего трафика на уровне трех моментов распределений временных интервалов

Получено аналитическое решение для среднего времени ожидания для систем массового обслуживания типа $H_2/M/1$ с гиперэкспоненциальным распределением второго порядка интервалов времени входного потока путем решения интегрального уравнения Линдли методом спектрального разложения. Показано, что в этом случае распределения интервалов между требованиями входного потока можно аппроксимировать на уровне их трех первых моментов. Представлена программа-дополнение к анализатору трафика Wireshark для расчета моментных характеристик распределения интервалов между пакетами.

Ключевые слова: система массового обслуживания $H_2/M/1$, уравнение Линдли, метод спектрального разложения, анализатор трафика Wireshark, моментные характеристики интервалов

Введение

Для описания трафика широко используют класс экспоненциальных распределений, включающий подкласс так называемых субэкспоненциальных распределений, куда входят, в частности, распределения Вейбулла, логнормальное, гиперэкспоненциальное и гамма-распределение, имеющие коэффициенты вариации большие 1 (рис. 1) и относящиеся к распределениям с "тяжелым хвостом" (*heavy-tailed distributions*).

На рис. 1 приведены фрагменты хвостов функций плотности этих распределений, а для сравнения приведено классическое экспоненциальное распределение. Для всех приведенных распределений средние значения равны $m_\tau = 0,25$, а дисперсии для распределений с "тяжелым хвостом" равны $D_\tau = 0,25$, что дает коэффициент вариации интервала времени $c_\tau = 2,0$.

Как видно из этих графиков, даже при сравнительно небольшом коэффициенте вариации распределения $c_\tau = 2,0$, заметна тяжесть хвостов затухания приведенных выше функций плотностей $f(t)$ по сравнению с экспонентой.

Очевидно, что с увеличением параметра c_τ весомость хвоста распределения только возрастет. При этом под хвостом распределения будем понимать функцию $Q(x)$: $Q(x) = P(\xi \geq x) = Q([x, \infty))$, где ξ — некоторая случайная величина.

Таким образом, рассматривая только статистические характеристики второго порядка, можно

получить определенное представление об отличии распределения времени между поступлениями трафикового процесса и экспоненциального распределения, соответствующего пуассоновскому потоку. С учетом статистики третьего порядка это отличие может только усилиться. Из теории телетрафика известно, что задержки при использовании тяже-

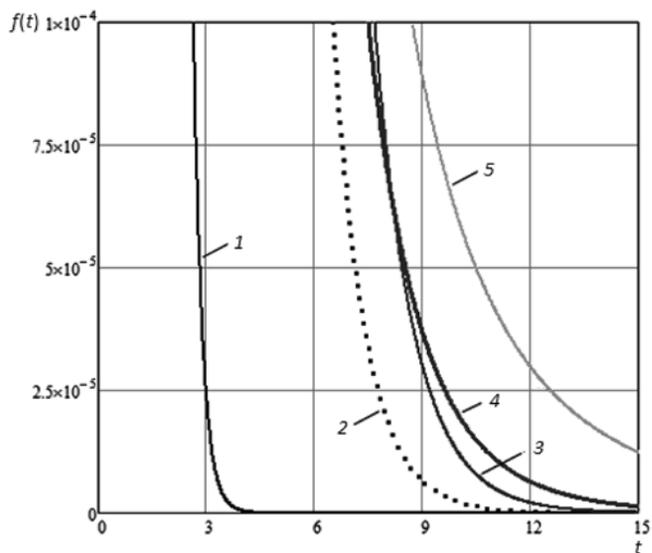


Рис. 1. Фрагменты хвостов функций плотности из класса экспоненциальных распределений:

1 — классическое экспоненциальное распределение; 2 — гамма; 3 — гиперэкспоненциальное; 4 — Вейбулла; 5 — логнормальное

лохвостных распределений в системах массового обслуживания (СМО) намного выше, чем при пуассоновском потоке. На практике такими распределениями описывается трафик с мультимедиа данными, так называемый "тяжелый контент".

К сожалению, в настоящее время не существует адекватного математического аппарата, позволяющего анализировать трафик с тяжелохвостным распределением интервалов между пакетами, в связи с чем, чаще всего для этого используется имитационное моделирование систем G/G/1 или G/M/1.

1. Постановка задачи

В работе ставится задача построения такого математического аппарата, который в отличие от классической модели массового обслуживания, позволял бы учитывать моменты распределения интервалов между пакетами входящего трафика более высокого порядка, чем первый момент. Из всех известных распределений с "тяжелыми хвостами", представленных на рис. 1, решение для среднего времени ожидания может быть получено только для гиперэкспоненциального распределения. В то же время даже интуитивно понятно, что для использования этого аппарата также потребуется информация о моментах более высокого порядка распределения интервалов реального трафика, которую можно получить, только внося коррективы в имеющиеся программные средства съема трафика. Таким образом, в статье решаются две задачи:

- получение точного решения для среднего времени ожидания для системы H₂/M/1;
- разработка дополнительного модуля выборки моментов поступления пакетов и расчета моментных характеристик интервалов.

Цель анализа телетрафика с помощью метода трех моментов распределения заключается в получении адекватных результатов по значению задержки пакетов, что позволяет:

- оценить возможные их потери;
- при больших задержках выявить узкие места сети или же угрозы безопасности сети.

Кроме того, большие задержки приводят к дополнительному увеличению общего трафика сети.

2. Анализ СМО H₂/M/1 и вывод формулы для среднего времени ожидания

Рассмотрим СМО H₂/M/1, где H₂ (см. рис. 1) — обозначение гиперэкспоненциального распределения 2-го порядка времени поступления требований в систему с функцией плотности

$$a(t) = p\lambda_1 e^{-\lambda_1 t} + (1 - p)\lambda_2 e^{-\lambda_2 t}, \quad (1)$$

а M — обозначение экспоненциального закона обслуживания с функцией плотности

$$b(t) = \mu e^{-\mu t}. \quad (2)$$

Преобразование Лапласа функции (1) имеет следующий вид:

$$A^*(s) = p \frac{\lambda_1}{s + \lambda_1} + (1 - p) \frac{\lambda_2}{s + \lambda_2}, \quad (3)$$

а функции (2)

$$B^*(s) = \frac{\mu}{s + \mu}. \quad (4)$$

Для исследования системы G/G/1, как известно, например из работы [1], используется интегральное уравнение Линдли:

$$W(y) = \begin{cases} y \int_{-\infty}^y W(y - u) dC(u), & y \geq 0, \\ 0, & y < 0, \end{cases} \quad (5)$$

где W(y) — функция распределения вероятностей (ФРВ) времени ожидания требования в очереди; C(u) — ФРВ предельной случайной величины $U = \lim_{n \rightarrow \infty} U_n = x_n - t_{n+1}$, где в свою очередь x_n — время обслуживания n-го требования C_n; t_{n+1} — интервал времени между поступлением требований C_n и C_{n+1}.

Суть решения уравнения (5) спектральным методом сводится к тому, чтобы для выражения

$$A^*(-s) \cdot B^*(s) - 1 \quad (6)$$

найти представление в виде произведения двух множителей, которое давало бы рациональную функцию от s [1]. Таким образом, для нахождения распределения времени ожидания необходимо следующее спектральное разложение:

$$A^*(-s) \cdot B^*(s) - 1 = \frac{\psi_+(s)}{\psi_-(s)}, \quad (7)$$

где ψ₊(s) и ψ₋(s) — некоторые рациональные функции от s, которые можно разложить на множители. Функции ψ₊(s) и ψ₋(s) должны удовлетворять определенным условиям согласно [1]:

1) для Re(s) > 0 функция ψ₊(s) является аналитической без нулей в этой полуплоскости;

2) для Re(s) < D функция ψ₋(s) является аналитической без нулей в этой полуплоскости, где D — некоторая положительная константа, определяемая из условия 3;

$$3) \lim_{t \rightarrow \infty} \frac{a(t)}{e^{-Dt}} < \infty.$$

Кроме того, функции $\psi_+(s)$ и $\psi_-(s)$ должны обладать следующими свойствами:

$$\begin{aligned} \text{для } \operatorname{Re}(s) > 0 \quad \lim_{|s| \rightarrow \infty} \frac{\psi_+(s)}{s} &= 1; \\ \text{для } \operatorname{Re}(s) < 0 \quad \lim_{|s| \rightarrow \infty} \frac{\psi_-(s)}{s} &= -1. \end{aligned} \quad (8)$$

Определим теперь выражение $A^*(-s) \cdot B^*(s) - 1 = \frac{\psi_+(s)}{\psi_-(s)}$ для распределений (1) и (2) с учетом выражений (3) и (4):

$$\begin{aligned} \frac{\psi_+(s)}{\psi_-(s)} &= \left[p \frac{\lambda_1}{\lambda_1 - s} + (1 - p) \frac{\lambda_2}{\lambda_2 - s} \right] \frac{\mu}{\mu + s} - 1 = \\ &= \frac{[p\lambda_1(s + \lambda_2) + (1 - p)\lambda_2(s + \lambda_1)]\mu - (\lambda_1 - s)(\lambda_2 - s)(\mu + s)}{(\lambda_1 - s)(\lambda_2 - s)(\mu + s)} = \\ &= \frac{\mu(a_0 - a_1s) - (\lambda_1 - s)(\lambda_2 - s)(\mu + s)}{(\lambda_1 - s)(\lambda_2 - s)(\mu + s)}, \end{aligned} \quad (9)$$

где коэффициенты $a_0 = \lambda_1\lambda_2$, $a_1 = p\lambda_1 + (1 - p)\lambda_2$.

В числителе правой части равенства (9) получился многочлен 3-й степени, и нам остается определить его коэффициенты для разложения многочлена на множители. Коэффициенты многочлена приведены в таблице.

Коэффициент многочлена

Степени слагаемых в числителе дроби	Коэффициенты многочлена
s^0	0
s^1	$\mu[\lambda_1(1 - p) + \lambda_2p] - \lambda_1\lambda_2 \equiv c_1$
s^2	$\lambda_1 + \lambda_2 - \mu \equiv c_2$
s^3	-1

Тогда выражение (10) можно разложить на множители:

$$\begin{aligned} \frac{\psi_+(s)}{\psi_-(s)} &= \\ &= \frac{s(s^2 - c_2s - c_1)}{(s - \lambda_1)(\lambda_2 - s)(\mu + s)} = \frac{s(s + s_1)(s - s_2)}{(s - \lambda_1)(\lambda_2 - s)(\mu + s)}, \end{aligned}$$

где $-s_1 = -\left(\sqrt{\frac{c_2^2}{4} + c_1} - \frac{c_2}{2}\right)$ — отрицательный корень

квадратного уравнения в числителе дроби; $s_2 = \frac{c_2}{2} +$

$+\sqrt{\frac{c_2^2}{4} + c_1}$ — положительный корень.

Исследуем знаки корней s_1 и s_2 , чтобы убедиться в сказанном. Для этого вспомним, что среднее значение интервала между требованиями $\bar{\tau}_\lambda = \int_0^\infty ta(t)dt = \frac{p}{\lambda_1} + \frac{(1-p)}{\lambda_2}$. Тогда средняя интенсивность входного потока $\lambda = \bar{\tau}_\lambda^{-1} = \frac{\lambda_1\lambda_2}{p\lambda_2 + (1-p)\lambda_1}$.

Так как в стационарном режиме загрузка $\rho = \frac{\lambda}{\mu} < 1$, то $\lambda < \mu$. Отсюда следует, что $\lambda_1\lambda_2 < \mu[p\lambda_2 + (1-p)\lambda_1]$ и коэффициент $c_1 > 0$.

Тогда один корень будет отрицательным ($-s_1$), а второй корень (s_2) — положительным.

С учетом знаков корней и приведенных выше условий (1–3) строим рациональные функции $\psi_+(s)$ и $\psi_-(s)$:

$$\psi_+(s) = \frac{s(s + s_1)}{s + \mu}, \quad \psi_-(s) = \frac{(s - \lambda_1)(\lambda_2 - s)}{s - s_2}. \quad (10)$$

Функции (10) удовлетворяют также и условиям (8). Следуя [1], определим постоянную $K =$

$$= \lim_{s \rightarrow 0} \frac{\psi_+(s)}{s} = \lim_{s \rightarrow 0} \frac{s + s_1}{s + \mu} = \frac{s_1}{\mu}, \text{ где } s_1 = \sqrt{\frac{c_2^2}{4} + c_1} - \frac{c_2}{2}.$$

Тогда функция $\psi_+(s)$ позволяет найти преобразование Лапласа для ФРВ времени ожидания $W(y)$:

$$\begin{aligned} \Phi_+(s) &= \frac{K}{\psi_+(s)} = \frac{s_1(s + \mu)}{\mu s(s + s_1)}. \text{ Заметив, что } s\Phi_+(s) = \\ &= W^*(s) \text{ есть преобразование Лапласа для функции} \end{aligned}$$

$$W^*(s) = \frac{s_1(s + \mu)}{\mu(s + s_1)}. \quad (11)$$

$$\text{Отсюда } \frac{dW^*(s)}{ds} = \frac{s_1\mu(s_1 + s) - s_1(s + \mu)\mu}{\mu^2(s + s_1)^2}.$$

Учитывая свойство преобразования Лапласа, найдем среднее время ожидания:

$$\bar{W} = -\left. \frac{dW^*(s)}{ds} \right|_{s=0} = \frac{-s_1^2\mu + \mu^2s_1}{\mu^2s_1^2} = \frac{1}{s_1} - \frac{1}{\mu}, \quad (12)$$

где $s_1 = \sqrt{\frac{c_2^2}{4} + c_1} - \frac{c_2}{2}$, $c_1 = \mu[\lambda_1(1 - p) + \lambda_2p] - \lambda_1\lambda_2$, $c_2 = \lambda_1 + \lambda_2 - \mu$.

Для того чтобы результат (12) использовать на практике, необходима информация о моментных характеристиках интервалов. Для этого воспользуемся анализатором Wireshark и приведем краткие сведения о дополнительно разработанной программе.

3. Программа определения моментных характеристик распределения трафика

Программа Wireshark (ранее — Ethereal) — программа-анализатор трафика для компьютерных сетей технологии Ethernet и некоторых других имеет графический пользовательский интерфейс и гораздо больше возможностей по сортировке и фильтрации информации, чем остальные анализаторы. Разработанная авторами программа в дополнение к анализатору позволяет осуществлять выборку моментов времени поступления пакетов, вычленив входящий трафик из всего набора данных, полученных программой Wireshark. Далее с использованием известных формул математической статистики определяются моментные характеристики распределения временных интервалов. В работе использованы статистики до 3-го порядка, которые позволяют судить о характере распределения интервалов. Например, коэффициент вариации показывает отличие трафика от пуассоновского потока и совместно с асимметрией позволяет судить о степени весомости хвостов распределений.

Среднее значение интервала между соседними пакетами

$$\bar{\tau} = \frac{1}{N} \sum_{k=0}^N (t_{k+1} - t_k),$$

где t_k — моменты времени поступления пакетов; N — число анализируемых интервалов.

Выборочная дисперсия равна $D_B = \bar{t}^2 - \bar{\tau}^2$, где

$$\bar{t}^2 = \frac{1}{N} \sum_{k=0}^N (t_{k+1} - t_k)^2 - \text{второй начальный момент.}$$

Коэффициент вариации $c = \sigma_B / \bar{\tau}$, где $\sigma_B = \sqrt{D_B}$.

$$\text{Асимметрия } A_s = \bar{t}^3 / \sigma_B^3, \text{ где } \bar{t}^3 = \frac{1}{N} \sum_{k=0}^N (t_{k+1} - t_k)^3.$$

Для расчета моментных характеристик была разработана программа, которая из всего входного файла с данными о захвате сетевого трафика выбирает только данные, относящиеся к поступившим пакетам, и для них выполняет расчет интервалов и их моментных характеристик.

В программе разработаны два класса (в терминах объектно-ориентированного программирования):

- TrafficLogParams хранит данные о временах приходов пакетов, их интервалов и рассчитывает моментные характеристики, также предоставляет методы сохранения данных в файлы и их загрузки из файлов;
- LogParser — статический класс, проводящий анализ входного файла и добавление данных в класс TrafficLogParams.

Главный метод класса LogParser получает в качестве входных данных имя файла и IP-адрес хоста.

Каждая строка исходного файла обрабатывается и из нее выбираются данные о времени и два IP-адреса — отправителя и получателя. Если поле получателя совпадает с IP-адресом хоста, тогда время поступления пакета добавляется в массив времен поступления пакетов класса TrafficLogParams.

4. Результаты работы программы

Авторской программой был проанализирован файл с данными об агрегированном трафике, поступающем на прокси-сервер вуза почти за час сема в целях определения задержки пакетов. Входной файл содержал более 2 150 000 строк, обработка вручную которых не представляется возможной. На рис. 2 приведены результаты (в секундах), полученные после анализа 628 182 интервалов между пакетами.

Файл	Справка
Начальный момент 1-го порядка:	5,097781e-003
Начальный момент 2-го порядка:	3,325837e-004
Начальный момент 3-го порядка:	5,505049e-005
Дисперсия:	3,065963e-004
Коэффициент вариации:	3,434807e+000
Асимметрия:	1,025441e+001
Количество пакетов:	628183

Рис. 2. Результат работы программы анализа лог-файлов

Полученные данные свидетельствуют о том, что анализируемый трафик сильно отличается от пуассоновского (коэффициент вариации $c = 3,43$ вместо 1), значение асимметрии, равной $A_s = 10,25$, говорит о том, что распределение интервалов между пакетами трафика относится к распределениям с тяжелыми хвостами (например, у экспоненциального закона $A_s = 2$). Для расчета характеристик такого трафика требуется соответствующий математический аппарат, в качестве которого выберем систему массового обслуживания $H_2/M/1$, для которой получена формула среднего времени ожидания.

5. Практическое применение полученных результатов

Рассмотрим полученный результат (12) на примере входного распределения с весомым хвостом.

С использованием преобразования Лапласа (3) определим начальные моменты распределения (1):

$$\begin{cases} \bar{\tau}_\lambda = \frac{p}{\lambda_1} + \frac{(1-p)}{\lambda_2} \\ \bar{\tau}_\lambda^2 = \frac{2p}{\lambda_1^2} + \frac{2(1-p)}{\lambda_2^2} \\ \bar{\tau}_\lambda^3 = \frac{6p}{\lambda_1^3} + \frac{6(1-p)}{\lambda_2^3} \end{cases} \quad (13)$$

Подставив в систему (14) полученные в разд. 3 результаты по начальным моментам распределения интервалов между пакетами (рис. 2), для определения неизвестных параметров входного распределения (1) λ_1 , λ_2 и p получим следующую систему уравнений:

$$\begin{cases} \frac{p}{\lambda_1} + \frac{(1-p)}{\lambda_2} = 5,0978 \cdot 10^{-3}, \\ \frac{2p}{\lambda_1^2} + \frac{2(1-p)}{\lambda_2^2} = 3,3258 \cdot 10^{-4}, \\ \frac{6p}{\lambda_1^3} + \frac{6(1-p)}{\lambda_2^3} = 5,5050 \cdot 10^{-5}, \end{cases} \quad (14)$$

решив которые найдем эти параметры.

Решение системы (14) в пакете MathCad методом сопряженных градиентов дает следующие результаты: $p \approx 0,950$, $\lambda_1 \approx 417,985$, $\lambda_2 \approx 17,556$.

Промежуточные параметры: $c_1 \approx 2,429 \cdot 10^3$, $c_2 \approx 174,441$, $s_1 \approx 12,962$, а среднее время ожидания в очереди $\bar{W} \approx 0,073$.

Определим для сравнения среднее время ожидания для классической системы M/M/1. В нашем случае среднее время обслуживания пакетов $\bar{\tau}_\mu \approx 0,00038$, интенсивность обслуживания $\mu \approx 2631,6$, загрузка канала $\rho \approx 0,075$.

Тогда среднее время ожидания пакетов $\bar{W} = \frac{\rho/\mu}{1-\rho} \approx 3 \cdot 10^{-4}$.

Таким образом, система массового обслуживания $H^2/M/1$ с учетом весомости хвоста входного распределения, даже при маленьких нагрузках дает задержку во много раз большую, чем классическая модель.

Заключение

Полученные результаты свидетельствуют о том, насколько оптимистичные результаты дает классическая система M/M/1 по сравнению с рассмотренной системой $H_2/M/1$ в случае высокой весомости хвоста распределения входного потока. Поэтому данный результат с успехом может быть применен в современной теории телетрафика, где задержки пакетов входящего трафика играют первостепенную роль.

Заметим, что распределение (1), содержащее три неизвестных параметра λ_1 , λ_2 и p , позволяет с помощью метода уравнений моментов (14) аппроксимировать неизвестные входные распределения с тяжелым хвостом на уровне трех первых моментов.

Список литературы

1. **Клейнрок Л.** Теория массового обслуживания / Пер. с англ. под редакцией В. И. Неймана. М.: Машиностроение, 1979. 448 с.
2. **Wireshark** official web-site. URL: <http://www.wireshark.org/>

V. N. Tarasov, Professor, N. F. Bahareva, Professor, G. A. Gorelov, Graduate Student, S. V. Malakhov, Graduate Student, Volga State University of Telecommunications and Informatics, Samara

Analyzing the Incoming Traffic at the Three Moments Distribution of Time Intervals

In this paper, an analytical solution for the average waiting time for a queuing system type $H_2/M/1$ Hyperexponential distribution invaded order intervals of the input stream by solving an integral equation method Lindley spectral decomposition. It is shown that in this case the distribution of intervals between the requirements of the input flow can be approximated at its first three moments.

The program of the complement of the traffic analyzer Wireshark to calculate the torque characteristics of the distribution of intervals between packets.

Keywords: queuing system $H_2/M/1$ equation Lindley, spectral decomposition method, traffic analyzer Wireshark, torque characteristics intervals

References

1. **Kleinrock L.** Theory. Queuing Systems. Vol. 1: Wiley, 1976. 448 p.
2. **Wireshark** official web-site. URL: <http://www.wireshark.org/>

СОЦИАЛЬНО-ЭКОНОМИЧЕСКИЕ СИСТЕМЫ SOCIAL AND ECONOMIC SYSTEMS

УДК 519.876.5:004.048

Ю. С. Сербулов, д-р техн. наук, проф., **Д. А. Глухов**, канд. техн. наук, доц., e-mail: gluhovda@yandex.ru, ФГБОУ ВПО "Воронежская государственная лесотехническая академия",
В. А. Мальшев, д-р техн. наук, зам. начальника каф.,
ВУНЦ ВВС "Военно-воздушная академия" им. проф. Н. Е. Жуковского и Ю. А. Гагарина,

Модель подсистемы нейтрализации внешних воздействий на производственно-экономическую систему в условиях конкурентной борьбы

Рассмотрено решение задачи нейтрализации внешних воздействий конкурентов на производственно-экономическую систему. Задача решается на основе применения искусственной иммунной системы. Сформулирована и разработана модель подсистемы нейтрализации внешних воздействий. Подробно раскрыты этапы функционирования наиболее сложных компонентов данной подсистемы.

Ключевые слова: производственно-экономическая система, искусственная иммунная система, нейтрализация внешних воздействий, модуль моделирования, база моделей

Введение

Современный рынок предполагает непрерывную борьбу всех участников рыночных отношений. В этих условиях на новый уровень выходят вопросы эффективности управления конкурентоспособностью производственно-экономических структур (ПЭС). Под ПЭС понимается любая организация (предприятие, банк, фирма и т. п.), производящая или потребляющая какой-либо товар (услугу). Для повышения конкурентоспособности ПЭС крайне необходима их конфликтная устойчивость к воздействиям внешней среды, а главное, преднамеренному воздействию конкурентов. В связи с этим становится актуальным создание подсистем нейтрализации таких воздействий на основе искусственных иммунных систем (ИИС), которые по аналогии с биологическими системами предназначены для защиты [1, 2].

Функции подсистемы нейтрализации внешних воздействий

Иммунная система человека обладает тремя основными свойствами: распределенность, самоорганизуемость и низкая требовательность к вычислительным ресурсам. Именно эти свойства должны иметь подсистема нейтрализации внешних воздействий (ПНВВ) ПЭС, которая по своим характеристикам приближалась бы к максимально эффективной.

В обобщенном виде ПНВВ должна выполнять ряд функций, которые можно разбить на несколько операций.

В первой операции создания и эволюции генной библиотеки происходит накопление информации о характере внешних воздействий на ПЭС. Генная библиотека ИИС может содержать, например, данные о характеристических параметрах воздействий, схеме воздействий, их экстенсionale и интенсionale, на основании которых будут генерироваться особые информационные агенты — детекторы, служащие аналогами лимфоцитов. Начальные данные для формирования генной библиотеки выбираются исходя из предметной области функционирования ПЭС, в частности, их слабых, с точки зрения надежности и эффективности функционирования, мест. В дальнейшем при обнаружении детекторами активности конкурентов на входе системы к библиотеке будут добавляться соответствующие этим проявлениям новые "гены". Следует заметить, что поскольку размер генной библиотеки ограничен, в ней сохраняются только "гены", проявляющиеся наиболее часто.

Во второй операции путем произвольного комбинирования "генов" происходит генерирование так называемых преддетекторов (аналоги "зародышевых" лимфоцитов), которые затем с помощью механизма негативной селекции проверяются на совместимость (или точнее, на несовместимость) с нормальным функционированием ПЭС.

модуль реализации данного алгоритма (МРАНВВ), который вырабатывает управляющие сигналы для исполнительной подсистемы (I_C) ПЭС. Таким образом, цепь замыкается и после начала выполнения алгоритма происходит постоянное слежение за функционированием системы и текущими результатами выполнения алгоритма.

Данная модель наглядно показывает все взаимосвязи между отдельными этапами процедуры нейтрализации внешних воздействий и позволяет определить формальное разделение подзадач в рамках ЗНВВ.

Алгоритм нейтрализации внешних воздействий

Самым сложным модулем является МРАНВВ, который собственно должен реализовать решение ЗНВВ. АНВВ является, по сути, обобщенным и его реализация требует выполнения определенных действий для его конкретной детализации. Поэтому представим процесс функционирования МРАНВВ более развернуто в виде следующих этапов:

1. Определение целевой функции ЗНВВ Γ_3 в соответствии с выбранным АНВВ.
2. Декомпозиция ЗНВВ до уровня элементарных задач (ЭЗ).
3. Построение динамической модели текущей ситуации $M_{\text{стек}}$ и модели предпочтений текущей ситуации $M_{\text{п.т.с}}$.
4. Определение критериев для реализации промежуточных задач (ПЗ) $R_{\text{пз}}$ и их ситуационных весов b_R .
5. Определение необходимого и достаточного объема информации $V_{\text{иэ}}$ для каждой ЭЗ.
6. Покритериальный сбор информации I^+UI^- .
7. Фильтрация информации I^+ .
8. Принятие элементарных решений (ЭР).
9. Проверка ЭР на согласованность и конфликт.
10. Разрешение конфликта на основе выбора ситуационных предпочтений $PR_{\text{с.тек}}$ и принятие решений по промежуточным задачам $S_{\text{п.з}}$.
11. Принятие общего решения на нейтрализацию внешних воздействий SR согласно целевой функции.

Опишем более подробно сущность и особенности каждого этапа.

Этап 1. По своей сути, цель ЗНВВ можно представить как выбор альтернативы динамического преобразования модели текущей ситуации для приобретения ею необходимых качественных и количественных показателей за минимальное время. Тогда можно записать

$$\Gamma_3(t): M_{\text{стек}} \xrightarrow{t \rightarrow \min} M_{\text{с.тр}}, \quad (1)$$

где $M_{\text{с.тр}}$ — требуемая модель ситуации.

Исходя из выражения (1), для достижения цели ЗНВВ нужно знать модели текущей и требуемой ситуаций. В свою очередь, в этих моделях для ЗНВВ интерес представляет только состояние управляемых

переменных [3]. Применительно к ЗНВВ управляемыми переменными являются наборы "антител", которые содержатся в макрофагах, вырабатываются Т-лимфоцитами и В-лимфоцитами. В ПНВВ эти роли могут выполнять: нейронная сеть Кохонена — как макрофаги; Т-лимфоциты и В-лимфоциты — частные алгоритмы нейтрализации внешних воздействий (решения элементарных задач), формируемых МРАНВВ. Отличие целей этих алгоритмов заключается в том, что Т-лимфоциты прекращают работу тех компонентов ПЭС, которые подверглись внешним воздействиям, приведшим к их неправильному функционированию. В-лимфоциты нейтрализуют непосредственно внешние воздействия до того, как они выведут из строя нормально функционирующие компоненты ПЭС.

Тогда применительно к ЗНВВ модели ситуаций можно записать как

$$M_c = \{C_{ci}\}, i = \overline{1, N_c},$$

где C_{ci} — управляемые переменные данной ситуации; N_c — число управляемых переменных в данной ситуации.

Следовательно, целевую функцию ЗНВВ необходимо преобразовать к следующему виду:

$$\Gamma_{\text{ЗНАВ}}(t): \{C_{ci \text{ тек}}\} \xrightarrow{t \rightarrow \min} \{C_{ci \text{ тр}}\}.$$

Следует отметить тот факт, что если состояние неуправляемых переменных или констант текущей ситуации не позволяет реализовать выражение (1), то задача в принципе не может быть решена и выполнение дальнейших этапов не имеет смысла. В этом случае следует дать формулировку новой задачи или отказаться от ее решения.

Этап 2. Если ЗНВВ является сложной (иерархической или взаимосвязанной параллельной), то необходимо провести ее декомпозицию в целях сведения ЗНВВ к множеству ЭЗ, не требующих вмешательства человека [4]. Декомпозиция выполняется следующим образом:

- формулируется множество критериев решения ЗНВВ;
- для каждого критерия в ЗНВВ определяется множество альтернатив в зависимости от числа управляемых переменных, их весов и взаимосвязей;
- если альтернатива не имеет аксиомного подтверждения, то на ее основе, в свою очередь, формируется новая промежуточная задача, и определяются свои варианты разрешения задачи и т. д.

Сложность данного этапа заключается в правильном нахождении альтернатив и формулировки новой задачи. Однако если первоначальная проблема укладывается в предметную область базы знаний системы, то с использованием семантических сетей или сети фреймов возможно достижение необходимого уровня декомпозиции общей задачи. Исходя из изложенного, этот этап можно описать следую-

щим образом. Пусть для какой-либо ПЗ, принадлежащей некоторой предметной области (ПО) по определенному критерию, существует множество альтернатив ее разрешения $A_3 = \{a_1, a_2, \dots, a_k\}$. Причем для каждой альтернативы справедливо

$$\forall a_k = \{C_{C_{i,k}}\},$$

где $C_{C_{i,k}}$ — множество управляемых переменных, состояние которых соответствует альтернативе a_k .

Тогда сначала необходимо проверить следующее условие:

$$\exists S_{ПЗ} \equiv \exists a_k \in B_a, \quad (2)$$

где $S_{ПЗ}$ — решение задачи; B_a — база аксиом.

Если условие (2) выполняется, задача является решенной. Если оно невыполнимо, то для каждой альтернативы a_k формулируется подзадача m со своими вариантами ее разрешения $a_{km} \subset a_k$ как

$$\Gamma_{ПЗm}(t): \{C_{ckm \text{ тек}}\} \xrightarrow{t \rightarrow \min} \{C_{ckm \text{ тр}}\},$$

где $\Gamma_{ПЗm}$ — целевая функция m -й ПЗ, сформулированная из альтернативы $a_m \in a_k$; $C_{ckm \text{ тек}}$, $C_{ckm \text{ тр}}$ — текущие и требуемые управляемые переменные в новой ПЗ соответственно.

Далее для новой задачи проверяется условие (2). Если оно не выполняется, то проводится дальнейшая декомпозиция ПЗ. Если условие выполнимо, то для данного критерия процесс прекращается. Выбирается следующий критерий и процесс повторяется.

Так происходит формирование множества ЭЗ, на основе решения которых на последующих этапах алгоритма формируется согласованное решение исходной ЗНВВ.

Этап 3. Динамическая модель текущей ситуации необходима, во-первых, для определения критериев в промежуточных и элементарных задачах и их ситуационных весов, а во-вторых, для реализации работы подсистем эвристики, диагностики и управления [4]. В общем случае модель текущей ситуации представляет собой совокупность фона и объекта проблемы [3]. Для того чтобы она была динамической необходимо определить алгоритм изменения всех переменных модели.

В общем виде модель текущей ситуации можно представить в виде выражения (1). Однако для более полного описания модели следует преобразовать его к следующему виду:

$$\frac{\partial C_i}{\partial t} = f_c(C_i, t, c_0); \quad \frac{\partial U_n}{\partial t} = f_u(U_n, t); \quad M_{c. \text{ тек}} = \{C_{ij}\} \cup \{U_n\},$$

где t — текущее время; f_c — функция изменения управляемых переменных; c_0 — текущее управляющее воздействие на объект проблемы до вмешательства лица, принимающего решения (ЛПР); f_u — функция изменения неуправляемых переменных.

Определение модели текущей ситуации позволит предсказать поведение системы и скорректировать управляющее воздействие на переменные в целях приведения их в требуемое состояние.

Этапы 4–7 раскрывают процедуру сбора необходимой и достаточной информации для формирования общего решения задачи.

Этап 4. Для каждой ПЗ и ЭЗ необходимо определить критерии ее решения, а в случае, когда их несколько — объявить вес каждого. Весовые коэффициенты критериев расставляются в зависимости от ситуации, в которой существует каждая задача. Причем, как критериев, так и альтернатив в каждой задаче может быть несколько. Тогда можно записать, что каждой задаче соответствует множество альтернатив $\{a_{31}, a_{32}, \dots, a_{3k}\}$, $k = \overline{1, K}$, где K — число альтернатив в данной задаче, и множество критериев $\{R_{31}, R_{32}, \dots, R_{3s}\}$, $s = \overline{1, S}$, где S — число критериев.

Для многокритериального выбора в промежуточных задачах необходимо каждому критерию присвоить свой вес b_{R_s} .

Тогда основная задача на данном этапе — поставить в соответствие модели текущей ситуации вес критериев каждой задачи:

$$\forall M_{c. \text{ тек}} \Leftrightarrow B \exists \{b_{R_s}\}, s = \overline{1, S}. \quad (3)$$

Причем выражение (10) отражает однозначное и единственное соответствие для каждой модели ситуации.

Этап 5. На данном этапе необходимо определить необходимый и достаточный объем входной информации для однозначного определения факта существования альтернатив для каждой ЭЗ. Иначе говоря, после выделения ЭЗ из общей задачи и выбора альтернатив для каждой из них необходимо проверить возможность существования той или иной альтернативы в текущей ситуации.

Для этого необходимо собрать информацию, наличие которой будет свидетельствовать о существовании альтернативы, т. е. определить ее объем. Тогда объем информации для решения m -й ЭЗ будет определяться как

$$V_{иэм} \Leftrightarrow \exists \{a_{1m}, a_{2m}, \dots, a_{km}\}, m = \overline{1, Z};$$

$$V_{иэм} = V_{иэ1m} \cup V_{иэ2m} \cup \dots \cup V_{иэkm},$$

где Z — число ЭЗ.

В то же время объем информации для определения каждой альтернативы должен отвечать следующему условию:

$$V_{иэkm} \Leftrightarrow (\exists a_{km} | \forall R_{эпрm}),$$

где $R_{эпрm}$ — множество критериев m -й ЭЗ.

Определение необходимого и достаточного объема информации для каждой ЭЗ позволяет перейти непосредственно к процедуре информационной фильтрации.

Этап 6. Прежде чем определить правила покрытия информационного сбора информации необходимо обозначить те объекты окружающего мира, которые будут ее составлять. Для выделения и полного определения альтернативы и задачи достаточно описать ее понятиями.

Определение 1. Под понятием будем понимать класс сущностей, объединяемых на основе общности атрибутивных структур [5]. Любое простое понятие можно описать тройкой: имя, интенционал, экстенционал. Имя понятия — это любой идентификатор; интенционал понятия — множество его свойств с областями их определения, экстенционал понятия — совокупность кортежей значений, удовлетворяющих интенционалу [5].

Каждое понятие описывается множеством существенных признаков, отвечающих требованиям полноты и неизбыточности [5]:

$$P_{\text{по}} = \{X_{\text{по}1}, X_{\text{по}2}, \dots, X_{\text{по}h}\},$$

где $P_{\text{по}}$ — понятие конкретной предметной области; $X_{\text{по}h}$ — существенные признаки $P_{\text{по}}$.

Для представления понятий в ЭЗ расширим определение 1 и представим имя понятия как его схему

$$\text{shm}P_{\text{по}} = \langle D_p, H_p, V_p \rangle,$$

где D_p — множество дифференциальных признаков, характеризующих содержание понятий; H_p — множество характеристических признаков, позволяющих отличать сущности в объеме одного понятия; V_p — множество валентных признаков, обеспечивающих бинарные связи между понятиями.

Определение 2. Конкретное значение сущности или ее признака — есть денотат этого объекта [5].

В соответствии с определением 2 экстенционал понятия $\text{ext}P_{\text{по}}$ будет содержать множество его допустимых денотатов

$$\text{ext}P_{\text{по}} = \{e_1, e_2, \dots, e_n\},$$

где e_n — денотаты понятия $P_{\text{по}}$, определяющиеся как

$$e_n = (\{y_1^n, y_2^n, \dots, y_m^n\}, Y_n),$$

где y_m^n — конкретное значение денотата сущности Y_n понятия $P_{\text{по}}$.

В свою очередь, интенционал понятия $\text{int}P_{\text{по}}$ можно представить как множество дифференциальных признаков

$$\text{int}P_{\text{по}} = \{(D_{p1}, d^1), (D_{p2}, d^2), \dots, (D_{pg}, d^g)\},$$

где D_{pg} — g -й дифференциальный признак; d^g — значение признака D_{pg} .

Тогда понятие можно записать в виде кортежа

$$P_{\text{по}} = \langle \text{int}P_{\text{по}}, \text{ext}P_{\text{по}}, \text{shm}P_{\text{по}} \rangle.$$

Таким образом, на этапе сбора информации необходимо определить компоненты понятий, составляющих альтернативы ЭЗ, т. е. выделить соот-

ветствующие признаки и определить их конкретные значения. Причем это нужно проделать для каждого критерия. При этом у альтернативы, присутствующей в ЭЗ, при разных критериях будет различное значение некоторых признаков.

Этап 7. Фильтрация собранной информации по своей сути является процедурой абстракции. Под абстракцией здесь понимается выделение существенных признаков и связей понятия (положительной информации), используемых при решении ЭЗ по определенному критерию, и игнорирование несущественных (отрицательной информации) [5].

Известны четыре типа абстрагирования понятий: агрегация, обобщение, типизация и ассоциация [5]. Анализ данных типов абстрагирования показал, что для организации автоматического процесса информационной фильтрации необходимо проводить абстрагирование в два шага. На первом шаге абстрагирование выполняется путем ассоциации. В данном случае для определения положительных дифференциальных признаков $\text{ext}P_{\text{по}}^+$ необходимо получить декартово произведение экстенционалов эталонного $\text{ext}P_{\text{по}0}$ и исследуемого $\text{ext}P_{\text{по}i}$ понятий

$$\text{ext}P_{\text{по}}^+ = \text{ext}P_{\text{по}i} \times \text{ext}P_{\text{по}0}.$$

В результате первого шага проводится "грубая" фильтрация информации. Для "точной" фильтрации необходимо выполнить абстракцию типизацией. На втором шаге проводятся операции со всеми компонентами понятий, а именно анализ на точное соответствие исследуемых понятий эталонным:

$$\text{int}P_{\text{по}}^+ = \text{int}P_{\text{по}i} = \text{int}P_{\text{по}0};$$

$$\text{shm}P_{\text{по}}^+ = \text{shm}P_{\text{по}i} = \text{shm}P_{\text{по}0};$$

$$\text{ext}P_{\text{по}}^+ = \text{ext}P_{\text{по}i} \amalg \text{ext}P_{\text{по}0}.$$

Здесь знак \amalg означает операцию размеченного объединения.

Таким образом, в результате информационной фильтрации в решении задачи присутствуют только те альтернативы, определяющие понятия которых позволяют сделать однозначный выбор в ЭЗ на основании конкретного критерия.

В процессе выполнения этапов 8—11 формируется общее решение задачи в обобщенном алгоритме.

Этап 8. Решение элементарных задач осуществляется по так называемым ситуационным аксиомам. Сущность ситуационной аксиомы заключается в том, что для каждой ситуации при условии существования множества критериев в ЭЗ для каждого критерия из множества альтернатив существует такая, которая в текущей ситуации является аксиомой, т. е.

$$\forall R_{\text{эзпр}m}^s \in \{R_{\text{эзпр}m}\} \rightarrow \exists a_{ms}^\alpha \in \{a_{ms} | R_{\text{эзпр}m}^s\},$$

где $R_{\text{эзпр}m}^s$ — s -й критерий в m -й ЭЗ; a_{ms}^α — аксиома для s -го критерия в m -й ЭЗ.

В каждой предметной области и текущей ситуации (критерии) существует только одна своя аксиома, поэтому принятие решения в ЭЗ можно осуществить автоматически. Для этого нужно провести операцию аналогии альтернатив ЭЗ с содержанием базы аксиом. Причем в этой базе аксиомы представлены не как альтернативы, а как варианты решения задачи при соблюдении определенных условий. В данном случае условием является критерий решения ЭЗ. Тогда аксиому можно представить как

$$a_{ms}^\alpha = (a_k^{ax} | C_k^{ax} = R_{\text{эз.пр}m}^s),$$

где a_k^{ax} — k -я аксиома из базы аксиом; C_k^{ax} — условие, при котором a_k^{ax} является аксиомой.

В случае если ни одной из альтернатив не соответствует ни одна аксиома, то задача не является ЭЗ в данной предметной области.

Эман 9. Идеальный вариант, когда все принятые решения по всем ЭЗ согласуются между собой, не всегда вероятен. Поэтому предусмотрена проверка всех ЭР на совместимость и отсутствие конфликта.

Допустим, что результатом решения i -й ЭЗ явилась альтернатива a_{ij}^α , k -й задачи — a_{km}^α , q -й задачи — a_{qp}^α . В свою очередь, i -я, k -я и q -я ЭЗ формируют три альтернативы задачи высшего уровня иерархии для критерия R_{31} . Предположим, что таких критериев три.

Для принятия решения в задаче высшего уровня необходимо оценить согласованность решений ЭЗ. В данной постановке вопроса под согласованностью решений ЭЗ будем понимать непротиворечие аксиом по разным критериям для одной задачи.

В рассматриваемом примере возможны три варианта ситуации принятия решения: согласованное, несогласованное, конфликтное.

Первый вариант предусматривает, что при различных критериях были выбраны одинаковые аксиомы. Тогда решение задачи принимается по максимальному количеству одинаковых аксиом. То есть, если решением i -й ЭЗ по критерию R_{31} и k -й ЭЗ по критерию R_{32} явилась аксиома a_{fg}^α , $f = i = k$, то эта аксиома будет решением задачи высшего уровня. В этом случае решения ЭЗ согласованы между собой.

В обобщенном виде согласованное решение задачи можно записать следующим образом:

$$Q_p(x, y) = \sum_{i=1}^{S_p} \sum_{j=1}^{Z_p} (a_{xy}^\alpha \times a_{ms}^\alpha); x, m = \overline{1, S_p};$$

$$y, s = \overline{1, Z_p}; Q_p(x, y) \leq S_p;$$

$$S_{\text{пз}q} = \max Q_p(x, y),$$

где S_p — число критериев в промежуточной задаче; Z_p — число ЭЗ в промежуточной задаче; $S_{\text{пз}q}$ — решение промежуточной задачи.

Эман 10. Если решения ЭЗ не являются согласованными, то они могут находиться в состоянии конфликта, т. е. когда существуют несколько $Q(x, y)$, значения которых равны. Следовательно, нельзя однозначно определить SR_p . Несогласованными решения ЭЗ будут считаться в том случае, если $Q(x, y) = 0$, т. е. все аксиомы ЭЗ в данной задаче различны.

Если провести экспертные оценки всех полученных аксиом по всем критериям, то задачу можно решить любым известным методом, например методом взвешенной выборки.

Для разрешения несогласованности и конфликта решений необходимо сформировать ситуационные предпочтения, т. е. определить множество понятий предпочтения $P_{\text{птс}}$ и отношений предпочтения $r_{\text{птс}}$ модели текущей ситуации, которые ее описывают в большей мере и составляют модель предпочтений $M_{\text{птс}}$:

$$M_{\text{с.тек}} \Rightarrow M_{\text{птс}} \{P_{\text{птс}}, r_{\text{птс}}\}.$$

На основе ситуационных предпочтений формируются правила разрешения конфликта и несогласованности при принятии решения.

Для устранения конфликта необходимо определить, какой из критериев в текущей ситуации предпочтительнее других (имеет больший вес). Очевидно, что определение таких весов нужно осуществлять на основе сформированных понятий предпочтения, используя механизмы логического вывода.

Если предпочтения критериев определены, то конфликт разрешается путем выбора того $Q(x, y)$, в котором большинство альтернатив, дающих при перемножении единицу, принадлежат критерию с большим весом (предпочтением)

$$S_{\text{пз}q} = \max\{Q_p(x, y) | \exists R_{3i} > \forall R_{3j}\}, (i \neq j) \in S.$$

В случае несогласованности элементарных решений выбор альтернативы (для задачи высшего уровня решения ЭЗ аксиомами не являются) осуществляется из вариантов того критерия, вес которого в текущей ситуации больше. Сами альтернативы так же, как и критерии взвешиваются на основе определения степени отношения к модели $M_{\text{птс}}$. Далее выбирается та альтернатива, у которой вес w_a наибольший:

$$S_{\text{пз}q} = \max\{w(a_{ms}) | R_{3i} > \forall R_{3j}\},$$

где $w(a_{ms})$ — вес альтернативы a_{ms} .

На данном этапе наибольшие сложности в реализации вызывает процедура взвешивания критериев и альтернатив. В общем случае ее можно описать операциями, рассмотренными на этапе 10.

Эман 11. Общее решение задачи выполняется после разрешения всех ситуаций наличия конфликтных и несогласованных решений. В процессе решения промежуточных задач степень сложности задачи повышается, заканчиваясь на высшем уровне.

не иерархии. По своей сути процесс выработки общего решения практически не отличается от решения промежуточной задачи. Особенность последнего этапа состоит в том, что в нем каждая полученная альтернатива проверяется на соответствие базе аксиом. Цель такой проверки заключается в контроле итоговой согласованности принятия решения.

В случае если принятое решение является подобным какой-либо аксиоме, а условие существования аксиомы не соответствует поставленной задаче, то принимается решение об ошибочном выборе альтернативы, проводится новый синтез модели текущей ситуации и процесс решения задачи повторяется.

Тогда принятие общего решения задачи можно определить как

$$SR = \max\{S_{ПЗ} \mid \forall a^{\alpha} \in B_a = \{true\}\}.$$

Таким образом, в результате выполнения всех этапов процедуры принятия решения, становится возможным автоматически или автоматизировано выработать общее решение задачи, которое является согласованным, не содержащим неразрешенные конфликты и учитывающим текущую ситуацию, в которой существует задача.

Заключение

В основу представленной подсистемы нейтрализации внешнего воздействия на ПЭС положена искусственная иммунная система, которая в процессе своего функционирования оценивает текущее внешнее воздействие, формирует его динамическую модель, выбирает или вырабатывает вариант противодействия. Она позволяет автоматизировать процесс принятия решений по нейтрализации внешних воздействий и при этом обеспечить максимальную эффективность функционирования и конкурентоспособность ПЭС.

Список литературы

1. **Рассел С., Норвиг П.** Искусственный интеллект: современный подход, 2-е изд.: пер. с англ. М.: Вильямс, 2006. 1408 с.
2. **Сербулов Ю. С., Степанов Л. В., Глухов Д. А.** Моделирование рыночных барьеров на основе искусственной иммунной системы // Лесотехнический журнал. 2013. № 4 (12). С. 223—231.
3. **Акофф Р.** Искусство решения проблем. М.: Мир, 1982.
4. **Мальшев В. А.** Метод принятия решения в условиях избыточности информации // Вестн. Воронеж. гос. техн. ун-та. 2006. Т. 2, № 3. С. 82—89.
5. **Гаскаров Д. В.** Интеллектуальные информационные системы. М.: Высш. школа, 2003.

Y. S. Serbulov, Professor, **D. A. Gluhov**, Associate Professor, Voronezh State Academy of Forestry, **V. A. Malyshev**, Deputy Chief, Air Force Academy Them. Prof. Zhukovsky and Yuri Gagarin

Model Partition Neutralize External Effects on Industrial and Economic Systems in a Competitive Environment

The solution of the problem of increasing conflict sustainability of industrial and economic systems to external influences competitors. The problem is solved by developing subsystem neutralize external influences based on artificial immune system. This subsystem during operation should solve the following intermediate objectives: to assess the current impact and the formation of its dynamic model; selection or development option against external influences; check on the consistency of the results of solving the first two problems; to reach a common decision on the opposition. The basis of the functioning of a subsystem on a model containing two types of components: database and special modules or agents. The most difficult is the module implements the algorithm neutralize external influences operating process which can be represented as a series of eleven stages of decision-making procedures. As a result of their implementation, it is possible to automatically or automated develop a common solution to the problem, which is consistent not contain unresolved conflicts and taking into account the current situation in which there is a problem.

Keywords: industrial and economic system, artificial immune system, the neutralization of external influences, modeling module, the base models

References

1. **Rassel S., Norvig P.** Iskustvennyy Intellect: sovremenny podkhod (Artificial intelligence: a modern approach). 2-e izd, per. s angl. M.: Vil'yams, 2006. 1408 p.
2. **Serbulov Yu. S., Stepanov L. V., Glukhov D. A.** Modelirovaniye rynochnykh bar'eroev na osnove iskustvennoy immunnoy sistemy (Modeling market barriers based artificial immune system). *Lesotekhnicheskij zhurnal*. 2013. No. 4 (12). P. 223—231.
3. **Akoff R.** Iskustvo resheniya problem (The art of solving problems). M.: Mir, 1982.
4. **Malyshev V. A.** Metod prinyatiya resheniya v usloviyakh izbytochnosti informatsii (Method of decision-making under conditions of redundancy). *Vestn. Voronezh. gos. tekhn. unta*. 2006. T. 2, No. 3. P. 82—89.
5. **Gaskarov D. V.** Intellektual'nye informatsionnye sistemy (Intelligent Information systems). M.: Vyssh. shkola, 2003.

ЖУРНАЛ В ЖУРНАЛЕ



НЕЙРОСЕТЕВЫЕ ТЕХНОЛОГИИ

№9
СЕНТЯБРЬ
2014

Главный редактор:

ГАЛУШКИН А.И.

Редакционная коллегия:

АВЕДЬЯН Э.Д.
БАЗИЯН Б.Х.
БЕНЕВОЛЕНСКИЙ С.Б.
БОРИСОВ В.В.
ГОРБАЧЕНКО В.И.
ЖДАНОВ А.А.
ЗЕФИРОВ Н.С.
ЗОЗУЛЯ Ю.И.
КРИЖИЖАНОВСКИЙ Б.В.
КУДРЯВЦЕВ В.Б.
КУЛИК С.Д.
КУРАВСКИЙ Л.С.
РЕДЬКО В.Г.
РУДИНСКИЙ А.В.
СИМОРОВ С.Н.
ФЕДУЛОВ А.С.
ЧЕРВЯКОВ Н.И.

Иностранные члены редколлегии:

БОЯНОВ К.
ВЕЛИЧКОВСКИЙ Б.М.
ГРАБАРЧУК В.
РУТКОВСКИЙ Л.

Редакция:

БЕЗМЕНОВА М.Ю.
ГРИГОРИН-РЯБОВА Е.В.
ЛЫСЕНКО А.В.
ЧУГУНОВА А.В.

Гриняк В. М., Головченко Б. С., Девятисильный А. С.

Нейронечеткая обучаемая система распознавания опасного сближения судов 68

Чеботарев В. В.

Логическая нейронная сеть в основе контроля данных о перевозке грузов на железнодорожном транспорте 74

В. М. Гриняк¹, канд. техн. наук, доц., e-mail: Viktor.Grinyak@vvsu.ru,
Б. С. Головченко¹, инженер, e-mail: Boris.Golovchenko@vvsu.ru,
А. С. Девятисильный², д-р техн. наук, гл. науч. сотр., e-mail: devyatis@iacp.dvo.ru

¹ Владивостокский государственный университет экономики и сервиса

² Институт автоматизации и процессов управления ДВО РАН

Нейронечеткая обучаемая система распознавания опасного сближения судов

Статья посвящена проблеме управления движением судов на морских акваториях. Рассматривается модель относительного движения двух судов и предлагается алгоритм выработки различных тревожных сигналов, соответствующих вербальному уровню опасности "судно — судно". В основу определения уровня опасности положен факт маневрирования судна и время, оставшееся до недопустимого сближения. Предложена конфигурация нейронечеткой сети для принятия решения об уровне опасности движения, обсуждаются способы обучения сети, в том числе на реальных данных о движении судов. Статья сопровождается результатами вычислительного эксперимента, демонстрирующего работу системы в типичных ситуациях.

Ключевые слова: управление движением судов, опасное сближение, траектория движения, маневрирование судна, нейронечеткая система

Введение

Управление коллективным движением судов в акваториях морских портов представляет собой особый раздел науки об управлении [1—5]. На практике оно реализуется системами управления движением судов (СУДС) — специализированными системами, основной задачей которых является предотвращение опасных ситуаций, в частности, недопущение опасного сближения судов [1, 4]. Информационной базой современных СУДС являются двухкоординатные радиолокационные станции (РЛС) кругового обзора, дополняемые средствами спутниковой навигации — транспондерами автоматической идентификационной системы (АИС).

В основе распознавания опасного сближения судов лежит оценка параметров траектории движения каждого судна (координат, скоростей и т. д.) и их экстраполяция [5]. Если суда идентифицированы как опасно сближающиеся, система управления движением генерирует тревожный сигнал и рекомендации по изменению траектории движения.

Прогнозирование движения судна и, соответственно, управленческое решение, обеспечивающее его безопасность, всегда несут в себе элемент неопределенности. Это требует формализации понятия "опасная ситуация" с выделением различных уровней опасности типа: "очень опасный", "опасный", "безопасный" и т. п. Такой подход предоставляет возможность судоводителю и оператору береговой СУДС упорядочить свои действия: принимать различные типы решений в ситуациях с различным уровнем опасности и, тем самым, понизить степень неопределенности при принятии конкретного решения [6, 7].

Настоящая работа посвящена исследованию возможности создания информационной системы оценки безопасности коллективного движения морских судов, обеспечивающей заблаговременное распознавание опасных ситуаций и оценку уровня опасности с использованием идей нечетких систем и нейронных сетей с обучением.

Основные модельные представления и постановка задачи

Традиционным подходом при моделировании навигационной безопасности коллективного движения является построение модели безопасности "судно — судно" для каждой пары судов (такой подход применяется во многих известных алгоритмах предупреждения столкновений [5, 8, 9]).

Рассмотрим два судна с координатами $x^{(1)}, y^{(1)}$ и $x^{(2)}, y^{(2)}$ и скоростями $v_x^{(1)}, v_y^{(1)}$ и $v_x^{(2)}, v_y^{(2)}$. Будем описывать их коллективное движение набором величин $s = (r_x, r_y, v, \eta_v, w_v)^T$ — вектором состояния коллективного движения двух судов, где $r_x = x^{(2)} - x^{(1)}$, $r_y = y^{(2)} - y^{(1)}$ — координаты вектора относительного положения судов; $v = \sqrt{(v_x^{(1)} - v_x^{(2)})^2 + (v_y^{(1)} - v_y^{(2)})^2}$ — скорость относительного движения судов; η_v — направление вектора скорости относительного движения судов; $w_v = d\eta_v/dt$ — скорость изменения угла η_v (рис. 1).

Опыт практического судовождения показывает, что для обеспечения безопасного плавания наибольшее значение имеет соблюдение некой "зоны

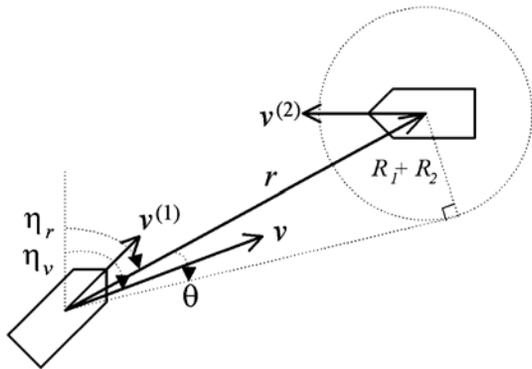


Рис. 1. Модель относительного движения пары "судно — судно"

безопасности" возле судна (называемой также "корабельным доменом" [5]), в которую другие суда не должны допускать вторжение. В настоящей работе рассматривается корабельный домен статического типа, жестко привязанный к судну с номером m и интерпретируемый окружностью заданного радиуса R_m .

Потенциально опасное сближение двух судов может быть формализовано следующим образом:

$$|\eta_v - \eta_r| < \theta; \quad (1)$$

$$|w_v - w_r| < \dot{\theta}; \quad (2)$$

$$0 < T < T^*. \quad (3)$$

Здесь $|r| = \sqrt{r_x^2 + r_y^2}$ — текущее расстояние между судами; $\eta_r = \arctg(r_x/r_y)$ — азимут вектора r ; $w_r = d\eta_r/dt = \frac{r_x v \cos \eta_v - r_y v \sin \eta_v}{|r|^2}$ — скорость измене-

ния азимута вектора r ; $\theta = \arcsin((R_1 + R_2)/|r|)$ — угол, определяемый расстоянием между судами и размерами доменов (считается, что в безопасном состоянии корабельные домены не должны "вторгаться" в область друг друга); $\dot{\theta} = d\theta/dt =$

$= -\frac{R_1 + R_2}{|r|^2 \sqrt{1 - (R_1 + R_2)^2/|r|^2}} \frac{d|r|}{dt}$ — скорость изме-

нения угла θ ; $\frac{d|r|}{dt} = -\frac{r_x v \sin \eta_v + r_y v \cos \eta_v}{|r|}$ — ско-

рость изменения расстояния между судами; $T = -\frac{|r|^2}{r_x v \sin \eta_v + r_y v \cos \eta_v}$ — приближенное время,

оставшееся до максимального сближения судов; T^* — пороговое значение для времени T .

Условие (1) формализует опасную ситуацию при равномерном и прямолинейном движении судов; условие (2) дополняет его в случае, если суда маневрируют; условие (3) отбирает из общего массива

лишь те суда, время до сближения которых меньше порогового.

Таким образом, в настоящей работе ставится задача об оценке вектора состояния коллективного движения двух судов $s = (r_x, r_y, v, \eta_v, w_v)^T$ и формулировке вывода о той или иной степени опасности сложившейся навигационной ситуации с учетом условий (1)–(3).

Концепция решения

Вектор s может быть найден на базе измерительной информации береговой РЛС или по данным GPS различными способами (см., например, [8, 10]).

Что касается вывода о той или иной степени опасности ситуации, то здесь следует отметить следующий факт. Маневрирующие и неманеврирующие суда с точки зрения безопасности коллективного движения имеют принципиальные различия. Во-первых, при внешнем наблюдении полностью достоверный прогноз траектории маневрирующего объекта невозможен. Во-вторых, на практике маневрирование судна, как правило, свидетельствует о попытке судоводителя придать движению безопасный характер и о его контроле над ситуацией [7, 10]; поэтому с точки зрения внешнего наблюдения для маневрирующего объекта вербальный уровень опасности заведомо ниже, чем для неманеврирующего.

Некоторые принципы вербальной оценки уровня опасности были предложены авторами в работах [6, 7, 10], близкие по смыслу методики описаны также другими авторами, например в работе [11]. Дискретная оценка уровня опасности ситуации типа "очень опасная", "опасная", "почти безопасная" позволяет привлечь внимание судоводителя (диспетчера) в первую очередь к наиболее опасным ситуациям. Вместе с тем, при высокой интенсивности движения может оказаться слишком много судов с одинаковым уровнем опасности, что повышает возможность ошибочных управленческих решений. В этом случае становится актуальной задача представления уровня опасности непрерывной величиной. Здесь оказываются продуктивными идеи, положенные в основу задач систем нечеткой логики.

Пусть $A = \frac{|\eta_v - \eta_r|}{\theta}$ — оцененное отношение величины $|\eta_v - \eta_r|$ к значению θ . Введем лингвистическую переменную P_A "оценка отношения A " с терминами "большое" и "малое" и функциями принадлежности типа "дополнение", определенными на универсальном множестве $A \in [0, 2]$:

$$\mu_{\text{малое}}(A) = 1 - \frac{1}{1 + \exp(-a_A(A - c_A))},$$

$$\mu_{\text{большое}}(A) = \frac{1}{1 + \exp(-a_A(A - c_A))},$$

где a_A, c_A — настраиваемые параметры функций принадлежности.

В случае, если движение судов происходит прямолинейно и равномерно, терм "малое" соответствует ситуации, когда возможно опасное сближение судов, а терм "большое" — безопасной ситуации.

Пусть $B = \frac{|w_v - w_r|}{\dot{\theta}}$ — оцененное отношение ве-

личины $|w_v - w_r|$ к значению $\dot{\theta}$. Введем лингвистическую переменную P_B "оценка отношения B " с термами "большое" и "малое" и функциями принадлежности типа "дополнение", определенными на универсальном множестве $B \in [0, 2]$:

$$\lambda_{\text{малое}}(B) = 1 - \frac{1}{1 + \exp(-a_B(B - c_B))},$$

$$\lambda_{\text{большое}}(B) = \frac{1}{1 + \exp(-a_B(B - c_B))}.$$

Здесь a_B, c_B — настраиваемые параметры функций принадлежности.

В данном случае величина B и термы "малое" и "большое" характеризуют интенсивность маневрирования.

Для описания величины T (приближенное время, оставшееся до максимального сближения судов) введем лингвистическую переменную P_T "оценка величины T " с термами "малое", "среднее" и "большое" и функциями принадлежности типа "кластер", определенными на универсальном множестве $T \in [0, 2000]$ с:

$$v_{\text{малое}}(T) = 1 - \frac{1}{1 + \exp(-a_1(T - c_1))};$$

$$v_{\text{среднее}}(T) = \exp\left(-\frac{(T - c_2)^2}{a_2}\right);$$

$$v_{\text{большое}}(T) = \frac{1}{1 + \exp(-a_3(T - c_3))}.$$

Здесь $a_1, c_1, a_2, c_2, a_3, c_3$ — настраиваемые параметры функций принадлежности.

Описание лингвистической переменной P_T тремя термами соответствует трем принятым на практике состояниям времени максимального сближения [12]. Терм "малое" определяет время, когда существует возможность принять только одно решение, которое поможет избежать столкновения. Терм "среднее" — время, которое требуется для грамотного проведения оптимального маневра, т. е. время, когда начинать маневр уже не рано, но еще есть время "исправить" результат ошибочного маневрирования, есть время на "вторую попытку";

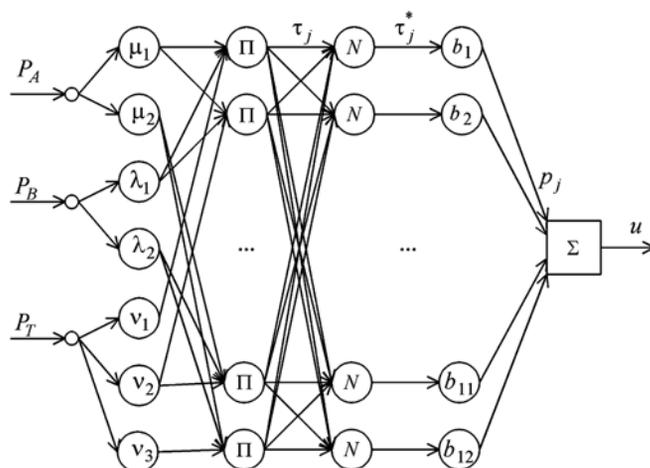


Рис. 2. Схема нейронечеткой сети, определяющей уровень опасности навигационной ситуации "судно — судно"

это период, когда критическая ситуация пока еще не возникла. Терм "большое" описывает время, когда еще нет никакого смысла предпринимать какие-либо действия, так как ситуация может измениться и это потребует, возможно, исполнения совсем другого маневра.

Величины P_A, P_B и P_T (вход) обрабатываются нейронечеткой сетью, показанной на рис. 2, на выходе которой формируется числовое значение $u \in [0, 3]$ — уровень опасности навигационной ситуации "судно — судно"; значение $u = 0$ соответствует наименьшему уровню опасности, $u = 3$ — наибольшему. Сеть состоит из пяти слоев [13] (рис. 2).

В узлах первого слоя $\mu_1, \mu_2, \lambda_1, \lambda_2, v_1, v_2, v_3$ вычисляются значения функции принадлежности $\mu_{\text{большое}}, \mu_{\text{малое}}, \lambda_{\text{большое}}, \lambda_{\text{малое}}, v_{\text{большое}}, v_{\text{среднее}}, v_{\text{малое}}$ соответственно. Узлы Π второго слоя (всего 12 узлов) соответствуют посылкам 12 возможных нечетких правил, комбинирующих все возможные значения величин P_A, P_B и P_T :

1. $P_A = \text{"большое"}$ И $P_B = \text{"большое"}$ И $P_T = \text{"большое"}$;
2. $P_A = \text{"большое"}$ И $P_B = \text{"большое"}$ И $P_T = \text{"среднее"}$;
3. $P_A = \text{"большое"}$ И $P_B = \text{"большое"}$ И $P_T = \text{"малое"}$;
- ...
10. $P_A = \text{"малое"}$ И $P_B = \text{"малое"}$ И $P_T = \text{"большое"}$;
11. $P_A = \text{"малое"}$ И $P_B = \text{"малое"}$ И $P_T = \text{"среднее"}$;
12. $P_A = \text{"малое"}$ И $P_B = \text{"малое"}$ И $P_T = \text{"малое"}$.

Каждый узел второго слоя соединен с теми узлами первого слоя, которые формируют посылки соответствующего правила. Выходом каждого узла второго слоя является степень выполнения j -го правила τ_j , которая рассчитывается как произведение входных сигналов.

Узлы N третьего слоя рассчитывают относительную степень выполнения каждого нечеткого правила по формуле

$$\tau_j^* = \frac{\tau_j}{\sum_{k=1}^{12} \tau_k}.$$

Узлы b_1, b_2, \dots, b_{12} четвертого слоя формируют заключения нечетких правил; фактически b_j — это значения выхода сети при однозначном выполнении только j -го правила. Каждый узел соединен с одним узлом третьего слоя и рассчитывает вклад одного нечеткого правила в выход сети по формуле $p_j = b_j \tau_j^*$.

Единственный узел пятого слоя агрегирует результат, полученный по разным правилам, суммируя вклады всех правил $u = \sum_{k=1}^{12} p_j$.

Обучение системы

Обучение нейронечеткой сети (рис. 2) состоит в настройке параметров функций принадлежности $a_A, c_A, a_B, c_B, a_1, c_1, a_2, c_2, a_3, c_3$ и коэффициентов b_1, b_2, \dots, b_{12} узлов четвертого слоя. Обучение может быть проведено с применением трех различных стратегий.

Стратегия 1. Обучение полностью экспертным способом. В этом случае все коэффициенты назначаются экспертом.

Стратегия 2. Обучение на обучающей выборке с экспертным формированием заключений нечетких правил. В этом случае коэффициенты b_j назначает эксперт, а параметры функции принадлежности определяются настройкой системы на обучающей выборке.

Стратегия 3. Обучение полностью на обучающей выборке. В этом случае все коэффициенты системы определяются настройкой на обучающей выборке.

Обучающая выборка формируется следующим образом. Моделируется решение задачи с оценкой условий (1)—(3) на различных траекториях. При безопасном движении считается, что выход системы $u = 0$. Если суда опасно сближаются при прекращении маневрирования в текущий момент, то $u = 1$ ("почти безопасная" ситуация); если суда недопустимо сближаются при продолжении маневрирования, то $u = 2$ ("опасная" ситуация); если суда сближаются при прямолинейном и равномерном движении, то $u = 3$ ("очень опасная" ситуация). Накопивая данные для различных траекторий и скоростей движения судов, формируют общую обучающую выборку "вход—выход", на базе которой обучают сеть (рис. 2), пользуясь известными методами обучения сетей такого типа [13, 14].

Результаты численного моделирования

Численное моделирование рассматриваемой задачи проводили в условиях, приближенных к характерной навигационной обстановке залива Петра Великого.

Обучение системы происходило в рамках стратегии 2 при следующих значениях параметров b_j , соответствующих 12 возможным нечетким правилам: $b_1, b_2, b_3, b_4, b_7, b_{10} = 0$; $b_5, b_8, b_9 = 1$; $b_6, b_{11} = 2$; $b_{12} = 3$. Обучающая выборка формировалась на реальных данных о движении судов в акватории порта Владивосток (рис. 3, см. четвертую сторону обложки), содержала данные о примерно 5000 траекторий и состояла примерно из 1 млн записей "вход—выход".

Следующий численный пример демонстрирует работу обученной нейронечеткой системы (см. рис. 2). Было принято, что информационной базой береговой СУДС является РЛС с периодом обращения 3 с и среднеквадратичными значениями погрешностей измерения дальности и азимута $\delta_r = 3$ м, $\delta_\varphi = 0,015^\circ$. Скорости движения судов задавали равными 5 м/с, радиус корабельного домена $R_1 = R_2 = 150$ м; число измерений, участвующих в оценке вектора s , было взято равным 20.

На рис. 4 изображены моделируемые траектории движения трех судов: два из них (I и II) движутся прямолинейно и равномерно, третье (III) — маневрирует.

Рис. 5 иллюстрирует определение уровня опасности навигационной ситуации по мере движения судов с течением времени: "судно I — судно III" (левая колонка) и "судно II — судно III" (правая колонка). На рис. 5, а, б показаны значения величины A ; на рис. 5, в, г — значения величины B ; на рис. 5, д, е — значения величины T (т. е. входы нейронечеткой сети). Из рис. 5, в, г видна особенность рассматриваемой задачи — неустойчивость оценки величины B при больших расстояниях между судами в условиях погрешностей измерений.

На рис. 5, ж, з показаны значения величины u (выход нейронечеткой сети). Так, на рис. 5, ж вид-

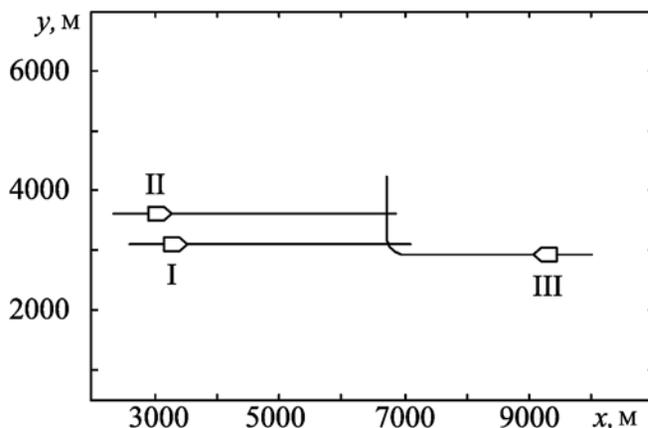


Рис. 4. Моделируемые траектории движения судов

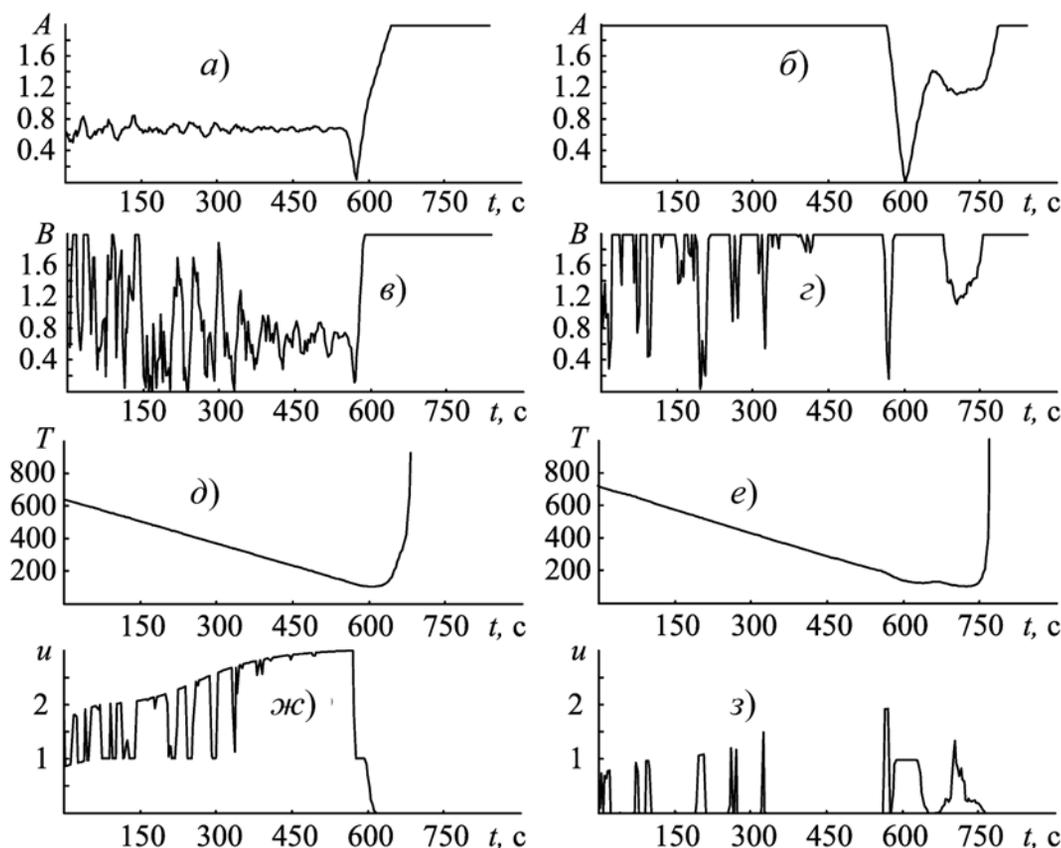


Рис. 5. Результаты численного моделирования работы нечеткой системы распознавания опасного сближения судов

но, что уровень опасности для судов I и III постоянно повышается по мере их сближения, достигая величины $u \approx 3$ при $t = 450$ с. При $t = 550$ судно III начинает маневр уклонения поворотом вправо, после чего уровень опасности "судно I — судно III" быстро снижается до уровня $u \approx 1$ и затем до 0. Из рис. 5, з видно, что суда II и III вначале движутся безопасно. После начала судном III маневра уровень опасности "судно II — судно III" скачкообразно увеличивается до $u \approx 2$, а затем, по мере продолжения поворота судна III, снижается до уровня $u \approx 1$ и затем до 0.

Снижение уровня тревоги с максимального $u \approx 3$ до безопасного $u \approx 1$ происходит почти сразу после начала маневрирования (рис. 5, ж), для судов II и III генерация уровня тревоги $u \approx 2$ также происходит заблаговременно (рис. 5, з). Это свидетельствует о высокой эффективности предлагаемой нечеткой системы оценки состояния безопасности по отношению к проблеме генерации ложных тревог. Результаты моделирования также подтверждают ожидаемый эффект разделения уровней опасности в зависимости от траекторных свойств движения судов.

Заключение

Для оператора (диспетчера) СУДС генерация тревожного сигнала уровня $u > 2$ означает, что необходимо немедленно обратить внимание на си-

туацию и принять решение о вмешательстве/невмешательстве в действия судоводителя. Генерация тревожного сигнала уровня $1 < u < 2$ означает, что хотя ситуация не является полностью безопасной, вмешиваться в нее немедленно не следует: скорее всего, судоводитель сам контролирует обстановку. Другими словами, если одновременно с ситуацией уровня $u < 2$ для других судов имеются ситуации уровня $u > 2$, то оператор СУДС должен работать прежде всего с ними.

Испытание разработанной системы проводили на реальных данных о движении судов по акватории, прилегающей к порту Владивосток. Было выявлено, что доля тревожных сигналов уровня $u < 2$ составляет около 20 %, места их генерации не образуют устойчивых зон. Это говорит о том, что выделение таких сигналов способно заметно снизить нагрузку на операторов СУДС, и свидетельствует об актуальности рассмотренной задачи для практики.

Результаты работы ориентированы на расширение навигационных функций современных систем управления движением судов.

Список литературы

1. ОАО Норфес. URL: <http://www.norfes.ru/>
2. Tam C. K., Bucknall R. Collision risk assessment for ships // Journal of Marine Science and Technology. 2010. Vol. 15. N. 3. P. 257—270.

3. **Мироненко А. А.** Модель программного движения судна в стесненных водах // Мехатроника, автоматизация, управление. 2013. № 2. С. 65—70.

4. **Астреин В. В.** Системы предупреждения столкновения судов, тенденции развития (к 40-летию МППСС-72) // Вестник Астраханского государственного технического университета. Сер. "Морская техника и технология". 2012. № 1. С. 7—17.

5. **Tam Ch. K., Bucknall R., Greig A.** Review of Collision Avoidance and Path Planning Methods for Ships in Close Range Encounters // *Journal of Navigation*. 2009. V. 62, N. 3. P. 455—476.

6. **Гриняк В. М., Головченко Б. С.** Многоуровневая модель идентификации опасных ситуаций на морских акваториях // Территория новых возможностей. Вестник Владивостокского государственного университета экономики и сервиса. 2012. № 4. С. 69—75.

7. **Grinyak V. M., Devyatisilny A. A.** Prediction of emergency situation in marine traffic control // *Journal of System and Computer Sciences International*. 2004. V. 43, N. 3. P. 448—457.

8. **Wilson P. A., Harris C. J., Hong X.** A Line Of Sign Counteraction Navigation Algorithm For Ship Encounter Collision Avoidance // *Journal of Navigation*. 2003. V. 56, N. 1. P. 111—121.

9. **Churkin V. I.** Optimal control of the ship divergence // *Journal of System and Computer Sciences International*. 1999. N. 2. P. 221—226.

10. **Гриняк В. М., Головченко Б. С., Малько В. Н.** Распознавание опасных ситуаций системами управления движением судов // Транспорт: наука, техника, управление. 2011. № 8. С. 42—45.

11. **Егоров И. Б., Логиновский В. А.** Концепция зон навигационной безопасности и ее применение в судовождении // Эксплуатация морского транспорта. 2012. № 3. С. 13—17.

12. **Коноплев М. А.** Применение аппарата нечеткой логики для определения уровня опасности столкновения // Эксплуатация морского транспорта. 2009. № 2. С. 34—39.

13. **Nauk D., Klawonn F., Kruse R.** Foundations of Neuro-Fuzzy Systems. John Wiley & Sons. 1997. 305 p.

14. **Штовба С. Д.** Проектирование нечетких систем средствами MatLab. М.: Горячая линия — Телеком, 2007. 288 с.

V. M. Grinyak¹, Associate Professor, **B. S. Golovchenko**¹, Postgraduate Student,
A. S. Devyatisilny², Professor, e-mail: devyatis@iacp.dvo.ru
¹ Vladivostok State University of Economics and Service,
² Institute of Automation and Control Processes, FEBRAS

Neuro-Fuzzy Decision-Making System for Ships Collision Avoidance

The problem of marine vessel traffic control is discussed in this paper. Two ships relative motion model is watched and algorithm of alarm generation is offered. Authors are justifying need separate ships by alarm level in case high intensity of traffic. Ships alarm level estimation based on ship maneuver detector and time to collision. The decision-making system of alarm level based on neuro-fuzzy logic system (ANFIS) is offered. There are 3 values on system input. The first input value is angle between relative velocity vector and relative coordinate vector with terms "large" and "little". The second value is derivation of the first one with the same terms. The third value is time to collision with 3 terms "little", "average" and "large". The output fuzzy logic system value is alarm level with 4 levels "safe", "almost safe", "dangerous" and "very dangerous". There are 12 rules of neuro-fuzzy logic system. The rules based on principle "if the vessel is maneuvering then alarm level less". It is possible to learn of neuro-fuzzy logic system by expert and by training sample for learning. Numerical results are demonstrating decision-making system effectiveness for typical vessel traffic.

Keywords: vessel traffic control, collision avoidance, trace, ship maneuver, neuro-fuzzy system, ANFIS

References

1. JSC NorFes. URL: <http://www.norfes.ru/>

2. **Tam C. K., Bucknall R.** Collision risk assessment for ships. *Journal of Marine Science and Technology*. 2010. V. 15. N. 3. P. 257—270.

3. **Mironenko A. A.** Model programnogo dvizheniya sudna v stesnyon vodah. *Mehatronika, avtomatizatsiya, upravlenie*. 2013. N. 2. P. 65—70.

4. **Astrein V. V.** Sistemy preduprezhdeniya stolknoveniya sudov, tendentsii pazvitiya (k 40-letiyu MPPSS-72). *Vestnik Astrakhanskogo gosudarstvenogo tekhnicheskogo universiteta*. Ser.: *Morskaya tekhnika i tekhnologiya*. 2012. N. 1. P. 7—17.

5. **Tam Ch. K., Bucknall R., Greig A.** Review of Collision Avoidance and Path Planning Methods for Ships in Close Range Encounters. *Journal of Navigation*. 2009. V. 62, N. 3. P. 455—476.

6. **Grinyak V. M., Golovchenko B. S.** Mnogourovnevaya model identifikatsii opasnih situatsiy na morskikh akvatoriayah. *Territriya novih vozmozhnostey*. *Vestnik Vladivostokskogo gosudarstvennogo universiteta ecinimiki i servisa*. 2012. N. 4. P. 69—75.

7. **Grinyak V. M., Devyatisilny A. S.** Prediction of emergency situations in marine traffic control. *Journal of System and Computer Sciences International*. 2004. V. 43. N. 3. P. 448—457.

8. **Wilson P. A., Harris C. J., Hong X.** A Line Of Sign Counteraction Navigation Algorithm For Ship Encounter Collision Avoidance. *Journal of Navigation*. 2003. V. 56, N. 1. P. 111—121.

9. **Churkin V. I.** Optimal control of the ship divergence. *Journal of System and Computer Sciences International*. 1999. N. 2. Vol. 38. P. 221—226.

10. **Grinyak V. M., Golovchenko B. S., Malko V. N.** Raspoznaniye opasnih situatsiy sistemami upravleniya dvizheniyem sudov. *Transport: nauka, tekhnika, upravleniye*. 2011. N. 8. P. 42—45.

11. **Yegorov I. B., Loginovskiy V. A.** Kontseptsiya zon navigatsionnoy bezopasnosti i yeyo primeneniye v sudovozhdenii. *Ekspluatatsiya morskogo transporta*. 2012. N. 3. P. 13—17.

12. **Konoplev M. A.** Primeneniye apparata nechetkoy logiki dlya opredeleniya urovnya opasnosti stolknoveniya. *Ekspluatatsiya morskogo transporta*. 2009. N. 2. P. 34—39.

13. **Nauk D., Klawonn F., Kruse R.** Foundations of Neuro-Fuzzy Systems. John Wiley & Sons. 1997. 305 p.

14. **Shtovba S. D.** Proyektirovaniye nechetkih system sredstvami MatLab. М.: Goryachaya liniya telekom, 2007. 288 p.

В. В. Чеботарев, ведущий программист, e-mail: VitalyChVV@mail.ru,
ОАО РЖД

Логическая нейронная сеть в основе контроля данных о перевозке грузов на железнодорожном транспорте

Предлагается практическая реализация информационной технологии, основанной на применении логической нейронной сети, в системе принятия решений по контролю поступления данных в центр управления перевозкой грузов из дорожных информационных центров сети ОАО РЖД.

Ключевые слова: логическая нейронная сеть, система принятия решений, факторное пространство событий, ситуация

Введение

Для сведения дорожной информации в единую базу данных (БД) и представления ее пользователям сети ОАО "РЖД" необходима ежедневная концентрация данных на сетевом сервере. Для этого организован регулярный сбор данных дорожного уровня. Из-за возможных сбоев на различных этапах работы системы требуется частое вмешательство человека для поддержания ее работоспособности. Однако данный процесс должен проходить в автоматическом режиме. Человек должен выполнять лишь функцию внешнего наблюдателя, следить за корректностью работы системы.

В работе [1] описывается практическая реализация системы принятия решений по контролю над процессом поступления данных в центр управления перевозкой грузов из дорожных информационных центров. В основе модели лежат методы ситуационного управления [2].

Однако теория логических нейронных сетей [3—7], использующая математическую логику событий, предлагает менее трудоемкий способ построения подобных систем принятия решений. Этот способ предлагает простую формализацию модели, позволяет значительно увеличить число решений (уточняемых в процессе эксплуатации) по всем возможным ситуациям, обрабатывать нечеткие данные, вводить и изменять веса влияния каждого фактора на результат, легко проводить расширение, модификацию и даже переориентацию системы.

Формализация системы принятия решений

Исходными данными (системой), по которым будет надстраиваться система принятия решений по контролю над поступлением данных в центр перевозки (ЦП) грузов, является комплекс программ, реализующих технологию сетевой концентрации данных (СКД).

СКД является двухуровневой. Первый уровень — уровень дорожных центров — состоит из 16 серверов,

территориально расположенных в вычислительном центре (ВЦ) по всей сети ОАО "РЖД". Эти серверы являются источниками исходных данных, которые по мере готовности должны поступить в ЦП для обработки. Следовательно, вторым уровнем СКД является сетевой уровень. Именно сюда поступают данные из дорожных центров, обрабатываются, накапливаются и форматируются для представления пользователям.

Программное обеспечение, реализующее СКД, разнородно и работает в различных аппаратно-программных средах. Это нередко порождает возникновение сбоев, препятствующих достижению цели СКД в установленный срок. Поэтому потребовалось создание программного комплекса, выполняющего автоматическое определение состояния процессов и немедленное принятие решения по устранению сбойных ситуаций. Таким программным комплексом является разработанная и обсуждаемая ниже система принятия решений (СПР).

Логическое описание системы принятия решений

Рассматриваемая СПР в факторном пространстве событий учитывает 12 факторов, каждый фактор принимает ряд значений. Каждая *ситуация* представляется вектором значений следующих факторов:

{(1) общее время сбора информации — отчетные сутки; (2) время сбора информации по дороге *XX*; (3) состояние отчета; (4) готовность БД на дороге *XX*; (5) наработка данных на дороге *XX*; (6) прием данных сервером центра перевозок; (7) запись в файл принятых данных по дороге *XX* на сервере ЦП; (8) запись данных по дороге *XX* в БД на сервере ЦП; (9) наработка готовых форм (ГФ) в формате HTML; (10) общая наработка готовых форм в формате EXCEL; (11) наработка готовых форм в формате EXCEL по дороге *XX*; (12) связь с сервером дороги *XX*}.

Ниже указаны значения факторов, используемые системой принятия решений.

(1) ОБЩЕЕ ВРЕМЯ СБОРА ИНФОРМАЦИИ — ОТЧЕТНЫЕ СУТКИ

(общий фактор для всех дорожных серверов) —

$\langle X_1^1, X_2^1 \rangle$, где

X_1^1 — отчетные сутки **ЗАВЕРШИЛИСЬ**;

X_2^1 — отчетные сутки еще **НЕ ОКОНЧИЛИСЬ**;

(2) ВРЕМЯ СБОРА ИНФОРМАЦИИ

ПО ДОРОГЕ ХХ — $\langle X_1^2, X_2^2 \rangle$, где

X_1^2 — время сбора информации по дороге ХХ **еще не** наступило;

X_2^2 — время сбора информации по дороге ХХ **наступило**;

(3) СОСТОЯНИЕ ОТЧЕТА — $\langle X_1^3, X_2^3, X_3^3, X_4^3 \rangle$, где

X_1^3 — отчет **заполнен** (полностью);

X_2^3 — отчет **не заполнен** по дороге ХХ;

X_3^3 — отчет **частично** заполнен по дороге ХХ;

X_4^3 — отчет **недоступен** (общий фактор для всех дорожных серверов);

(4) ГОТОВНОСТЬ БД НА ДОРОГЕ ХХ —

$\langle X_1^4, X_2^4, X_3^4, X_4^4 \rangle$, где

X_1^4 — БД на дороге ХХ **еще не** готова (**сбой**);

X_2^4 — БД на дороге ХХ готова **не полностью**;

X_3^4 — БД на дороге ХХ **еще нарабатывается (за-**

держка);

X_4^4 — БД на дороге ХХ **готова**;

(5) НАРАБОТКА ДАННЫХ НА ДОРОГЕ ХХ —

$\langle X_1^5, X_2^5, X_3^5 \rangle$, где

X_1^5 — наработка информации на дороге ХХ **еще не** окончена;

X_2^5 — **сбой** при наработке информации на дороге ХХ;

X_3^5 — **сбой** при передаче информации с дороги ХХ;

(6) ПРИЕМ ДАННЫХ СЕРВЕРОМ ЦП —

$\langle X_1^6, X_2^6 \rangle$, где

X_1^6 — данные **не** приняты ИЛИ приняты **не полностью**;

X_2^6 — приняты **не соответствующие** данные;

(7) ЗАПИСЬ В ФАЙЛ ПРИНЯТЫХ ДАННЫХ ПО ДОРОГЕ ХХ НА СЕРВЕРЕ ЦП — $\langle X_1^7, X_2^7 \rangle$, где

X_1^7 — данные по дороге ХХ в файл **не сохранены** или сохранены **не полностью**;

X_2^7 — данные по дороге ХХ в файл **сохранены**;

(8) ЗАПИСЬ ДАННЫХ В БД ПО ДОРОГЕ ХХ — $\langle X_1^8, X_2^8 \rangle$, где

X_1^8 — данные по дороге ХХ в БД **не сохранены** или сохранены **не полностью**;

X_2^8 — данные по дороге ХХ в БД **сохранены**;

(9) НАРАБОТКА ГФ HTML-формата (общий фактор) — $\langle X_1^9, X_2^9, X_3^9 \rangle$, где

X_1^9 — наработка ГФ в формате HTML **еще не** окончена, или **время ожидания** наработки не закончилось;

X_2^9 — **сбой** при наработке ГФ в формате HTML;

X_3^9 — наработка ГФ в формате HTML **окончена**;

(10) ОБЩАЯ НАРАБОТКА ГФ В ФОРМАТЕ

EXCEL (общий фактор) — $\langle X_1^{10}, X_2^{10} \rangle$, где

X_1^{10} — наработка ГФ в формате EXCEL по всем дорогам **еще не** окончена;

X_2^{10} — **сбой** при наработке ГФ в формате EXCEL по всем дорогам;

(11) НАРАБОТКА ГФ В ФОРМАТЕ EXCEL ПО ДОРОГЕ ХХ — $\langle X_1^{11}, X_2^{11} \rangle$, где

X_1^{11} — наработка ГФ в формате EXCEL по выбранным дорогам **еще не** окончена;

X_2^{11} — **сбой** при наработке ГФ в формате EXCEL по выбранным дорогам;

(12) СВЯЗЬ С СЕРВЕРОМ ДОРОГИ ХХ (PING) — $\langle X_1^{12}, X_2^{12} \rangle$, где

X_1^{12} — **нет связи** с сервером дороги ХХ (ping не проходит);

X_2^{12} — **есть связь** с сервером дороги ХХ (ping проходит нормально);

На основе данных значений факторов число N всех возможных ситуаций, без рассмотрения их совместимости, находится следующим образом:

$$N = 2 \times 2 \times 4 \times 4 \times 3 \times 2 \times 2 \times 3 \times 2 \times 2 \times 2 = 36\ 864.$$

Таким образом, каждая ситуация s , требующая решения $R(s)$, характеризуется 12 значениями факторов. Значения каждого фактора образуют исчерпывающее множество событий. Каждая ситуация должна обладать единственным основным решением, а некоторые еще и дополнительными решениями. Иными словами, система принятия решений должна быть полной и непротиворечивой.

Практически нескольким ситуациям может соответствовать одно решение [1], и таких ситуаций большинство. Так что в действительности число решений R значительно меньше N .

Приведем значения решений R , которые СПР должна выдавать на выходе нейронной сети.

(1) ОЖИДАНИЕ ОКОНЧАНИЯ ОТЧЕТНЫХ СУТОК — $\langle R^1 \rangle$, где

R^1 — остановка работы СПД до 18:01 (времени готовности данных за отчетные сутки (ОС) по сети);

(2) ВРЕМЕННАЯ ПАУЗА — $\langle R^2 \rangle$, где

R^2 — остановка работы СПД на заданное время (~5400 с);

(3) ДИРЕКТИВА НА НАРАБОТКУ ДАННЫХ — $\langle R^3, R^4 \rangle$, где

R^3 — директива дорожному центру ХХ на наработку данных;

R^4 — директива дорожному центру ХХ на "принудительную" наработку;

(4) ДИРЕКТИВА НА ПЕРЕДАЧУ ДАННЫХ — $\langle R^5, R^6 \rangle$, где

R^5 — директива дорожному центру ХХ на передачу данных;

R^6 — директива дорожному центру ХХ на "принудительную" передачу;

(5) ПОДЪЕМ ДАННЫХ ИЗ ФАЙЛА НА СЕРВЕРЕ ЦП — $\langle R^7 \rangle$, где

R^7 — директива на загрузку данных дорожного центра ХХ из файла на сервере ЦП;

(6) ПОВТОРНАЯ НАРАБОТКА ГФ В ФОРМАТЕ HTML — $\langle R^8 \rangle$, где

R^8 — директива на наработку ГФ HTML-формата;

(7) ПОВТОРНАЯ НАРАБОТКА ГФ В ФОРМАТЕ EXCEL — $\langle R^9, R^{10} \rangle$, где

R^9 — директива на наработку ГФ по **выбранным** дорогам;

R^{10} — директива на наработку ГФ по **всем** дорогам;

(8) ОПОВЕЩЕНИЕ АДМИНИСТРАТОРОВ — $\langle R^{11}, R^{12}, R^{13}, R^{14}, R^{15} \rangle$, где

R^{11} — сообщение администраторам дорожных центров и ЦП "данные за хх.хх.хх ОС наработаны";

R^{12} — сообщение администратору ЦП "WEB-отчеты недоступны";

R^{13} — сообщение администратору дорожного центра ХХ "проблема с БД";

R^{14} — сообщение администратору дорожного центра ХХ "ОС не соответствуют";

R^{15} — сообщение администратору дорожного центра ХХ "нет связи с сервером";

В рамках 15 приведенных решений представим логическое описание исследуемой СПР:

$$\begin{aligned}
 & X_1^1 \vee (X_2^1 \wedge X_1^3) \rightarrow R^1; \\
 & X_2^1 \wedge (X_1^2 \vee (X_1^4 \vee X_2^4 \vee X_3^4) \vee X_1^5 \vee X_1^9 \vee X_1^{10} \vee \\
 & \vee X_1^{11} \vee X_1^{12}) \rightarrow R^2; \\
 & X_2^1 \wedge X_2^2 \wedge X_2^3 \wedge X_4^4 \wedge X_3^9 \wedge X_2^{12} \rightarrow R^3; \\
 & X_2^1 \wedge X_2^2 \wedge X_3^3 \wedge X_4^4 \wedge X_3^9 \wedge X_2^{12} \rightarrow R^4; \\
 & X_2^1 \wedge X_2^2 \wedge X_2^3 \wedge (X_3^5 \vee X_1^6 \vee X_1^7) \wedge X_2^{12} \rightarrow R^5; \\
 & X_2^1 \wedge X_2^2 \wedge X_3^3 \wedge (X_3^5 \vee X_1^6 \vee X_1^7) \wedge X_2^{12} \rightarrow R^6; \\
 & X_2^1 \wedge X_2^2 \wedge (X_2^3 \vee X_3^3) \wedge X_2^7 \wedge X_1^8 \rightarrow R^7; \\
 & X_2^1 \wedge X_2^2 \wedge (X_2^3 \vee X_3^3) \wedge X_2^8 \wedge X_2^9 \rightarrow R^8; \\
 & X_2^1 \wedge X_2^2 \wedge (X_2^3 \vee X_3^3) \wedge X_2^8 \wedge X_2^{11} \rightarrow R^9; \\
 & X_2^1 \wedge X_2^2 \wedge (X_2^3 \vee X_3^3) \wedge X_2^8 \wedge X_2^{10} \rightarrow R^{10}; \\
 & X_1^1 \vee (X_2^1 \wedge X_1^3) \rightarrow R^{11}; \\
 & (X_1^1 \vee X_2^1) \wedge X_4^3 \rightarrow R^{12}; \\
 & X_2^1 \wedge X_2^2 \wedge (X_2^3 \vee X_3^3) \wedge (X_1^4 \vee X_2^4 \vee X_3^4) \rightarrow R^{13}; \\
 & X_2^1 \wedge X_2^2 \wedge X_2^6 \rightarrow R^{14}; \\
 & X_2^1 \wedge X_2^3 \wedge X_1^{12} \rightarrow R^{15}.
 \end{aligned}
 \tag{1}$$

В работах [3–7] доказаны преимущества и всегда существующие возможности построения однослойной нейронной сети, идентичной "электронной" схеме, реализующей логическое описание СПР. В такой схеме конъюнкторы и дизъюнкторы заменяются нейронами, реализующими единую функцию активации. Однако схема (1), будучи реализована аппаратно, содержит логические цепочки длиной более единицы. Ее следует преобразовать так, чтобы все логические цепочки были единичной длины.

Тогда в соответствии с известной технологией, представленной в тех же источниках, рекомендуется представить все логические функции в выражении (1) в виде дизъюнктивных нормальных форм, т. е. в виде дизъюнкции конъюнкций, а затем выполнить операцию "размножения" решений. При формировании дизъюнктивной нормальной формы приходится часто использовать свойства дистрибутивности операции \wedge относительно операции \vee и дистрибутивности операции \vee относительно операции \wedge .

Размножение решений заключается в том, что для каждой конъюнкции отдельно записывается то решение, что и для всей формы. Однако для сокращения записи можно с большой осторожностью использовать скобки так, чтобы при их раскрытии не возникало ситуаций, не соответствующих данному решению.

Тогда логическое описание СПР, соответствующее однослойной логической нейронной сети, имеет следующий вид:

$$\begin{aligned}
 & X_1^1 \rightarrow R^{11}; \\
 & X_2^2 \wedge X_1^3 \rightarrow R^{12}; \\
 & X_2^1 \wedge X_1^2 \rightarrow R^{21}; \\
 & X_2^1 \wedge X_1^5 \rightarrow R^{22}; \\
 & X_2^1 \wedge X_1^9 \rightarrow R^{23}; \\
 & X_2^1 \wedge X_1^{10} \rightarrow R^{24}; \\
 & X_2^1 \wedge X_1^{11} \rightarrow R^{25}; \\
 & X_2^1 \wedge X_1^{12} \rightarrow R^{26}; \\
 & X_2^1 \wedge X_1^4 \rightarrow R^{27}; \\
 & X_2^1 \wedge X_2^4 \rightarrow R^{28}; \\
 & X_2^1 \wedge X_3^4 \rightarrow R^{29}; \\
 & X_2^1 \wedge X_2^2 \wedge X_3^3 \wedge X_4^4 \wedge X_3^9 \wedge X_2^{12} \rightarrow R^3; \\
 & X_2^1 \wedge X_2^2 \wedge X_3^3 \wedge X_4^4 \wedge X_3^9 \wedge X_2^{12} \rightarrow R^4; \\
 & X_2^1 \wedge X_2^2 \wedge X_2^3 \wedge X_3^5 \wedge X_2^{12} \rightarrow R^{51}; \\
 & X_2^1 \wedge X_2^2 \wedge X_2^3 \wedge X_1^6 \wedge X_2^{12} \rightarrow R^{52}; \\
 & X_2^1 \wedge X_2^2 \wedge X_2^3 \wedge X_1^7 \wedge X_2^{12} \rightarrow R^{53}; \\
 & X_2^1 \wedge X_2^2 \wedge X_3^3 \wedge X_3^5 \wedge X_2^{12} \rightarrow R^{61}; \\
 & X_2^1 \wedge X_2^2 \wedge X_3^3 \wedge X_1^6 \wedge X_2^{12} \rightarrow R^{62}; \\
 & X_2^1 \wedge X_2^2 \wedge X_3^3 \wedge X_1^7 \wedge X_2^{12} \rightarrow R^{63}; \\
 & X_2^1 \wedge X_2^2 \wedge X_2^3 \wedge X_2^7 \wedge X_1^8 \rightarrow R^{71}; \\
 & X_2^1 \wedge X_2^2 \wedge X_3^3 \wedge X_2^7 \wedge X_1^8 \rightarrow R^{72}; \\
 & X_2^1 \wedge X_2^2 \wedge X_3^3 \wedge X_2^8 \wedge X_2^9 \rightarrow R^{81}; \\
 & X_2^1 \wedge X_2^2 \wedge X_2^3 \wedge X_2^8 \wedge X_2^9 \rightarrow R^{82}; \\
 & X_2^1 \wedge X_2^2 \wedge X_2^3 \wedge X_2^8 \wedge X_2^{11} \rightarrow R^{91}; \\
 & X_2^1 \wedge X_2^2 \wedge X_3^3 \wedge X_2^8 \wedge X_2^{11} \rightarrow R^{92}; \\
 & X_2^1 \wedge X_2^2 \wedge X_2^3 \wedge X_2^8 \wedge X_2^{10} \rightarrow R^{101}; \\
 & X_2^1 \wedge X_2^2 \wedge X_3^3 \wedge X_2^8 \wedge X_2^{10} \rightarrow R^{102}; \\
 & X_1^1 \rightarrow R^{111}; \\
 & X_2^1 \wedge X_1^3 \rightarrow R^{112}; \\
 & X_1^1 \wedge X_4^3 \rightarrow R^{121}; \\
 & X_2^1 \wedge X_4^3 \rightarrow R^{122}; \\
 & X_2^1 \wedge X_2^2 \wedge X_2^3 \wedge X_1^4 \rightarrow R^{131}; \\
 & X_2^1 \wedge X_2^2 \wedge X_2^3 \wedge X_2^4 \rightarrow R^{132}; \\
 & X_2^1 \wedge X_2^2 \wedge X_2^3 \wedge X_3^4 \rightarrow R^{133}; \\
 & X_2^1 \wedge X_2^2 \wedge X_3^3 \wedge X_1^4 \rightarrow R^{134}; \\
 & X_2^1 \wedge X_2^2 \wedge X_2^3 \wedge X_2^4 \rightarrow R^{135}; \\
 & X_2^1 \wedge X_2^2 \wedge X_3^3 \wedge X_3^4 \rightarrow R^{136}; \\
 & X_2^1 \wedge X_2^2 \wedge X_2^6 \rightarrow R^{14}; \\
 & X_2^1 \wedge X_2^3 \wedge X_1^{12} \rightarrow R^{15}.
 \end{aligned}$$

Так как каждое решение выдается в текстовой форме, то в текстах "размноженных" решений можно приводить их обоснование. Тогда формально одинаковые решения станут разными. На это и указывает дополнительный индекс.

Построение логической нейронной сети

По технологии построения логических нейронных сетей значением каждого фактора является *высказывание о состоянии данных разного уровня либо отдельных узлов системы*. Значения переменных на рецепторах формируются на основе *достоверности высказываний о состоянии данных или узлов системы*. Каждый фактор однозначно определен и имеет несколько значений, в совокупности образующих *исчерпывающее множество событий*. (Этот факт и позволяет формально получать дизъюнктивные нормальные формы, не содержащие отрицаний.)

Таким образом, логическая нейронная сеть (рис. 1, см. четвертую сторону обложки) может использовать на входе нечеткие данные в диапазоне [0, 1] (высказывание и значение его достоверности будем обозначать одинаково). В этом случае значение фактора, с которым осуществляется вход, преобразуется в возбуждение двух близких рецепторов, ограничивающих это значение (на основе деления единицы обратно пропорционально расстоянию до них). Связи, соответствующие решениям, на рисунке показаны. В рассматриваемой реализации веса всех связей равны единице. При накоплении опыта эксплуатации СПР веса связей могут быть уточнены. Число рецепторов определяется суммарным числом элементов всех исчерпывающих множеств событий и также может уточняться по мере накопления опыта эксплуатации СПР.

Единственный слой, являющийся выходным, образуется нейронами, максимальное возбуждение которых указывает на предлагаемое решение. Очевидно, количественный и содержательный набор решений также легко может развиваться.

Учет не всех комбинаций признаков приводит к необходимости тщательного подбора параметров функции активации. Решение должно приниматься не просто по максимальному возбуждению нейронов выходного слоя (оно существует всегда!), а по возбуждению, равному достаточно высокому порогу.

Воспользовавшись рекомендациями, приведенными в работах [4–7], выбираем функцию активации на основе счета значений следующих выражений:

$$\begin{aligned}
 f_i &= \frac{1}{k_i} \sum_j \omega_j V_j; \\
 f_{\text{вых}i} &= \begin{cases} f, & \text{если } f_i \geq h; \\ 0 & \text{в противном случае.} \end{cases} \quad (2)
 \end{aligned}$$

Здесь V_j — значение возбуждения на j -м входе нейрона выходного слоя, поступившего от связанного с ним рецептора; ω_j — вес входа (связи, здесь

большей достоверности, к оперативной модификации сети в связи с модификацией структуры системы, позволяет оперативно вводить и уточнять веса факторов, вводить новые факторы и их значения и т. д.

Исключительно важной является возможность работы с нечеткими данными.

Алгоритмы обработки логических нейронных сетей на основе примитивной пороговой функции активации обладают малой сложностью и не предъявляют высоких требований к производительности вычислительных средств.

Принципы ассоциативных вычислений обеспечивают весьма простое задание исходных возбуждений рецепторов на основе приближенных оценок ситуации лицом, принимающим решение, и допускают оперативное исправление неправильных действий.

Список литературы

1. **Зябиров Х. Ш., Кутыркин А. В., Левин Б. А.** Компьютерная поддержка принятия решений в транспортных системах. М.: ИКЦ "Академкнига", 2010. 520 с.
2. **Поспелов Д. А.** Ситуационное управление. Теория и практика. М.: Наука, 1986. 288 с.
3. **Барский А. Б.** Математическая логика событий и логические нейронные сети // Информационные технологии. Приложение. 2007. № 2. 32 с.
4. **Барский А. Б.** Нейросетевые технологии на транспорте // Мир Транспорта. 2011. № 2. С. 4—11; № 3. С. 14—19.
5. **Барский А. Б.** Логические нейронные сети: учеб. пособие. М.: ИНТУИТ; БИНОМ. Лаборатория знаний, 2007. 352 с.
6. **Барский А. Б.** Нейронные сети логического вывода. Курс лекций. Saarbrücken: LAP LAMBERT Academic Publishing, 2011. 400 с.
7. **Барский А. Б., Дмитриев А. А., Барская О. А.** Медицинские информационно-справочные системы на логических нейронных сетях // Информационные технологии. Приложение. 2010. № 1. 32 с.

V. V. Chebotarev, Leaging Programmer, e-mail: VitalyChVV@mail.ru, Russian Railways

Logical Neural Network in the Basis of Monitoring Data on the Carriage of Goods by Rail

Initial data for systems of practical implementation of information technologies application logic neural network is the automated system of daily concentrations of data on the carriage of goods by rail in a single database with a view to its external users. The complexity and diverse nature of the nodes in the system leads to malfunction, and require human intervention. Instead, the system provides for automatic mode, where the individual performs the function of an external observer, monitors the correctness of the system as a whole.

The article offers practical implementation of information technologies of application of logic neural networks in the system of decision making for the control of receipt of data in the center for the transportation of goods from road information centres network of Russian Railways. The model are methods of situational management.

Keywords: *logical neural network, decision making system, factor space of events, situation, loading, hardly points of unloading*

References

1. **Zjabirov H. Sh., Kutyrkin A. V., Ljovin B. A.** Komp'yuternaja podderzhka prinjatija reshenij v transportnyh sistemah. M.: IKC "Akademkniga", 2010. 520 p.
2. **Pospelov D. A.** Situacionnoe upravlenie. Teorija i praktika. M.: Nauka, 1986. 288 p.
3. **Barskij A. B.** Matematicheskaja logika sobytij i logicheskie nejronnye seti. *Informacionnye tehnologii. Prilozhenie.* 2007. N. 2. 32 p.
4. **Barskij A. B.** Nejosetevye tehnologii na transporte. *Mir Transporta.* 2011. N. 2. P. 4—11; N. 3. P. 14—19.
5. **Barskij A. B.** Logicheskie nejronnye seti: Uchebnoe posobie. M.: INTUIT; BINOM. Laboratorija znaniy, 2007. 352 p.
6. **Barskij A. B.** Nejrionnye seti logicheskogo vyvoda. Kurs lekcij. Saarbrjucken, LAP LAMBERT Academic Publishing, 2011. 400 p.
7. **Barskij A. B., Dmitriev A. A., Barskaja O. A.** Medicinskie informacionno-spravochnye sistemy na logicheskikh nejronnyh setjah. *Informacionnye tehnologii. Prilozhenie.* 2010. N. 1. 32 p.

Адрес редакции:

107076, Москва, Стромьинский пер., 4

Телефон редакции журнала (499) 269-5510

E-mail: it@novtex.ru

Дизайнер Т.Н. Погорелова. Технический редактор Е.В. Конова.

Корректор Е.В. Комиссарова.

Сдано в набор 08.07.2014. Подписано в печать 21.08.2014. Формат 60×88 1/8. Бумага офсетная.

Усл. печ. л. 8,86. Заказ IT914. Цена договорная.

Журнал зарегистрирован в Министерстве Российской Федерации по делам печати, телерадиовещания и средств массовых коммуникаций.

Свидетельство о регистрации ПИ № 77-15565 от 02 июня 2003 г.

Оригинал-макет ООО "Авансед солюшнз". Отпечатано в ООО "Авансед солюшнз".

119071, г. Москва, Ленинский пр-т, д. 19, стр. 1.



2014
CEE-SEC'R

Десятая юбилейная конференция
«Разработка ПО / CEE-SEC'R»

Выступления лидеров мировой
и российской индустрии разработки ПО

- Майк Милинкович - Исполнительный директор, Eclipse Foundation
- Дэвид Гарлан - Профессор, Университет Карнеги-Меллон
- Дино Эспозито - Технический евангелист, JetBrains
- Александр Тормасов - Профессор, МФТИ

И еще более 100 докладчиков...

Без малого 1000 участников из 250 компаний. Дискуссии, мастер-классы, блиц-доклады, общение со спикерами, Beer Party, Хакатон, конкурсы и многое другое...

www.secr.ru

Спонсоры

