

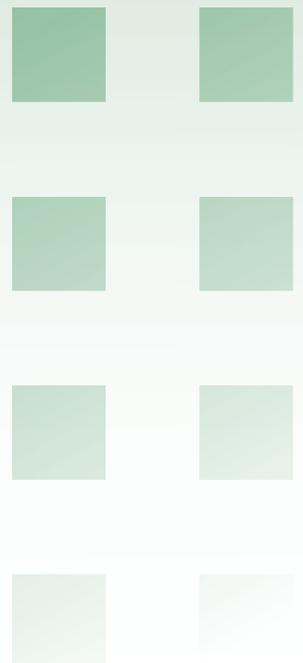
PSI 19

12-я Международная Ершовская
конференция по информатике

Труды семинара «Наукоемкое программное обеспечение»

2–3 июля 2019

Россия, Новосибирск



ИНСТИТУТ СИСТЕМ ИНФОРМАТИКИ ИМ. А. П. ЕРШОВА СО РАН
МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ
НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ПСИ'19

12-я МЕЖДУНАРОДНАЯ ЕРШОВСКАЯ КОНФЕРЕНЦИЯ
ПО ИНФОРМАТИКЕ

НАУКОЕМКОЕ ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ

Труды семинара

2–3 июля 2019 г.

Новосибирск
2019

УДК 004.4
ББК В185.123я431
Н 340

Программный комитет:

Председатель – д-р физ.-мат. наук *А. Г. Марчук*

Сопредседатель – канд. техн. наук *Ю. А. Загоруйко*

Секретарь программного комитета – канд. физ.-мат. наук *Т. В. Батура*

д-р физ.-мат. наук *А. И. Аветисян*, канд. физ.-мат. наук *И. Р. Агамирзян*,
канд. физ.-мат. наук *З. В. Апанович*, канд. физ.-мат. наук *М. А. Бульонков*,
канд. техн. наук *Ю. М. Зыбарев*, д-р техн. наук *В. Е. Зюбин*,
д-р техн. наук *А. А. Колташев*, д-р техн. наук *М. Г. Курносов*,
д-р физ.-мат. наук *М. М. Лаврентьев*, канд. физ.-мат. наук *Д. С. Мигинский*,
канд. физ.-мат. наук *Ф. А. Мурзин*, канд. физ.-мат. наук *А. Ю. Пальянов*,
д-р техн. наук *О. И. Потатуркин*, д-р физ.-мат. наук *А. Н. Терехов*,
И. А. Травина, д-р техн. наук *В. К. Трофимов*,
д-р техн. наук *А. Н. Фионов*, канд. физ.-мат. наук *Д. Н. Штокало*

Н 340 Научное программное обеспечение : труды семинара 12-й Междунар. Ершовской конф. по информатике (ПСИ'19). 2–3 июля 2019 г. / Ин-т систем информатики им. А. П. Ершова СО РАН, Новосиб. гос. ун-т. — Новосибирск : ИПЦ НГУ, 2019. — 152 с.

ISBN 978-5-4437-0909-3

В сборнике содержатся материалы, представленные на рабочем семинаре «Научное программное обеспечение», который проходил в рамках 12-й Международной Ершовской конференции по информатике (ПСИ'19).

Семинар организован Институтом систем информатики им. А.П. Ершова СО РАН при поддержке Новосибирского государственного университета.

УДК 004.4
ББК В185.123я431

ISBN 978-5-4437-0909-3 © Институт систем информатики им. А. П. Ершова СО РАН, 2019
© Новосибирский государственный университет, 2019

A. P. ERSHOV INSTITUTE OF INFORMATICS SYSTEMS SB RAS
MINISTRY OF SCIENCE AND HIGHTER EDUCATION OF THE RUSSIAN
FEDERATION
NOVOSIBIRSK STATE UNIVERSITY

PSI'19

A.P. ERSHOV INFORMATICS CONFERENCE
(the PSI Conference Series, 12th edition)

SCIENCE INTENSIVE SOFTWARE

Workshop proceedings

July 2–3, 2019

Novosibirsk
2019

УДК 004.4
ББК В185.123я431
Н 340

Program Committee:
Chair – *Alexander Marchuk*
Co-chair – *Yury Zagorulko*
Secretary – *Tatiana Batura*

Igor Agamirzyan
Zinaida Apanovich
Arutyun Avetisyan
Mikhail Bulyonkov
Andrey Fionov
Andrey Koltashev
Mikhail Kurnosov
Mikhail Lavrentiev
Denis Miginsky
Feodor Murzin
Andrey Palyanov
Oleg Potaturkin
Dmitry Shtokalo
Andrey Terekhov
Irina Travina
Viktor Trofimov
Yuri Zybarev
Vladimir Zyubin

Н 340 Science Intensive Software : Workshop proceedings of the 12th A. P. Ershov Informatics Conference (PSI'19). July 2–3, 2019 / A. P. Ershov Institute of Informatics Systems SB RAS, Novosibirsk State University. — Novosibirsk : IPC NSU, 2019. — 152 p.

ISBN 978-5-4437-0909-3

This volume comprises the papers selected for presentation at the Workshop on Science Intensive Software of the A.P. Ershov Informatics Conference PSI'19 (the PSI Conference Series, 12th edition)

The workshop is organized by the A.P. Ershov Institute of Informatics Systems SB RAS and Novosibirsk State University.

УДК 004.4
ББК В185.123я431

ISBN 978-5-4437-0909-3

© A. P. Ershov Institute of Informatics Systems SB RAS, 2019
© Novosibirsk State University, 2019

Содержание

1. <i>G. B. Abdikerimova, A. L. Bychkov, W. Xinyu, F. A. Murzin, N. E. Russkikh, E. I. Ryabchikova, S. S. Khayrulin.</i> Algorithms and software tools for analysis of textural images.....	6
2. <i>S. S. Krainikovskiy, M. Ye. Melnikov, R. S. Samarev.</i> Predicting personality from image preferences: how machine learning can help us?	14
3. <i>Е. П. Абрамов, У. Т. Махажанова, Ф. А. Мурзин.</i> Принятие решений в области кредитования на основе размытой логики Заде	20
4. <i>А. И. Адамович, А. В. Климов.</i> Принципы построения системы детерминированного параллельного программирования	26
5. <i>А. Ж. Ахметова, Л. Л. Ла, Ф. А. Мурзин.</i> Теория социального влияния Латане и ее применение для анализа окружения пользователя в социальной сети.....	34
6. <i>Д. Е. Бабурин.</i> Как устроены современные высокопроизводительные биржевые инфраструктуры	40
7. <i>Т. В. Батура, А. М. Бакиева.</i> Система автореферирования научно-технических текстов	47
8. <i>Т. В. Батура, Л. В. Ефимова, А. С. Еримбетова, А. Б. Касекеева, Ф. А. Мурзин.</i> Анализ временных и пространственных понятий, встречающихся в текстах на естественном языке. 53	53
9. <i>А. В. Бочарников, Е. В. Игнатьева, О. В. Вишневецкий.</i> Использование графических ускорителей для выявления функциональных сигналов в регуляторных районах генов прокариот	59
10. <i>М. А. Бульонков, Т. В. Нестеренко.</i> Система поддержки прогнозирования развития опорной транспортной сети России.....	68
11. <i>Л. А. Голубева, В. С. Гориунов, В. П. Ильин.</i> Модель управления вычислительным комплексом, реализованным на основе концепции БСМ	76
12. <i>Л. В. Городня.</i> О классификации парадигм и языков программирования.....	83
13. <i>Т. Н. Есикова, С. В. Вахрушева.</i> Разработка МАС для имитации процесса реализации проекта трансконтинентальной магистрали через Берингов пролив (моделирование информационного пространства).....	90
14. <i>Г. Б. Загорулько, Л. В. Массель.</i> Разработка интеллектуальной СППР по угрозам энергетической безопасности	97
15. <i>М. А. Кантуреева.</i> Применение клеточных автоматов для моделирования движения группы людей.....	104
16. <i>М. М. Лаврентьев, Л. В. Городня, М. А. Держо, Д. С. Мигинский.</i> Вопрос карьерных перспектив в области ИТ.....	111
17. <i>М. В. Платонова.</i> Графическое моделирование на базе Eclipse в системах управления.....	121
18. <i>К. Е. Сердюков, Т. В. Авдеенко.</i> Применение генетического алгоритма для генерации входных данных при тестировании программного кода	130
19. <i>Е. А. Сидорова.</i> Инструментарий для исследования лексических характеристик текста.....	138
20. <i>В. И. Шелехов.</i> Разработка сертифицированного компилятора предикатных программ.....	145

Algorithms and software tools for analysis of textural images**G. B. Abdikerimova**

Senior lecturer, L. N. Gumilyov Eurasian National University,
010000, Astana, Kazakhstan, Satbayev str., 2, e-mail: gulzira1981@mail.ru

A. L. Bychkov

Ph.D., Senior researcher, Institute of Solid State Chemistry and Mechanochemistry SB RAS,
630128, Novosibirsk, Russia, Kutateladze str., 18, e-mail: bychkov.a.l@gmail.com

W. Xinyu

Ph.D. student, Heilongjiang University,
150000, Harbin, China, Xuefu Rd, 74, e-mail: xinyuwei2016@163.com

F. A. Murzin

Ph.D., Deputy director, A. P. Ershov Institute of Informatics Systems SB RAS,
630090, Novosibirsk, Russia, Acad. Lavrentjev pr., 6, e-mail: murzin@iis.nsk.su

N. E. Russkikh

Ph.D. student, A. P. Ershov Institute of Informatics Systems SB RAS,
630090, Novosibirsk, Russia, Acad. Lavrentjev pr., 6, e-mail: russkikh.nikolay@gmail.com

E. I. Ryabchikova

Professor, Chief researcher, Institute of Molecular Biology and Fundamental Medicine,
630090, Novosibirsk, Russia, Acad. Lavrentjev pr., 8, e-mail: lenryab@niboch.nsc.ru

S. S. Khayrulin

Junior researcher, A. P. Ershov Institute of Informatics Systems SB RAS,
630090, Novosibirsk, Russia, Acad. Lavrentjev pr., 6, e-mail: s.khayrulin@gmail.com

Abstract. The purpose of the conducted research is development and search of analysis algorithms of textural images. The software products, which allow us to analyze successfully textures in details, can be used in different fields of science and the industry. First of all, it is chemistry and materials science. It is possible to analyze materials of organic origin, cuts of metals and minerals, ceramics, etc. Another field of research, where we can effectively apply these methods, is the diagnosis of internal pathologies of human, including malignant, according to the images received by means of the thermal imager. We consider images obtained using the transmission electron microscopy. This research was carried out for the Institute of solid state chemistry and mechanochemistry of the Siberian Branch of the Russian Academy of Sciences. Investigations are supported by the Russian Foundation for Basic Research (grant № 18-08-01284).

Key words: textural features, image processing, R/S-analysis, k-means method, agglomerative method, microphotography analysis, electron microscopy

Introduction

The aim of the research is to create a scientific and technical foundation in the field of texture image processing. Another aim of the research is to develop and search algorithms for analysis of

images obtained from various sources, for example, with the help of modern electron microscopic methods. Besides, images of ultrastructure of plant cell walls, obtained by transfer electron microscopy of herbal raw materials, after various physico-chemical, mechanochemical treatment were analyzed. Nowadays, this information is popular among scientists from various fields – chemists, biologists, technologists – but it is still handled manually at a qualitative (rarely semi-quantitative) level. The transition to algorithms, which allow operating large volumes of data, would allow us to the mentioned fields of science to make a significant step forward and to improve existing and create new technological processes.

This work focuses on developing software tools for image analysis, such as textural analysis, textural feature segmentation, R/S analysis programs, etc. The source is the microphotography of vegetable raw materials, which is shredded in special mills. In principle, our interests are not limited only with microphotography analysis. For example, during processing aerospace pictures, researchers also deal with different textures. It becomes possible to determine the coniferous or deciduous forest it is, whether the fields are planted with grain or bean plants, etc. just by using texture features. It is also possible to define whether the woods are affected by pests or deserted territories. Another field of research, where these methods can be used effectively, it is a diagnosis of internal pathologies of a person, including malignant, with the help of images obtained with the help of a thermal imager.

The fundamental difference between the ideas of our project and the existing analogues is the correct application of mathematical methods and in their deeper study. For example, we know more than two hundred texture characteristics, but in scientific reviews usually only about fifty kinds are used. At the same time, in practice, only 3–4 features are usually used, for example, during processing space images. That means that original images are explored in the full way. We can say the same about the application of integral transformations. For example, the Haar transformation is used during investigating the strength of metals under loads in order to characterize the fissuring. The question about information that can be obtained on the basis of other transformations is almost not studied. The literature on wood chemistry states the usefulness of R/S analysis and fractal analysis for corresponding studies, but even with this method, information is fragmentary.

1. Methods of textural features analysis

Despite the ubiquity of textures in images, so far there is no unique formal approach to describing the texture and its strict definition. As a rule, methods for texture analysis are developed for each individual case.

In [1], a texture is understood as a “spatial organization of elements within part of the surface”. It also explains that this organization is due to a certain statistical distribution of the intensity of gray tones or tones of different colors. A site can be considered as a texture if the number of intensity changes or color changes observed on it is sufficiently large. In [2], the texture is called “a surface area organized in some way”. In [3], the texture is defined as a matrix or a fragment of the spatial properties of sections of images with homogeneous statistical characteristics.

Textures can be divided into several classes, as follows:

- by origin: artificial (for example, graphic patterns) and natural (for example, grass, forest, land);
- by the surface structure: structural, consisting of geometrically correct repeating elements, and stochastic, formed by a sequence of random elements;
- by the relative dimensions of texture elements: fine-grained and coarsegrained;
- by the form of texture elements: wavy, spotted, incorrect, linear, and so forth [1].

From the above definitions and characteristics, it follows that a texture is a piece of an image, one that has homogeneous statistical characteristics. This means that each texture of this class can be described by a characteristic property common to all textures of this class [1]. These properties are called textural features. Textural features play an important role in dividing an image into separate areas.

In this problem, gray (halftone) images are considered. Thus, the image is given in the form of a matrix whose values are the brightness values of pixels in the range from 0 to 255.

The standard approach for calculating textural features is: It's necessary to select a so-called scrolling window from odd-numbered side: 3, 5, 7 pixels. The feature is calculated inside the scrolling window. The size of the local fragment is a carrier of the texture property. The feature value is written to the new matrix of the same size as the original. In the new matrix, the value is written to the coordinate, which is equal to the center of the scrolling window coordinate. The elements of the new matrix are obtained in a certain interval [A, B]. Further, this interval is linearly mapped into a segment [0,255]. After this, it is possible to visualize the result of calculating the texture feature.

The experiments have shown that the standard approach in our case is not informative. It was therefore decided to use the non-standard approach. Specifically, the textural windows are calculated by large windows (including, by non-square ones), that the user can specify by selecting an area that may be of interest to him/her. That is about calculating numerical characteristics that pertain to vast areas that include different kinds of artifacts.

There were considered 18 of the most important textural features [4], all of which are implemented in the program. Some textural features are represented bellow.

Textural features

Features based on statistical characteristics	Features taking into account relative position
k-th initial moment $T_1^k = n^{-2} \sum_{i=1}^n \sum_{j=1}^n [f(i, j)]^k .$	Average $T_1 = \mu_i = \mu_j = \sum_{i=0}^{N-1} \left[i \sum_{j=0}^{N-1} P(i, j) \right] .$
Entropy $T_2 = - \sum_{g=0}^{N-1} F(g) \log_{10} F(g) .$	Energy $T_2 = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} [P(i, j)]^2 .$
Energy $T_3 = \sum_{k=0}^{N-1} [F(g)]^2 .$	Variation $T_3 = \sigma_i^2 = \sum_{i=0}^{N-1} \left[(i - \mu_2)^2 \sum_{j=0}^{N-1} P(i, j) \right]$
Variation $T_4 = - \sum_{g=0}^{N-1} (g - \mu)^2 F(g) .$	Homogeneity $T_4 = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} P(i, j) / (1 + i - j) .$

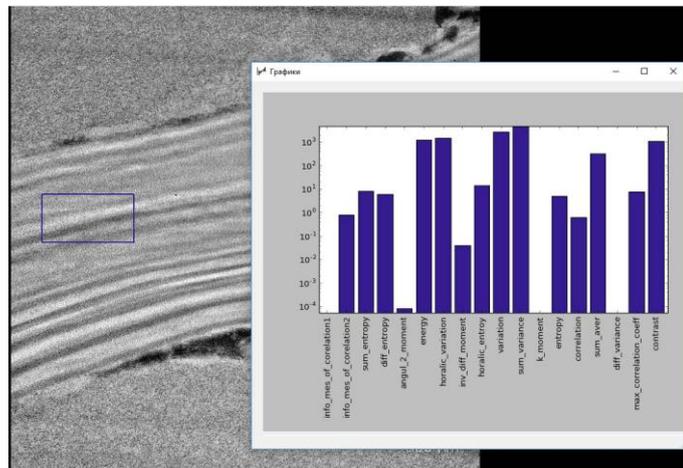


Fig. 1. Textural features calculation

2. Clustering algorithms

Image segmentation can be implemented in a local and global way. The local method evaluates whether there is a boundary between the areas by the features behave in the neighborhood of the image point. The global method presupposes the provisional clustering of the features space, and then establishes a match between the image pixel and the cluster to which its vector of features falls [5].

One of the heuristic methods of clustering is method based on a sequential agglomerative procedure. The advantage of these methods is the simplicity of computational procedure and algorithms.

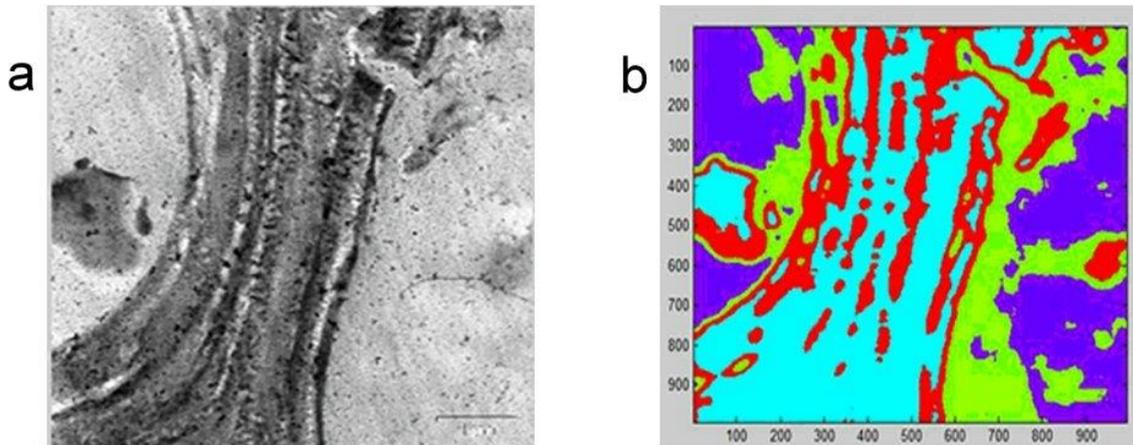


Fig. 2. Segmentation algorithm: *a* – original image; *b* – result of segmentation

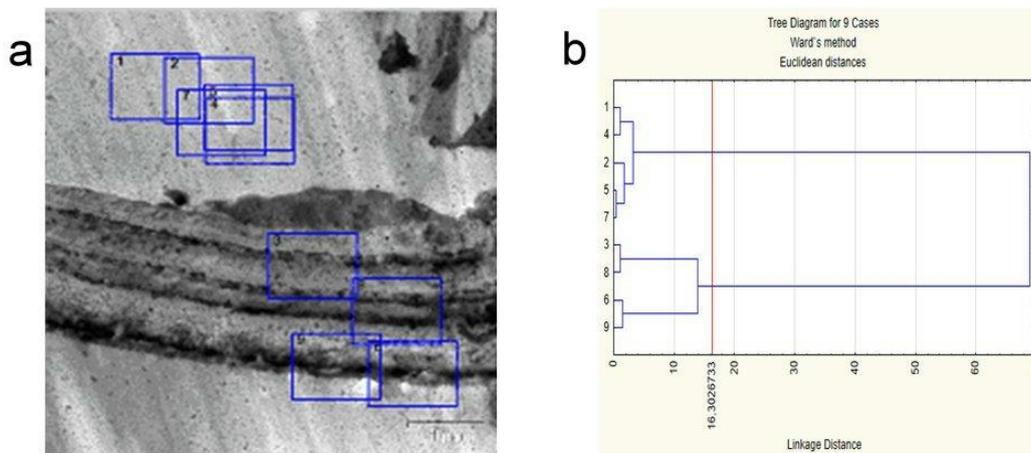


Fig. 3. Hierarchical clustering: *a* – selection of image fragments; *b* – result of clustering

3. R/S image analysis

The R/S-analysis method is a statistical method of analyzing time series or numerical sequences (primarily financial) to identify some important characteristics, such as the existence of non-periodic cycles, the “memory” of the process, degree of chaotic nature, etc. [6]. The method also applies to image analysis, because you can view the sequence of brightness functions along a line or curve. An important characteristic calculated by R/S-Analysis is the Hurst exponent, usually referred H , which characterizes the degree of chaoticness of the process [7]. Interestingly, the method is used in chemistry to analyze the microphotographs (for example, lignin) [8,9], and the method has shown some efficiencies.

The method is implemented in the MATLAB environment and an image analysis is performed. Images are divided into separate areas by software tools. If each area of the image belongs to the

scaling region, it will be possible to identify, from the point of view of estimating the fractal dimension of the neighboring sections, the smaller electron-dense (black) formations that are likely to be the residue intracellular of lipid or protein content (fractional research method). In this case, the fractal dimension of the microphotographs of the cell walls of the wheat straw will lie in the interval of the norm, and the fractal dimension of the sections with the lignified layers is below the norm boundary (Fig. 1). The image was divided into 1024 parts. For each part of the image, a fractal dimension calculation was performed.

The results of the fractionary method in the form of circles calculate the fractal dimension of the microphotographs. The blue circles correspond to the blue squares in the original image in Fig. 4. In this case, we have very big disordering the material structure. And white circles correspond to the background.

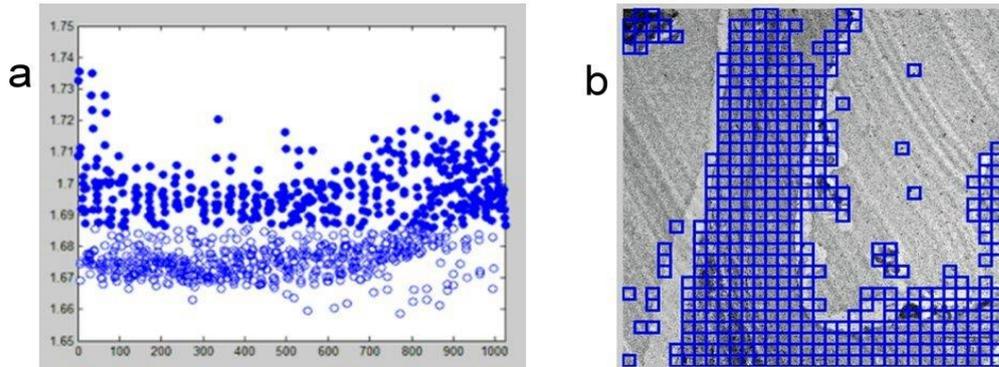


Fig. 4. R/S analysis of images: *a* – values of Hurst exponent; *b* – selected area

4. Orthogonal transformations

Spectral analysis is a powerful tool for analyzing signals and images [10], since it was noted long before that the spectrum is very sensitive to various changes in the structure of signals and images.

It is necessary to predefine previously the signal or image by frequency in order to perform a spectral analysis. Different sets of basic functions are used for this purpose [11]. The corresponding algorithms are called transformations: cosine, Hadamard, Haar, etc. We would like to notice that the transformations of Haar and Daubechies are the simplest wavelet transformations.

The program is implemented in Matlab environment and it allows us to calculate spectral transformations of six types: 1) cosine, 2) Hadamard of order $n = 2^k$, 3) Hadamard of order $n = p + 1$, $p \equiv 3(\text{mod}4)$ is a prime number, i.e. based on the Legendre symbol, 4) Haar, 5) slant, 6) Daubechies – 4.

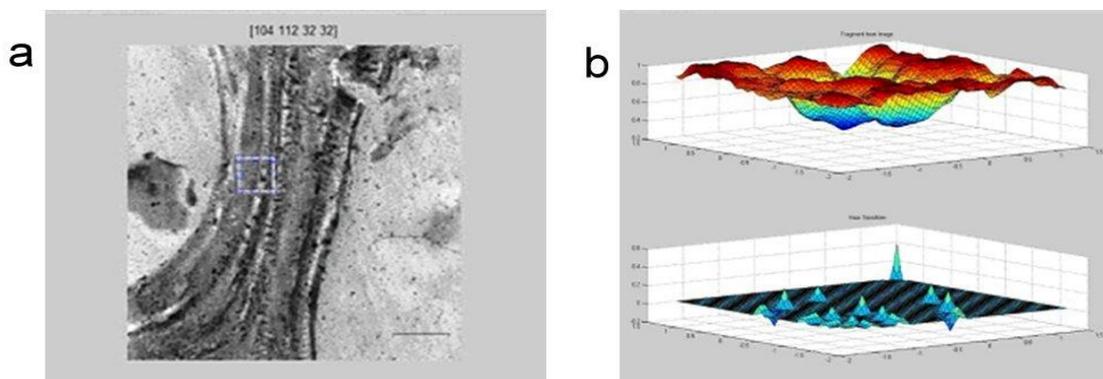


Fig. 5. Example of orthogonal transformation: *a* – selected window; *b* – brightness function in the window and the result of Haar transformation

It is possible to apply different non-standard approaches to the study of textures using orthogonal transformations. For example, the original image is split into non-intersecting square windows. As experiments show, it is expedient to take the size of a window large enough; such as 32×32 , 64×64 , etc. Then in each window we make an integral transformation. In this case, some spectral coefficients can be reset, for example, high-frequency, or in several definite parts of the spectrum. In the two-dimensional case, the frequency spectra represent two-dimensional matrices. You can arrange elements of the matrices in vectors. For example, the rows of the matrix can be placed sequentially one after the other. Actually, the positions in the vector in which the spectral coefficients were zeroed can be struck out of the resulting vector. Further, a clustering procedure can be carried out according to the available vectors.

Another approach is the wavelet-decomposition [12]. We can say that the wavelet-decomposition [8] is performed with the help of tree-connected dual-channel filter blocks. At the height of the tree, which is equal to d , it is usually assumed that the signal should have a length equal to 2^d . So, the image accordingly, should have the size $2^d \times 2^d$. If this is not done, then the missing counts are usually added, for example, by adding zero.

At each step of the transformation, the image is split into 4 matrices. One of them represents the image, which is similar to the original, but it is smaller in horizontally and vertically in 2 times. We can say that this is a "rough" version of the original image. In three other matrices, various brightness differences from the original image are "encoded". So, in such way the "detailing" of the information contained in the image occurs.

Values of elements may not fall into the interval $\{0, \dots, 255\}$. For each of the matrices, we find the values of the minimal and maximal elements and display in-line corresponding interval in the interval $\{0, \dots, 255\}$. This allows us to visualize the results in the form of gray images.

The simplest forms of wavelet transformations for images are the Haar and Daubechies -4 transformations. Below are the results (Fig. 6) of this kind of experiment for the Haar transformation.

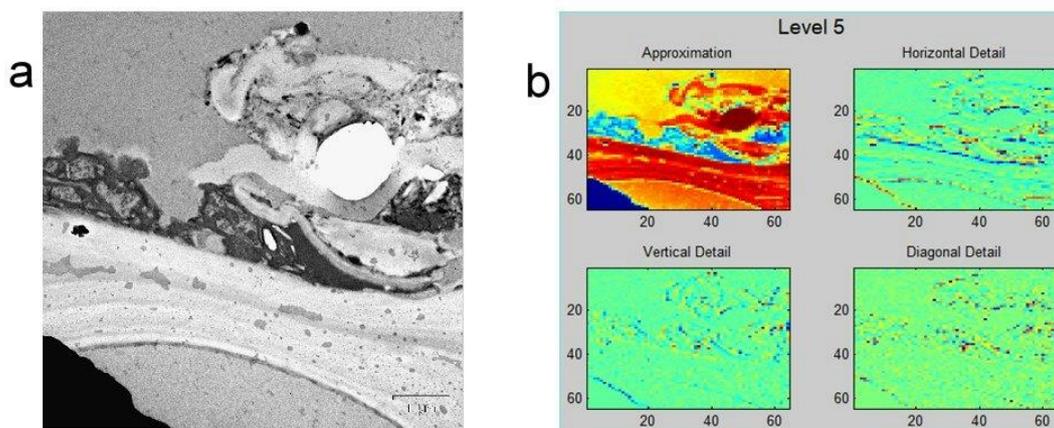


Fig. 6. Example of wavelet-decomposition: *a* – original image; *b* – 5-th level of wavelet – decomposition by Haar transformation

5. Conclusion

The work focuses on texture image studies, texture segmentation, and R/S analysis, orthogonal transformations, etc. The source is the microphotography of vegetable raw materials, which is shredded in special mills. The work was fulfilled for the Institute of Solid State Chemistry and Mechanochemistry of the Siberian branch of the Russian Academy of Sciences (ISSC SB RAS).

The work is about calculating different kinds of textural features and other image characteristics. Further, in principle, vectors of indicators can be "linked" to chemical reactivity. For example, a software system can be trained on examples based on neural-computer or other similar approaches. A large set of texture-study methods was analyzed, and the most informative ones were

chosen from the methods reviewed. Finally, a program was developed to calculate various textural features. The aim of the program is to carry out computer experiments and to accumulate numerical data (textural feature values) for cell walls images treated at different temperatures. Totally, 18 textural features are calculated. The results of calculation can be saved. As a result, the following software tools were developed.

1. The program to calculate various textural features.

2. Programs for cluster analysis have been implemented, in the standard version and hierarchical variant, and experiments have been carried out. The first method used 18 textural signs with the 9x9 pixels scrolling window, and in the second 5 textural features with the window of 400 x 300 pixels. In the second approach, after carrying out research on textural features, five texture features were selected, on which we conducted a segmentation.

3. Prototypes of programs implementing R/S-analysis have also been implemented. Primary experiments which showed interesting results that could be useful in practice, namely, for the support of the work in the field of chemistry. Note that similar utterances are found in the scientific literature on the study of lignin, but without particular details Segmentation based on the Hurst exponent in the microphotographs was performed. And with the help of these studies, we have come to the fact that the method has shown some efficiencies. The background and the cell wall clearly differ from each other in images.

4. The program is implemented in the Matlab environment, which allows for the implementation of spectral transformations of six types. Some their applications are considered. For example, the wavelet – decomposition of images.

In the future, machine learning algorithms, for example, based on a brain-computer approach, can be used to determine chemical reactivity. After the system in the data set has being trained, you can then make predictions about chemical reactivity.

References

1. *Andreev H. A., Bazarskii O. V., Hlauberman A. S., Kolesnikov A. I., Korzhik Yu. V., Khliavich Ya. L.* Analysis and synthesis of random spatial textures // Foreign radio electronics. Vol. 2, 1984. P. 3–33. (in Russian)
2. *Kharalik R. M.* Statistical and structural approaches to the description of textures // TIIEP, Vol. 67(5), 1979. P. 98–119. (in Russian)
3. *Potapov A. A.* New information technologies based on probabilistic textural and fractal features in radar detection of low contrast targets // Radiotechnology and electronics. Vol. 48(9), 2003. P. 1101–1119. (in Russian)
4. *Kolodnikova N. V.* Overview of textural features for image recognition tasks // Reports of the Tomsk State Univ. of Control Systems and Radioelectronics Automated Systems for Information Processing, Management and Design, 2004. P. 117–118. (in Russian)
5. *Sidorova V. S.* Hierarchical Cluster Algorithm for Remote Sensing Data of Earth // Pattern Recognition and Image Analysis. Vol. 22(2), 2012. P. 373–379. (in Russian)
6. *Feder E.* Fractals. M. : Mir, 1991. 254 p. (in Russian)
7. *Peters E.* Chaos and order in capital markets. M. : Mir, 2000. 333 p. (in Russian)
8. *Karmanov A. P.* Lignin: Structural Organization and Self-Organization // Theses of the Report of the III All-Russian Conference: Woodchemistry and Organic Synthesis, Syktyvkar: Komi Scientific Center of Ural Branch of the RAS, 1998. P. 66–67. (in Russian)
9. *Karmanov A. P., Matveyev D. V.* Problems of timber chemistry and wood chemistry // Syktyvkar: Institute of Chemistry of Komi Scientific Centre Ural Branch of the Russian academy of Sciences, 2001. P. 50–52. (in Russian)
10. *Rani R.* Performance analysis of different orthogonal transform for image processing application // Inter. J. of Applied Research. Vol. 1(12), 2015. P. 844–847.

11. *Shahdoosti H. R., Mirzapour F.* Spectral–spatial feature extraction using orthogonal linear discriminant analysis for classification of hyperspectral data // *European J. of Remote Sensing*. Vol. 50(1), 2017. P. 111–124.
12. *Vorobiev V. I., Gribunin V. G.* Theory and practice of wavelet transform. St. Petersburg : Military Communication Univ., 1999. 203 p. (in Russian)

Predicting personality from image preferences: how machine learning can help us?**S. S. Krainikovsky**

Researcher, programmer, Academ Gene,
630090, Novosibirsk, Russia, Acad. Lavrentiev pr., 6,
Dotin Inc, 194, Francisco Ln, Fremont, CA 94539, USA, e-mail: stas.k7@gmail.com

M. Ye. Melnikov

Psychologist, Academ Gene,
630090, Novosibirsk, Russia, Acad. Lavrentiev pr., 6,
Dotin Inc, 194, Francisco Ln, Fremont, CA 94539, USA, e-mail: mikhail-melnikov@mail.ru

R. S. Samarev

Technical director
Dotin Inc, 194, Francisco Ln, Fremont, CA 94539, USA, e-mail: info@dotin.us

Annotation. This paper is dedicated to machine learning in predicting personality traits from such indirect features as image preferences. Personality traits named Big 5, were measured. Several methods of machine learning were applied, including gradient boosting and neural networks, and the results, including prediction accuracies and correlations of real and predicted values were presented. The features, extracted from images were also identified, which give the best result in terms of personality prediction, and models that are most suitable for solving this problem were discussed.

Keywords: Personality, Big Five, machine learning, gradient boosting

Introduction

Personality information plays very important role and used widely in a lot of areas of our life and activities: applied psychology, HR, sociology, marketing and PR etc. But how we can measure it? The traditional approach is to calculate some characteristics from questionnaires, filled by people while passing some tests or surveys. However, in some cases this information cannot be gained by assessment with inventories. Besides, this method may have sufficient drawback, if respondent is intended to manipulate the results of the survey. For example, during job interview, people may try to give answers that they consider will lead them to desired result. So, the problem is how to estimate the psychometric data without answering questions, with only indirect features, obtained from respondent. In this article an approach, based on image preferences, is presented. Two types of initial data were tested: according to one approach people were asked to click the preferred images from a gallery, in the other – to upload favorite pictures from their computer or found by them in internet. Because the dependences between features, extracted from images, might be nonlinear, we used machine learning regression models, based on gradient boosting and neural network architecture. In conclusion a comparison of different models, features, and kinds of initial data is presented.

1. Models and experiments

Predicting personality from different features, related to person is a complicated task, because there is no single gold standard approach to implement it. Different approaches and variants of estimating the psychometric data exist. The second problem is an intra-subject variability of the psychometric data: one person may give different answers in the different moments of the time. Research [1] shows, that there are different kinds of errors, e.g. transient errors, long-term tendencies,

etc. All of these errors may be the case of intrapersonal data variation with 0,7–0,9 in terms of correlation. The third problem is a number of latent factors, which could not be taken into account, because corresponding questions are not present in the survey. Cultural, social, geographical and other factors may have a great influence on image and color preferences. All this factors may tend to decrease the accuracy of predicting models. In this research we try to estimate the power of machine learning methods in personality prediction.

Some attempts were made by [2], where such psychometric characteristics as Big Five, were predicted from Facebook “likes” using text information, related to liked object – image or post on Facebook. These results shows a tendency of increasing the correlation between values, predicted by the model and values, obtained through the questionnaire, while number of liked objects is increasing. The main result is that mean correlation become higher than 0,5 after analyzing 100 or more Facebook “likes” from one person. Our experiment was based on image preferences, and contains two main types of retrieving image preference information: collecting “likes” from pre-defined gallery images, and analyzing the images, that people uploaded from their computer or found through the internet. Also participants passed a questionnaire, so, finally, we have a dataset with personality information and image preferences. Here we will describe the experiment more precisely.

1.1. Experiment design

Participants underwent a paid online survey, where they need both to choose pictures from gallery and to pass psychological tests. On each page there were 15 images and participant was suggested to choose from one to five pictures per one page. A total 20 pages were exposed to participant, which resulted in total of 20 to 100 liked images. Image gallery was collected by qualified psychologist in order to allow for better prediction of target psychological traits and several tags were assigned to each picture by the same expert. Each tag described either subjects/objects in a picture, or their activity, or emotion, which was likely to be provoked by image, or the general description of a scene in the picture. Second test was to upload 10 favorite pictures from participant’s computer or found in the internet. After choosing images, participant passed the questionnaire, comprising 108 questions in total and containing Big Five test (NEO-PI), Gardner Intelligence Quiz, and Personal Globe Inventory (40 questions) [3]. Target characteristics were calculated with authorized keys for each test.

1.2. Feature extraction and data preparing

Data sample of 1149 participants, who passed the survey, were analyzed using several validation procedures. During the survey, participants were asked 5 validation questions, which proposed that participant will always answer it correctly. For example: “I am travelling trans-Atlantic five or more times in a month”, or attention questions, asking to press button with the particular number. Validation test was considered to be passed, if at least 4 answers were correct. 76,51 % of participants, who passed the validation test (868 persons) were included in the training dataset.

Several groups of features, extracted from images were analyzed. In the work [4] has been considered more image features but influenced to images perception. First group comprised image tags suggested by psychologist (participants did not see them). A total set of 105 tags was used to describe the participant’s choice. For each tag with index $i = 1 \dots N$ a frequency score S_i was calculated:

$$S_i = \frac{K_i}{\sum_{j=1}^N K_j} ,$$

where K_i is a number of times, where tag with index i arrears in user’s liked pictures set. So, these scores are like a “bag of words”, but formed from picture’s tags.

Second group was color histograms. Each of three color dimensions was down-sampled to 32 bins, and a total vector of 96 color bins was used as training vector.

Target labels, used for training, were obtained from questionnaire using psychological tests. There were 46 attributes, containing 5 from Big Five, 3 from buying scale, 8 from Multiple Intelligence Quiz and 40 from PGI (Personal globe inventory [3]).

Big 5 model characterizes personality as a point in five-dimensional space with such dimensions as extraversion, neuroticism, openness, agreeableness, and conscientiousness. The prominence of these traits determine person's behavior style. This model is well validated, and is the most often used one for personality prediction from internet behavior (likes, posts, tweets, Facebook status etc.). Buying scale measures maturity of three styles of buying behavior which is of great importance for marketing. Multiple intelligence model includes 8 intellectual abilities, characterizing predisposition for successful work within related field. Personal globe inventory estimates preferences of certain kinds of duties and self-assessed competence in performing each. It partly corresponds to well-known Holland test of profession type preferences with some additional parameters.

1.3. Models training and validating

A gradient boosting algorithm from the XGBoost library, was used. Randomized search with the 5-fold cross-validation was implemented to find the best set of hyperparameters. Experiment was executed for 4 types of input features: semantic tags, color histograms and joined features (including both histograms and semantic information) from gallery images and mean color histogram from pictures, uploaded by user. Input vector, based on tags, was 105 in length, and color histogram input vector was 96 in length. For joined model we use the concatenation of two vectors with total length of 201 respectively. For each of 46 attributes and each input vector type a separate model was designed and trained. For validation purposes, we randomly separate the initial dataset of 868 people, to training and testing in a ratio of 0,25 (217 samples in testing and 651 in the training sets). Additional test (Test 2) was made, using information from second experiment, which had been held after two months after the first. There were 129 participants, who passed validation questions. Data of all these responders were included in the second validation testing dataset. The second dataset was checked to ensure that noone participated both in the first and the second survey.

1.4. Results

Table 1 contains the results of psychometric data prediction using gradient boosting method and different set of input features of the first test. Pearson's Correlation coefficients and p-values between predicted and real values on two tests are presented. This table can allow us to say about gallery pictures preferences as a more powerful predictor, than pictures, uploaded by participant from their own. Also the worst parameter in terms of prediction was Neuroticism. Correlation values higher than 0,3 are highlighted.

Second calculating experiment was made in order to compare our results to [2]. Though Facebook liked tags and image tags are not much in common, it was interesting about tendency, how much different tags we need to predict a personality with high correlation? Fig. 3 describes the dependency of the number of tags, used to describe the gallery images, "liked" by participant, and correlation on Big Five prediction. For all five components there is increasing tendency, so it can be the sign we have an ability to collect more features and make more forceful feature extraction in order to increase the prediction accuracy.

Table 1

Correlations and p-values of psychometric data prediction

Psychometric parameter	Gallery semantic tags		Gallery color histograms		Gallery joined model		Uploaded pictures color histograms	
	Corr	P-value	Corr	P-value	Corr	P-value	Corr	P-value
Openness	0,237	0,000416	0,179	0,007957	0,215	0,001378	0,056	0,406272
Conscientiousness	0,336	0,000000	0,299	0,007957	0,360	0,000000	0,049	0,465445
Extraversion	0,263	0,000086	0,115	0,089099	0,196	0,003731	0,030	0,650520
Agreeableness	0,402	0,000000	0,412	0,000000	0,348	0,000000	0,214	0,001483
Neuroticism	0,081	0,231126	-0,02	0,680963	0,068	0,312462	-0,022	0,743245

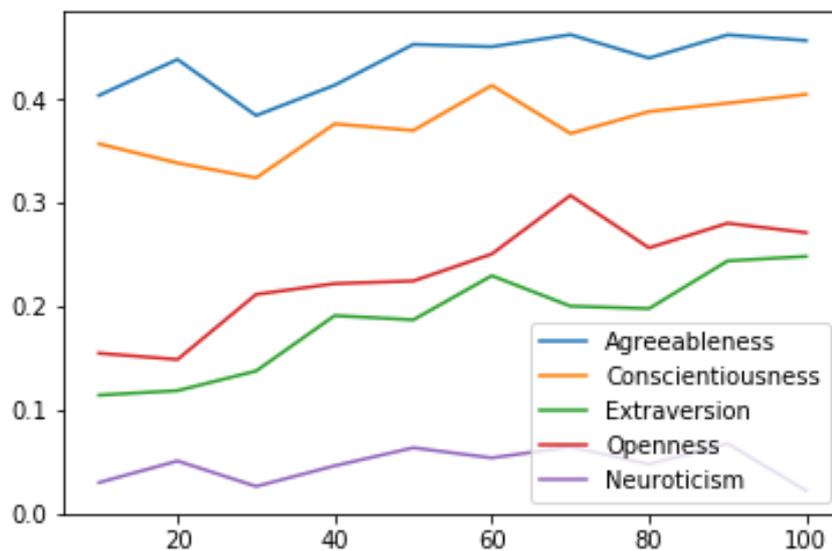


Fig. 1. Computer-based predicted accuracy of Big Five components: on the vertical axis a correlation between predicted values and values, calculated from questionnaire, is shown. Horizontal line describes, how many semantic keywords were used to construct and train the model

1.5. Joined model, based on semantic and color features

In order to increase the accuracy an extended dataset of 1400 participants was tested. Participants involved in this set have passed the gallery test, psychological assessment and answered correctly to validation questions, however not necessarily uploaded their own images. 519

participants from the second survey were used as a validation set (Test 2). A joint model was implemented, based both on semantic tags and color histograms. The result is presented on the Table 2, for each of psychometric parameter. It shows more precise results of psychometric data prediction (correlations higher than 0.4 are highlighted). It is a result of XGBoost method, because this method gave us more accuracy than typical fully-connected neural network architecture. Values of correlations tend to keep from one test to another (organized two month after first survey!). Also mean absolute errors (MAE) are presented in this table. In all cases XGBoost prediction reduced the mean absolute error, comparing to dummy estimator, based on constant (mean value on training set).

Table 2

Accuracy of a 1400 – point joined model

Psychometric parameter	Test 1 correlation	Test 2 correlation	Test 1 MAE	Test 2 MAE	Dummy (constant) MAE
Openness	0,216	0,348	4,070	3,814	4,244
Conscientiousness	0,295	0,419	4,443	4,207	4,980
Extraversion	0,172	0,173	4,288	4,143	4,328
Agreeableness	0,373	0,500	4,486	4,276	5,211
Neuroticism	0,147	0,211	3,660	3,626	3,767

1. Discussion

The results show, that prediction accuracy is very sensitive to the type of the experiment (choose of images from gallery vs free uploading own ones), however certain kind of input features (color vs semantic) and model architecture have no such influence on the result. The dependency was demonstrated between number of tags participant implicitly choosing from during the gallery test, and the correlation coefficients between predicted and real values. This tendency is the same as in [2], where experiment was organized with Facebook “likes”, when real X predicted correlation coefficients grows with number of “data points” in a logarithmic mode. This fact, and the presence of intrapersonal variation [1] justify a hypothesis, that there is a “ceiling” of about 0.7 – 0.8 of a prediction accuracy in terms of correlation. One more observation is that different psychometric attributes can have the extremely different values of prediction accuracy. This can motivate to find diverse forms of collecting media-data from participant from one side, and to find other forms of target psychometric attributes from another. There is a chance, that some of input features and output targets will have higher correlation that will allow us to make a good-quality prediction of personality traits in the future.

References

1. *Gnambs T.* A meta-analysis of dependability coefficients (test-retest reliabilities) for measures of the Big Five // *Journal of Research in Personality.* 2014. Vol. 52. P. 20–28. DOI:10.1016/j.jrp.2014.06.003.
2. *Youyou W., Kosinski M., Stillwell D.* Computers judge personalities better than humans. // *Proceedings of the National Academy of Sciences* Jan 2015. Vol. 112 (4). P. 1036–1040. DOI:10.1073/pnas.1418680112.

3. *Tracey T.* Personal Globe Inventory: Measurement of the Spherical Model of Interests and Competence Beliefs. // *Journal of Vocational Behavior*. 2002. Vol. 60. P. 113–172. DOI:10.1006/jvbe.2001.1817.

4. *Machajdik J., Hanbury A.* Affective Image Classification Using Features Inspired by Psychology and Art Theory // *Proceedings of the 18th ACM International Conference on Multimedia*. Firenze, 2010. P. 83–92.

Принятие решений в области кредитования на основе размытой логики Заде

Е. П. Абрамов

аспирант, Институт систем информатики им. А. П. Ершова СО РАН,
630090, Новосибирск, Россия, пр. Акад. Лаврентьева, 6, e-mail: trueabramov@gmail.com

У. Т. Махажанова

аспирант, Евразийский национальный университет им. Л. Н. Гумилева,
010000, Астана, Казахстан, ул. Сатбаева, 2, e-mail: umt87@mail.ru

Ф. А. Мурзин

канд. физ.-мат. наук, Институт систем информатики им. А. П. Ершова СО РАН,
630090, Новосибирск, Россия, пр. Акад. Лаврентьева, 6, e-mail: murzin@iis.nsk.su

Аннотация. В статье рассматриваются особенности кредитования малого (среднего) бизнеса. Предложен метод оценки кредитоспособности предприятий малого бизнеса, основанный на применении математического аппарата теории нечетких множеств. В настоящее время реализуется программная система для поддержки процесса кредитования.

Ключевые слова: кредитование малого и среднего бизнеса, нечеткая логика, лингвистическая переменная

Введение

Кредитование малого и среднего бизнеса в настоящее время является динамично развивающимся направлением банковской деятельности. В общем виде система оценки кредитоспособности малых предприятий складывается из четырех этапов:

- 1) наблюдение за работой предприятия малого бизнеса;
- 2) собеседование эксперта с владельцем малого предприятия;
- 3) оценка личного финансового положения владельца предприятия;
- 4) анализ финансового положения малого предприятия на основе первичных документов.

Традиционные методы оценки кредитоспособности неприемлемы для малого бизнеса. Это объясняется высоким уровнем ошибок в финансовой отчетности малых предприятий, использованием различных схем ухода от уплаты налогов и т. д. В связи с этим формирование инструментария для оценки кредитоспособности предприятий малого бизнеса является весьма актуальным. Сегодня управление малым предприятием протекает в условиях неопределенности относительно будущего финансового состояния самого предприятия и его экономической среды функционирования, так как этот сектор экономики наиболее подвержен мировому финансовому кризису. С этой точки зрения анализ целесообразности кредитования предприятий малого бизнеса необходимо рассматривать как задачу, решаемую в условиях высокой степени неопределенности [1].

Для решения данной проблемы применяются альтернативные методы оценки кредитоспособности малых предприятий, основанные на нечеткой логике Заде. Как известно, Заде заменил классическое канторовское понятие множества, допустив, что характеристическая функция принадлежности элемента множеству может принимать значения в интервале $[0, 1]$ вещественных чисел, а не только значения 0 либо 1. Такие множества были названы им нечеткими (fuzzy). Далее он ввел понятие лингвистической

переменной и допустил, что в качестве ее значений выступают нечеткие множества. Можно сказать, что Заде создал математический аппарат для описания процессов интеллектуальной деятельности, включающий нечеткость и неопределенность понятий и математических выражений [2].

Процедура принятия решения в обобщенном виде включает: формулирование и сопоставление альтернатив, выбор альтернатив, построение и корректировку программы действий. Формализация процесса кредитоспособности малого предприятия может быть представлена следующим образом.

Предположим, что заемщик обладает неким набором характеристик, которые подлежат оценке при принятии решения о кредитовании, представленных в виде $X = \{x_i : i = \overline{1, N}\}$.

Оценка характеристик происходит на основе набора критериев $C = \{c_j : j = \overline{1, M}\}$.

Далее определяется набор возможных исходов (альтернатив) $A = \{a_t : t = \overline{1, T}\}$.

На следующем этапе строится набор правил нечеткого логического вывода $P = \{p_l, l = \overline{1, L}\}$ вида «Если ..., то ...».

Затем выполняется свертка левой части правил по схемам, которые будут более подробно описаны в разделах 1–2.

В свою очередь выбор той или иной альтернативы также является нечеткой мерой, определяющей такие характеристики предоставляемого кредита, как:

- 1) максимальный размер кредита;
- 2) размер процентной ставки;
- 3) срок кредита.

Определение набора этих показателей и является выбором схемы кредитования $S = \{s_k, k = \overline{1, K}\}$.

1. Показатели оценки кредитоспособности предприятия малого (среднего) бизнеса

Формализуем задачу оценки кредитоспособности предприятия малого бизнеса путем представления упомянутого выше набора параметров в виде лингвистических переменных. Исходные показатели отображаются на интервал $[0,1]$. Подробности в данной работе опускаем. В целом, как именно отображать, этот вопрос требует серьезных исследований. Много полезной информации содержится в литературе [3].

Оценка кредитоспособности конкретной организации – потенциального заемщика, предприятия сферы малого (среднего) бизнеса, может быть представлена следующими лингвистическими переменными (альтернативами):

- 1) a_1 – низкая (L);
- 2) a_2 – ниже среднего (LM);
- 3) a_3 – средняя (M);
- 4) a_4 – выше среднего (HM);
- 5) a_5 – высокая (H).

С каждой лингвистической переменной ассоциируется своя функция принадлежности. Один из возможных способов задания функций принадлежности предложен в работе [2]. В ней рассматривается управление технической системой (паровым котлом), но предложенный там вариант задания функций принадлежности для нашего случая тоже подходит.

В конечном итоге выбор вида функций принадлежности как самих альтернатив, так и оцениваемых критериев определяется исходя из экспертных предпочтений.

Очевидно, что неуверенность эксперта в оценке растет с увеличением отклонения значения оцениваемого параметра от оптимального значения. Более того, в большинстве случаев данная неуверенность растет не линейно. Однако применение нелинейных функций

принадлежности влечет за собой существенное усложнение математических расчетов и графических построений.

В силу этих причин в данной работе в качестве исходных функций принадлежностей используются треугольные и трапецеидальные функции принадлежности, что в первую очередь обусловлено простотой выполнения последующих вычислений и графических построений. Можно сказать, что функции принадлежности кусочно линейные.

Определение степени соответствия выбранного набора оцениваемых критериев той или иной альтернативе является ключевым фактором при последующем выборе наиболее подходящей схемы кредитования, обладающей оптимально подобранным набором условий для последующего кредитования.

Набор оцениваемых параметров был определен ранее в работе [4], и он приведен ниже (табл. 1).

Таблица 1

Финансовые и общеэкономические показатели

№№	Исходные показатели	Значение показателя
1	Динамика развития отрасли	Ниже среднего – LM
2	Перспективы развития отрасли	Ниже среднего – LM
3	Потребность рынка (отрасли) в подобного рода продукции (работе, услугах)	Средняя – М
4	Динамика развития экономики региона	Средняя – М
5	Перспективы развития экономики региона	Выше среднего – НМ
6	Потребность рынка (региона) в подобного рода продукции (работе, услугах)	Средняя – М
7	Коэффициент текущей платежеспособности и ликвидности	Ниже среднего – LM $K_{\text{тпл}}=1,35603$
8	Коэффициент долгосрочной финансовой независимости	Средняя – М $K_{\text{дфн}}=0,3691$
9	Коэффициент обеспеченности запасов собственным оборотным капиталом	Низкая – L $K_{\text{озсок}}=0,26878$
10	Коэффициент покрытия	Низкая – L $K_{\text{п}}=0,5967$
11	Коэффициент оборачиваемости дебиторской задолженности	Высокая – Н $K_{\text{одз}} = 20,7878$
12	Коэффициент оборачиваемости кредиторской задолженности	Средняя – М $K_{\text{окз}}=85,6914$
13	Коэффициент оборачиваемости готовой продукции	Низкая – L $K_{\text{огп}} = 30,7515.$
14	Коэффициент соотношения собственных и заемных средств	Ниже среднего – LM $K_{\text{ссзс}}=0,585$
15	Коэффициент рентабельности продукции (продаж)	Низкая – L $K_{\text{рп}} = 0,0379$
16	Оценка профессионального уровня кадрового состава	Ниже среднего – LM
17	Оценка морально-психологической атмосферы на предприятии	Средняя – М
18	Достаточность срока пребывания предприятия на рынке	Выше среднего – НМ
19	Экономическая политика предприятия	Средняя – М
20	Техническая политика предприятия	Ниже среднего – LM
21	Кадровая политика предприятия	Ниже среднего – LM
22	Кредитная история заемщика (отсутствует)	Низкая – L

2. Процесс принятия решений

Отметим, что параметры (показатели) принимают довольно произвольные значения. Более точно, каждый параметр изменяется в определенном присущем ему интервале. Далее можно произвести унификацию параметров, т. е. отобразить соответствующие интервалы на отрезок $[0,1]$. Детали мы здесь опускаем.

С каждым i -м показателем связываем три одноместных предикате $P_L^i(x)$, $P_M^i(x)$, $P_H^i(x)$.

Для простоты введем также формульно определимые предикаты:

$$P_{LM}^i(x) = P_L^i(x) \vee P_M^i(x);$$

$$P_{HM}^i(x) = P_M^i(x) \vee P_H^i(x).$$

Аналогично с каждым финансовым показателем связываем предикате Q_D^i , $D \in \{L, M, H, LM, HM\}$.

Таблица 2

Характеристики предоставляемого кредита

№	Результирующие показатели
1	Объем кредитования
2	Процентная ставка
3	Срок кредитования

Правила, на основе которых принимаются решения, имеют вид

$$\varphi(x_1, \dots, x_n) \rightarrow \psi(y_1, \dots, y_m).$$

Обычно $\varphi(x_1, \dots, x_n) = \bigwedge_{j=1}^n Q_j(x_j)$, т. е. имеем конъюнкцию одноместных предикатов указанной выше сигнатуры.

При этом каждое Q_j имеет вид P_D^i , $D \in \{L, M, H, LM, HM\}$.

В итоге получаем правило

$$P_{LM}^1(x_1) \wedge P_{LM}^2(x_2) \wedge P_M^3(x_3) \wedge P_M^4(x_4) \wedge P_{HM}^5(x_5) \wedge P_M^6(x_6) \wedge P_{LM}^7(x_7) \wedge P_M^8(x_8) \wedge P_L^9(x_9) \wedge P_L^{10}(x_{10}) \wedge P_H^{11}(x_{11}) \wedge P_M^{12}(x_{12}) \wedge P_L^{13}(x_{13}) \wedge P_{LM}^{14}(x_{14}) \wedge P_L^{15}(x_{15}) \wedge P_{LM}^{16}(x_{16}) \wedge P_M^{17}(x_{17}) \wedge P_{HM}^{18}(x_{18}) \wedge P_M^{19}(x_{19}) \wedge P_{LM}^{20}(x_{20}) \wedge P_{LM}^{21}(x_{21}) \wedge P_L^{22}(x_{22}) \rightarrow Q_M^1(y_1).$$

Может использоваться несколько правил

$$R_i : \varphi_i(x_1 \dots x_n) \rightarrow \psi_i(y_1 \dots y_m), i = 1, \dots, N.$$

Для краткости пишем $R_i = R_i(x_1 \dots x_n, y_1 \dots y_m)$.

Можно рассмотреть их дизъюнкцию

$$R = \bigvee_i R_i.$$

Параметры $x_1 \dots x_n$ получаются в результате анализа деятельности предприятия. Это показатели, упомянутые в разделе 1. Параметры $y_1 \dots y_m$ предсказываются. Это такие показатели, как объем кредита, процентная ставка, длительность кредита. В самом общем виде предсказывается один параметр называемый индекс кредитоспособности, меняющийся от 0 до 1 и имеющий естественную интерпретацию: предпочтение отдается предприятиям с более высоким индексом кредитоспособности.

В настоящее время исследование и разработки ведутся по следующим направлениям:

- 1) формулирование новых правил кредитования;
- 2) разработка правил унификации разнородных данных;
- 3) апробирование алгоритмов в системах Matlab, Maple, Statistica, Excel;

4) проектирование системы, ориентированную на будущую реализацию средствами Python, Visual C++, MySQL.

Список литературы

1. *Заболоцкая В. В., Аристархов А. А.* Методика оценки кредитоспособности предприятий малого бизнеса // Финансы и кредит. 2009. № 12(348). С. 61–73.
2. *Пивкин В. Я., Бакулин Е. П., Кореньков Д. И.* Нечеткие множества в системах управления : учеб. пособие. Новосибирск : Изд-во НГУ, 1997. 52 с.
3. *Синянская Е. Р., Баженов О. В.* Основы бухгалтерского учета и анализа : учеб. пособие. Екатеринбург : Изд-во Уральского ун-та, 2014. 270 с.
4. *Илларионов А. В.* Разработка математических моделей и алгоритмов принятия решений по кредитованию предприятий малого (среднего) бизнеса на основе аппарата теории нечетких множеств: дис. ... канд. экон. наук. Владимир, 2006.

UDK 51-74

DECISION-MAKING IN THE FIELD OF CREDITING ON THE BASIS OF ZADEN FUZZY LOGIC

E. P. Abramov

Ph.D. student, A. P. Ershov Institute of Informatics Systems SB RAS,
630090, Novosibirsk, Russia, Acad. Lavrentjev pr., 6, e-mail: trueabramov@gmail.com

U. T. Makhazhanova

Ph.D. student, L. N. Gumilyov Eurasian National University,
010000, Astana, Kazakhstan, Satbayev str., 2, umt87@mail.ru

F. A. Murzin

Ph.D., deputy director, A.P. Ershov Institute of Informatics Systems SB RAS,
630090, Novosibirsk, Russia, Acad. Lavrentjev pr., 6, e-mail: murzin@iis.nsk.su

Abstract. The article discusses the features of crediting (lending) of small (medium) businesses. A method for assessing the creditworthiness of small businesses based on the use of the mathematical apparatus of the theory of fuzzy sets is proposed. In the current time, software system is being implemented to support the crediting process.

Keywords: SME lending, fuzzy logic, linguistic variable

References

1. *Zabolotskaya V. V., Aristarkhov A. A.* Metodika otsenki kreditosposobnosti predpriyatiy malogo biznesa [Methods for assessing the creditworthiness of small businesses] // Finansy i credit = Finance and credit. 2009. № 12(348). P. 61–73. (in Russian)
2. *Sinyanskaya E. R., Bazhenov O. V.* Osnovy bukhgalterskogo ucheta i analiza [Basics of accounting and analysis]: uchebnoye posobiye = the tutorial.– Yekaterinburg : Publishing House of the Ural University 2014. 270 p. (in Russian)
3. *Pivkin V. Ya., Bakulin E. P., Korenkov D. I.* Nechetkiye mnozhestva v sistemakh upravleniya [The concept of a linguistic variable and its application to making approximate decisions]: uchebnoye posobiye = the tutorial. Novosibirsk : NSU, 1997. 52 p. (in Russian)
4. *Illarionov A. V.* Razrabotka matematicheskikh modeley i algoritmov prinyatiya resheniy po kreditovaniyu predpriyatiy malogo (srednego) biznesa na osnove apparata teorii nechetkikh mnozhestv [Development of mathematical models and decision-making algorithms for

lending to small (medium) businesses based on the apparatus of the theory of fuzzy sets]: dis. ... kand. ekon. nauk = dis. ... Cand. econ sciences. Vladimir, 2006.

Принципы построения системы детерминированного параллельного программирования

А. И. Адамович

ст. науч. сотр., Институт программных систем им. А. К. Айламазяна РАН,
152021, Ярославская обл., Переславский район, с. Вельково, Россия, ул. Петра Первого,
д. 4 «а», e-mail: lexa@adam.botik.ru

А. В. Климов

ст. науч. сотр., Институт прикладной математики им. М. В. Келдыша РАН,
125047, Москва, Россия, Миусская пл., д. 4, e-mail: klimov@keldysh.ru

Аннотация. В связи с взрывным ростом сложности программ для многоядерных процессоров и суперкомпьютеров в последнее десятилетие приобретает популярность и становится все более актуальной идея параллельных вычислений с детерминированностью, гарантированной языком и системой программирования. В докладе анализируется проблема, как сделать параллельное программирование как можно более детерминированным. Дается обзор некоторых подходов к ее решению. Описываются принципы построения системы программирования, разрабатываемой авторами, предоставляющей возможность писать как детерминированный, так и недетерминированный код с гарантиями прикладному программисту, что его программа будет детерминированной. Характеризуются некоторые показательные классы задач, реализуемые в данной системе.

Ключевые слова: модели параллельных вычислений, детерминированные программы, функциональное программирование, объектно-ориентированное программирование, монотонные объекты

Введение

Параллельные и конкурентные (англ. *concurrent*) программы, в общем случае недетерминированные, дают разные результаты при нескольких прогонах, так как для эффективной реализации на современной аппаратуре требуется явное использование таких средств программирования, как процессы, потоки, треды (англ. *threads*), читающие и изменяющие общие ресурсы и дающие разные результаты при различном порядке доступа потоков к ресурсам. Отладка, модификация, сопровождение таких программ намного более трудозатратны, чем привычных массовым программистам детерминированных последовательных программ. Поэтому многие языки высокого уровня «прячут» от «обычного» программиста средства конкурентного программирования на уровень реализации и в библиотеки, разрабатываемые экспертами. Однако такие решения ограничивают изобразимые на этих языках классы программ и вынуждают писать менее эффективные программы, немасштабируемые при увеличении числа процессоров, ядер и других аппаратных средств параллельного исполнения.

Таким образом мы имеем следующую ситуацию. Во-первых, детерминированные языки параллельного и конкурентного программирования существуют и развиваются (в качестве яркого примера приведем чисто функциональный язык Haskell и его библиотеки для параллельного программирования). Во-вторых, нет и не может быть одного языка и библиотеки детерминированного параллельного программирования, который удовлетворил бы все потребности, даже если зафиксировать круг языковых понятий конкурентного

программирования, который был бы базой для детерминированного, например, объектно-ориентированные языки типа Java с понятием тредов (англ. *threads*). В результате значительно разнесены уровень детерминированного программирования, считающийся «высоким», и уровень реализации языков и библиотек, считающийся «низким». На этих уровнях используются сильно различающиеся языки и инструменты и требуются разные степени квалификации разработчиков — столь же отличающиеся, как, например, у разработчиков компиляторов и систем программирования и их пользователей.

Возникает вопрос: на сколько можно приблизить эти уровни? Нельзя ли в рамках одного языка программирования (пусть это будет объектно-ориентированный язык типа Java) дать и универсальные средства недетерминированного программирования для создания базовых инструментов и библиотек, и определить «высокий» уровень, подязык, проверяемый компилятором, при программировании на котором гарантируется детерминированность? «Нижний», универсальный уровень потребуется для постоянного расширения и развития проблемно-ориентированных библиотек для реализации определенных классов алгоритмов. Здесь требуются повышенные трудозатраты для отработки библиотек. Но потом они используются не в одной, а во многих прикладных программах данного класса, программирование и отладка которых намного легче и дешевле благодаря детерминированности, гарантированной авторами библиотек и системы программирования.

Авторы данного доклада ставят цель дать положительный ответ на этот вопрос в виде двухуровневой системы программирования на языке типа Java (реализованному на виртуальной Java-машине, JVM). Нижний, базовый уровень — это язык Java, на котором определяются классы, используемые на «верхнем» уровне. Верхний уровень — это Java-подобный язык (или Java-подмножество), напоминающий чисто функциональный язык с естественной (для функциональных языков) параллельной реализацией и с объектно-ориентированными расширениями, обеспечивающими использование классов «нижнего» уровня, не нарушая детерминированности параллельных вычислений. Библиотечные классы нижнего уровня и их объекты мы называем *монотонными* (по причинам, которые объясним ниже). Проект продолжает наши работы по T-системе и монотонным объектам [1–6].

Данный доклад основан на статье авторов [7], содержащей обзор основных средств детерминированного параллельного программирования, разработанных к настоящему времени, и нашими предложениями по дальнейшему развитию этих средств. Здесь мы даем «выжимку» этого обзора, приводим обоснование и архитектуру разрабатываемой нами системы и кратко характеризуем два класса задач, на которых продемонстрируем методику создания детерминированных параллельных программ в наших будущих работах.

1. Обзор средств детерминированного параллельного программирования

1.1. Детерминированный параллелизм чисто функциональных языков

Отправной точкой, общей основой многих языков со средствами параллельного исполнения являются функциональные языки, которые в своем «чистом» виде не имеют побочных эффектов и потому «легко» распараллеливаются. Распараллеливание кода на функциональном языке, конечно, подразумевает сохранение семантики языка, т. е. эквивалентность результата параллельного исполнения каждой программы эталонному последовательному, а также денотационной семантике, тем самым — детерминированность.

Максимально параллельное исполнение функциональной программы делается вызовами каждой функции в своем параллельном процессе. Это отнюдь не является эффективным решением в общем случае: наоборот, возникает проблема ограничить параллелизм, чтобы эффективно использовать ограниченные аппаратные ресурсы, т. е. не «распараллелить», а «секвенциализировать» (англ. *sequentialize*), объединить потенциально параллельные группы вычислений в последовательный поток управления, тред. Прагматичное решение (которому мы следуем в нашем проекте) — дать программисту языковые средства обозначать, где

параллельные вызовы, а где последовательный код. Если двигаться с другой стороны — от последовательных языков к параллельным —, то аналогичное решение наблюдаем, когда вводится конструкция (или библиотечные средства), называемая в ряде языков «future»: вызов функции в отдельном процессе, исполняемом параллельно до тех пор, пока вызвавшему процессу не понадобится результат и он не встанет на ожидание завершения.

Большинство языков, называемых функциональными, — «грязные», т. е. позволяют побочные эффекты и не прячут понятие параллельного процесса, треда, позволяя явно управлять ими, как и в массовых объектно-ориентированных языках со средствами параллелизма. Из распространенных функциональных языков лишь Haskell сохраняет «чистоту» и осторожно расширяется средствами параллелизма — как правило, библиотечными с соответствующей поддержкой на нижнем уровне в реализации языка [8].

Среди большого разнообразия публикаций по этому направлению особенно интересен сборник, подводящий итоги по состоянию на конец 1990-х годов [9]. Не выходя за рамки функциональной модели вычислений также строятся проблемно-ориентированные декларативные языки для конкретных областей применения. Например, таким является язык Норма [10] для решения сеточных задач некоторого класса из математической физики.

С другой стороны, современные объектно-ориентированные языки содержат в качестве своего подмножества чисто функциональный язык, на котором можно программировать в функциональном стиле без побочного эффекта. Таковы Java, C#, Kotlin, JavaScript и др. Здесь основное неудобство — то, что компилятор не проверяет принадлежность функциональному подмножеству и соблюдение этого условия — дело программиста.

1.2. Неизменяемые объекты, *immutability*

Если двигаться со стороны императивных и объектно-ориентированных языков, существуют ряд работ по введению ограничений, проверяемых компилятором и / или во время исполнения и обеспечивающих нужные свойства. Если потребовать, чтобы все объекты были неизменяемыми (англ. *immutable*), т. е. их состояние не менялось после отработки инициализаторов, то объектно-ориентированный язык практически превращается в функциональный, которому присущ детерминированный параллелизм.

На практике трудно обходиться совсем без изменений и побочных эффектов, поэтому понятию неизменяемости (англ. *immutability*) придают различную степень «строгости». Статья [10] содержит хороший обзор работ по этой теме.

1.3. Статические методы обеспечения детерминированности

Имеется много работ по статическому анализу кода, содержащему побочный эффект, имеющих целью выявить частные случаи, когда параллельное исполнение сохраняет детерминированность результата из-за отсутствия «гонок». Из ряда работ, упомянутых в нашем обзоре [7], здесь отметим разработку языка Deterministic Parallel Java, DPJ [12] как имеющую прагматическую цель в рамках популярного объектно-ориентированного языка. В наших работах эти результаты не используются, мы подходим к задаче с другой стороны.

1.4. Обеспечение детерминированности операциями над данными

Следующий подход к обеспечению детерминированности параллельных программ, в отличие от методов предыдущего раздела, не использует никаких статических средств анализа. За основу берется чисто функциональный язык программирования, быть может, со всевозможными расширениями, не нарушающими функциональность и распараллеливаемость. Затем для взаимодействия параллельных процессов предоставляются специальные структуры данных с такими определенными операциями, чтобы не нарушалась детерминированность. Такая идея породила целое направление, например:

- 1) I-структуры (англ. *I-structures*) [13];
- 2) сети Кана (англ. *Kahn networks*) [14];
- 3) TStreams, Concurrent Collections [15];
- 4) структуры данных на решетках (англ. *lattice-based data structures*) [15, 16].

У этих подходов есть общая черта: переменные, объекты, через которые осуществляется взаимодействие параллельных процессов, изменяют свое состояние монотонно, только вверх на некоторой полурешетке от неопределенного состояния (\perp) к «все более определенному». При этом верхний элемент решетки (Т) обозначает «переопределено»; в программе это соответствует ошибке, выработке исключения. Например, множество значений I-структур с целыми числами описывается решеткой, называемой «плоской», состоящей из нижнего элемента «не определено» (\perp), не сравнимых между собой целых чисел и верхнего элемента «переопределено» (Т). При выполнении операции присваивания значения y в переменную со значением x в нее записывается наименьшая верхняя грань значений x и y . Если полученный результат оказывается верхним элементом Т, то вырабатывается исключение.

Эта идея в общем виде была проработана в диссертации Линдси Купер [15] и в публикациях вместе с ее коллегами [16]. Она доказала детерминированность параллельных вычислений для процессов, взаимодействующих через переменные, принимающие значения из произвольной (полу)решетки.

В нашем проекте мы используем эти идеи, обобщая их на объекты, определяемые пользователем, с монотонно изменяющимся состоянием.

2. Архитектура двухуровневой системы детерминированного параллельного программирования

Поскольку не существует единого набора средств, зафиксированных в языке программирования и библиотеке, обеспечивающих все случаи детерминированного параллельного программирования, входной язык должен позволять как детерминированный, так и недетерминированный код. Но чтобы прикладной программист мог ограничивать себя средствами гарантированно детерминированного программирования, входной язык системы должен быть двухуровневым, т. е. код должен четко подразделяться на две части: гарантированно детерминированную и потенциально недетерминированную:

1) верхний уровень — детерминированная часть: прикладной код на подмножестве языка, который пишет эксперт в данной предметной области. Ему гарантируется детерминированность любой его программы, использующей заготовленные библиотеки нижнего уровня. Принадлежность детерминированному подмножеству проверяется компилятором. Оно примерно соответствует функциональному подмножеству языка Java и ему подобных;

2) нижний уровень — недетерминированная часть: библиотеки классов, создаваемые квалифицированными программистами для определенных областей применения на универсальном объектно-ориентированном языке (типа Java) с богатым набором изобразительных средств, позволяющим кодировать недетерминированные параллельные алгоритмы. Авторы библиотек гарантируют детерминированность параллельных программ их пользователям, кодирующим на функциональном подмножестве языка на верхнем уровне. Такие классы и объекты мы называем *монотонными*, поскольку, как и в работах упомянутых в предыдущем разделе 1.4, их состояние меняется монотонно на некоторой (полу)решетке, которую, в принципе, можно построить по коду класса. В будущем мы рассчитываем, что удастся разработать средства автоматизации доказательств в подавляющем числе случаев, что определение данного класса или группы класса является монотонным и гарантирует детерминированность.

Входным языком может быть любой объектно-ориентированный язык со средствами параллельного и конкурентного программирования, у которого можно выделить функциональное подмножество. Такими являются большинство современных объектно-

ориентированных языков. Однако в таком случае некоторые детали эффективной реализации будут «торчать» и загромождать прикладной код. Поэтому мы разрабатываем специализированный Java-подобный язык, названный Ajl [3], в котором эти «детали» будут прикрыты синтаксическим сахаром и генерироваться из привычного кода. Таким образом мы получаем степень свободы предложить пользователям разные механизмы реализации параллельных процессов, тредов, легких тредов (англ. *light-weight thread*), фиберов (англ. *fiber*), например, использовать (или не использовать) стренды (англ. *strand*) библиотеки системы Quasar (URL: <https://github.com/puniverse/quasar>.)

3. Примеры классов задач

Для изучения и демонстрации возможностей программирования с монотонными объектами в настоящее время мы обкатываем систему на двух классах задач, представленных ниже.

1. Порождение и обработка графов. Ценность этого класса задач в том, что классические чисто функциональные языки не дают возможности обрабатывать графы эффективно, когда узлы (и, быть может, дуги) представлены объектами, а связи между ними — ссылками в полях объектов. В функциональных языках можно эффективно представлять только деревья. В системе программирования с монотонными объектами мы сохраняем большинство положительных свойств функциональных языков (детерминированность — лишь одно из них) и расширяем области эффективно представимых данных.

2. Переборные алгоритмы на графах типа поиска кратчайшего пути методом ветвей и границ (англ. *branch-and-bound*). Они интересны тем, что по смыслу задачи в них присутствуют несколько видов монотонно изменяемых данных: монотонно растут пути, причем их много и нужно отличать их друг от друга; монотонно изменяется рекорд (кратчайший путь, найденным к настоящему времени), причем порядок его изменения зависит от порядка исполнения параллельных процессов, а результат не зависит. Кроме того, необходима эквивалентность реализаций с полным перебором и с «отсевом» (англ. *pruning*) вариантов, которые заведомо не улучшат рекорд. Гарантии монотонности и эквивалентности должны быть реализованы на «нижнем уровне» в как можно меньшем по размеру коде специально сконструированных монотонных классов. Здесь появляются «объекты высшего порядка», методы которых принимают лямбда-выражения, описывающие продолжения вычислений, которые могут подвергнуться «отсеву».

Более подробная информация о методике программирования этих классов задач будет опубликована отдельно в наших будущих работах.

Заключение

В докладе дана мотивировка и принципы построения системы детерминированного параллельного программирования, разрабатываемой авторами. Представлен краткий обзор предшествующих работ, идеи которых используются в данной разработке. Охарактеризованы два класса прикладных задач, на которых система отрабатывается и будет продемонстрирована в наших будущих публикациях.

Список литературы

1. *Абрамов С. М., Адамович А. И., Коваленко М. Р.* Т-система — среда программирования с поддержкой автоматического динамического распараллеливания программ. Пример реализации алгоритма построения изображений методом трассировки лучей // Программирование. 1999. № 25:2. С. 100–107.
2. *Адамович А. И.* Струи как основа реализации понятия Т-процесса для платформы JVM // Программные системы: теория и приложения. 2015. № 6:4 (27). С. 177–195. DOI: 10.25209/2079-3316-2017-8-4-221-244.
3. *Адамович А. И.* Язык программирования Ajl: автоматическое динамическое распараллеливание для платформы JVM // Программные системы: теория и приложения. 2016. № 7:4 (31). С. 83–117. DOI: 10.25209/2079-3316-2016-7-4-83-117.
4. *Адамович А. И., Климов А. В.* Об опыте использования среды метапрограммирования Eclipse/TMF для конструирования специализированных языков // Научный сервис в сети Интернет. Труды XVIII Всеросс. науч. конф. М. : ИПМ им. М.В. Келдыша, 2016. С. 3–8. DOI: 10.20948/abrau-2016-45.
5. *Klimov A. V.* Dynamic Specialization in Extended Functional Language with Monotone Objects // SIGPLAN Not., 1991. № 26:9. P. 199–210. DOI: 10.1145/115865.376287.
6. *Климов А. В.* Детерминированные параллельные вычисления с монотонными объектами // Научный сервис в сети Интернет: многоядерный компьютерный мир. Труды Всеросс. науч. конф. М. : Изд-во Московского ун-та, 2007. С. 212–217.
7. *Адамович А. И., Климов А. В.* Как создавать параллельные программы, детерминированные по построению? Постановка проблемы и обзор работ // Программные системы: теория и приложения. 2017. № 8:4(35). С. 221–244. DOI: 10.25209/2079-3316-2017-8-4-221-244.
8. *Marlow S. S.* Parallel and Concurrent Programming in Haskell. O'Reilly, CA, USA, 2013.
9. *Hammond K., Michelson G. (eds.).* Research Directions in Parallel Functional Programming, Springer-Verlag, London, UK, 2000.
10. *Андрианов А. Н., Баранова Т. П., Бугеря А. Б., Ефимкин К. Н.* Трансляция непроцедурного языка Норма для графических процессоров // Препринты ИПМ им. М. В. Келдыша. 2016. № 73. 24 с. DOI: 10.20948/prepr-2016-73.
11. *Potanie J., Östlund Y., Zibin M. D.* Ernst. Immutability // Aliasing in Object-Oriented Programming, eds. D. Clarke, J. Noble, T. Wrigstad. Springer-Verlag, 2013. P. 233–269.
12. *Vocchino (Jr.) R. L., Adve V. S., Snir M.* Parallel Programming Must Be Deterministic by Default // Fifth USENIX Conference on Hot Topics in Parallelism, HotPar'09. USENIX Association, 2009. P. 4–4.
13. *Arvind R. S. Nikhil K. K. Pingali.* I-structures: Data Structures for Parallel Computing // ACM Trans. Program. Lang. Syst., 1989. № 11:4. P. 598–632. DOI: 10.1145/69558.69562.
14. *Kahn G.* The Semantics of Simple Language for Parallel Programming // IFIP Congress, 1974. P. 471–475.
15. *Burke M. G., Knobe K., Newton R., Sarkar V.* Concurrent Collections Programming Model // Encyclopedia of Parallel Computing. Springer US, 2011. P. 364–371. DOI: 10.1007/978-0-387-09766-4_238.
16. *Kuper L.* Lattice-based Data Structures for Deterministic Parallel and Distributed Programming : Ph.D. Thesis, 2015.
17. *Kuper L., Todd A., Tobin-Hochstadt S., Newton R. R.* Taming the Parallel Effect Zoo: Extensible Deterministic Parallelism with LVish // ACM SIGPLAN Not. 2014. № 49:6. P. 2–14. DOI: 10.1145/2594291.2594312.

PRINCIPLES OF CONSTRUCTING A DETERMINISTIC PARALLEL PROGRAMMING SYSTEM

A. I. Adamovich

Senior researcher, Ailamazyan Program Systems Institute of Russian Academy of Sciences
152021, Pereslavl region, Yaroslavl oblast, Russia, selo Veskovo, Peter I str. 4a, e-mail:
lexa@adam.botik.ru

A. V. Klimov

Senior researcher, Keldysh Institute of Applied Mathematics of Russian Academy of Sciences
125047, Moscow, Russia, Miusskaya sq., 4, e-mail: klimov@keldysh.ru

Annotation. Due to the explosive growth of complexity of programs for multi-core processors and supercomputers, the idea of parallel computing with determinism guaranteed by the programming language and system is becoming increasingly significant. This paper analyzes the problem of how to make parallel programming as deterministic as possible. An overview of some approaches to its solution is given. It describes the principles of constructing a programming system developed by the authors, which provides the opportunity to write both deterministic and non-deterministic code with guarantees to the application programmer that his program will be deterministic. Some representative classes of tasks implemented in this system are characterized.

Keywords: models of parallel computation, deterministic programs, functional programming, object-oriented programming, monotonic objects

References

1. Abramov S. M., Adamovich A. I., Kovalenko M. R. T-sistema — sreda programirovanija s podderzhkoj avtomaticheskogo dinamičeskogo rasparallelivanija programm. Primer realizacii algoritma postroenija izobrazhenij metodom trassirovki lucnej [T-System — An Environment Supporting Automatic Dynamic Parallelization of Programs: An Example of the Implementation of an Image Rendering Algorithm Based on the Ray Tracing Method]. Programirovanie = Programming. 1999. № 25:2. P. 100–107.
2. Adamovich A. I. Strui kak osnova realizacii ponjatija T-processa dlja platformy JVM [Fibers as the basis for the implementation of the notion of the T-process for the JVM platform]. Programmnye sistemy: teorija i prilozhenija = Program Systems: Theory and Applications. 2015. № 6:4 (27). P. 177–195. DOI: 10.25209/2079-3316-2017-8-4-221-244.
3. Adamovich A. I. Jazyk programirovanija Ajl: avtomaticheskoe dinamičeskoe rasparallelivanie dlja platformy JVM [The Ajl programming language: the automatic dynamic parallelization for the JVM platform]. Programmnye sistemy: teorija i prilozhenija = Program Systems: Theory and Applications, 2016. № 7:4 (31). P. 83–117. DOI: 10.25209/2079-3316-2016-7-4-83-117.
4. Adamovich A. I., Klimov A. V. Ob opyte ispol'zovanija sredy metaprogramirovanija Eclipse/TFM dlja konstruirovannija specializirovannyh jazykov [On the experience of using the environment metaprogramming Eclipse/TFM for designing specialized languages]. Nauchnyj servis v seti Internet. Trudy XVIII Vserossijskoj nauchnoj konferencii. M. : IPM im. M.V. Keldysha, 2016. P. 3–8. DOI: 10.20948/abrau-2016-45.
5. Klimov A. V. Dynamic Specialization in Extended Functional Language with Monotone Objects. SIGPLAN Not., 1991. № 26:9. P. 199–210. DOI: 10.1145/115865.376287.
6. Klimov A. V. Determinirovannye parallel'nye vychislenija s monotonnymi ob'ektami [Deterministic parallel computation with monotonic objects]. Nauchnyj servis v seti Internet:

mногоядерный комп'ютерный мир. Trudy Vserossijskoj nauchnoj konferencii. M. : MSU Publ., 2007. P. 212–217.

7. *Adamovich A. I., Klimov A. V.* Kak sozdavat' parallel'nye programmy, determinirovannye po postroeniju? Postanovka problemy i obzor rabot [How to create deterministic by construction parallel programs? Problem statement and survey of related works]. Programmnye sistemy: teorija i prilozhenija = Program Systems: Theory and Applications. 2017. № 8:4(35). P. 221–244. DOI: 10.25209/2079-3316-2017-8-4-221-244.

8. *Marlow S. S.* Parallel and Concurrent Programming in Haskell. O'Reilly, CA, USA, 2013.

9. *Hammond K., G. Michelson (eds.)*. Research Directions in Parallel Functional Programming, Springer-Verlag, London, UK, 2000.

10. *Andrianov A. N., Baranova T. P., Bugerya A. B., Yefimkin K. N.*: Transljacija neprocedurnogo jazyka Norma dlja graficheskikh processorov [Nonprocedural NORMA language translation for GPUs]. Preprinty IPM im. M.V. Keldysha = Keldysh Institute Preprints. 2016. №. 73. 24 p. DOI: 10.20948/prepr-2016-73. (In Russian)

11. *Potanin J., Östlund Y., Zibin M. D.* Ernst. Immutability. Aliasing in Object-Oriented Programming, eds. D. Clarke, J. Noble, T. Wrigstad. Springer-Verlag, 2013. P. 233–269.

12. *Bocchino (Jr.) R. L., Adve V. S., Adve S. V., Snir M.* Parallel Programming Must Be Deterministic by Default // Fifth USENIX Conference on Hot Topics in Parallelism, HotPar'09. USENIX Association, 2009. P. 4.

13. *Arvind R. S. Nikhil K. K.* Pingali. I-structures: Data Structures for Parallel Computing // ACM Trans. Program. Lang. Syst. 1989. № 11:4. P. 598–632. DOI: 10.1145/69558.69562.

14. *Kahn G.* The Semantics of Simple Language for Parallel Programming // IFIP Congress. 1974. P. 471–475.

15. *Burke M. G., Knobe K., Newton R., Sarkar V.* Concurrent Collections Programming Model. Encyclopedia of Parallel Computing. Springer US. 2011. P. 364–371. DOI: 10.1007/978-0-387-09766-4_238.

16. *Kuper L.* Lattice-based Data Structures for Deterministic Parallel and Distributed Programming : Ph.D. Thesis, 2015.

17. *Kuper L., Todd A., Tobin-Hochstadt S., Newton R. R.* Taming the Parallel Effect Zoo: Extensible Deterministic Parallelism with LVish // ACM SIGPLAN Not. 2014. № 49:6. P. 2–14. DOI: 10.1145/2594291.2594312.

Теория социального влияния Латане и ее применение для анализа окружения пользователя в социальной сети

А. Ж. Ахметова

аспирант, Евразийский национальный университет им. Л. Н. Гумилева,
010000, Астана, Казахстан, ул. Сатбаева, 2, e-mail: akhmetova_azh@mail.ru

Л. Л. Ла

канд. физ.-мат. наук, доц., Евразийский национальный университет им. Л. Н. Гумилева,
010000, Астана, Казахстан, ул. Сатбаева, 2, e-mail: lira_la@hotmail.com

Ф. А. Мурзин

канд. физ.-мат. наук, зам. дир., Институт систем информатики им. А. П. Ершова СО РАН,
630090, Новосибирск, Россия, пр. Акад. Лаврентьева, 6, e-mail: murzin@iis.nsk.su

Аннотация. В статье речь идет об анализе данных из компьютерных социальных сетей. Предложены модификации теории динамического социального влияния Латане. Видоизменены формулы, описывающие количество социального давления, направленного на индивидуума для различных ситуаций. В статье кратко сообщается о разработанном программном комплексе, позволяющем извлекать информацию из социальных сетей, проводить обработку, анализ и визуализацию данных. Тестирование производилось на данных, полученных из социальной сети vkontakte. Исследования выполнены при финансовой поддержке Министерства образования и науки Республики Казахстан (грант 2018-2020 MES RK № AP05133550) и Интеграционного проекта СО РАН (AAAA-A18-118022190008-8).

Ключевые слова: анализ социальных сетей, интернет, vkontakte, анализ данных, теория Латане

Введение

Социальная сеть представляет собой интерактивный многопользовательский веб-сайт, содержание (контент) которого наполняется самими участниками сети. Сайт представляет собой автоматизированную социальную среду, позволяющую общаться группе пользователей, объединенных общими интересами. Количество зарегистрированных пользователей только одной сети может превышать население целой страны. Социальные сети открывают новые возможности для решения исследовательских и бизнес-задач. Этим обуславливается повышенный интерес к сбору и анализу социальных данных со стороны компаний и исследовательских центров.

Анализ социальных сетей [1, 2] используется для исследования взаимодействий между участниками сети, прогнозирования их поведения, классификации, моделирования информационных потоков в сетях. Однако отметим, что социальная сеть – это инструмент, который может активно использоваться государствами для формирования и манипулирования общественным мнением. В ряде стран (США, Великобритания, Китай, Индия, Франция и т. д.) приняты решения об исследовании и использовании социальных сетей в интересах этих стран для моделирования социальных, экономических, политических и других процессов и разработки механизмов воздействия на эти процессы.

Формирование взглядов человека происходит не только за счет собственных убеждений и анализа информации, но и с учетом мнений группы, друзей и т. д. Авторами предпринята попытка адаптировать теорию динамического социального влияния Латане [3, 4] для того,

чтобы вычислить уровень влияния окружающих людей на мнение конкретного человека. Для проведения экспериментов в рамках данного исследования был создан программный комплекс для извлечения, обработки и анализа пользовательских данных. В частности, в системе имеется модуль, позволяющий извлекать данные из социальной сети vkontakte. Кроме того, этот модуль имеет возможность функционального расширения практически на любую социальную сеть, в зависимости от предоставляемого API.

1. Количественные характеристики, отношения и множества, вычисляемые на основе данных, получаемых из социальных сетей

В процессе анализа социальных сетей целесообразно рассматривать ряд числовых и нечисловых характеристик, отношений и множеств, естественным образом связанных с пользователями сети и сообщениями, циркулирующими в ней. Важным является то обстоятельство, что все они конструктивны, т. е. могут быть вычислены или построены при помощи соответствующих алгоритмов, конечно, при наличии программного обеспечения, позволяющего «выкачивать» необходимую информацию из сети.

Обозначим p – сообщение («пост») социальной сети, u – пользователь сети, который может принадлежать сообществу, т. е. быть на него подписанным. Он также может создавать и пересылать сообщения.

Числовые одноместные характеристики:

$Followers_Count(u)$ – количество людей, которые читают сообщения данного пользователя (т. е. подписчиков этого пользователя);

$Friends_Count(u)$ – количество друзей у данного пользователя (пользователь сам заносит некоторых людей в список друзей);

$Timeline_Count(u)$ – количество сообщений («постов»), созданных данным пользователем;

$Retweets(p)$ – количество пересылок данного сообщения;

$Likes(P)$ – количество «лайков» данного сообщения, т. е. отметок, что данное сообщение нравится;

Нечисловые характеристики:

$Real_name(u)$ – «настоящее» имя пользователя, если соответствующая позиция заполнена (это нечисловая характеристика);

$Bdate(u)$ – дата рождения пользователя;

$City(u)$ – город проживания пользователя;

$Hometown(u)$ – родной город пользователя или нет;

$Relation(u)$ дает информацию о семейном положении пользователя:

1 – не женат / не замужем; 2 – есть друг / есть подруга; 3 – помолвлен / помолвлена; 4 – женат / замужем; 5 – все сложно; 6 – в активном поиске; 7 – влюблен / влюблена.

$Career(u)$ – место работы, должность;

$University(u)$ – место учебы (вуз, колледж и т. д.), факультет;

$School(u)$ – школа, в которой учился пользователь.

Если указан год выпуска из учебного заведения, то иногда условно можно считать его не числовой характеристикой. Например, если для пользователей осуществляется сравнение годов только на совпадение, и факт совпадения или не совпадения отражается в виде ± 1 или в виде 0,1. Обычно такой тип оцифровывания данных называется индикаторным.

Множества:

$Followers(u)$ – подписчики данного пользователя;

$Friends(u)$ – друзья данного пользователя;

$Mentions(u)$ – имена пользователей, упоминаемые в сообщениях данного пользователя;

$Hashtags(u)$ – хэштеги, которые встречаются в сообщениях данного пользователя;

$Urls(u)$ – внешние ссылки, которые встречаются в сообщениях данного пользователя.

Числовые характеристики, ассоциированные с множествами:

$Count_Mentions_u(v)$ – количество упоминаний пользователя v пользователем u ;

$Count_Hashtags_u(v)$ – количество употреблений хештега v пользователем u ;

$Count_Urls_u(v)$ – количество упоминаний внешней ссылки v пользователем u .

$Count_Retweets_u(u_1)$ – количество сообщений пересланных пользователем u , полученных от пользователя u_1 .

$Count_CommonFriends_u(v)$ – количество общих друзей пользователей v и u ;

Чем больше количество подписчиков, тем более популярнее пользователь. Таким образом, если проводить рассуждения в терминах теории социального влияния Латане [3,4], то функцию $Followers_Count(u)$ можно считать «силой влияния» пользователя u .

2. Теория социального влияния Латане и ее модификация

Далее, рассмотрим, каким образом можно адаптировать теорию динамического социального влияния, предложенную Латане на случай социальных сетей.

Латане подчеркивал важность трех атрибутов отношений между получателем информации и источником: сила – социальный статус и сила доверия; расстояние – физическое или психологическое; текущее количество – число источников, влияющих на получателя.

Согласно теории динамического социального влияния, уровень влияния, испытываемого индивидуумом, может быть выражен следующей формулой

$$I_i = -S_i\beta - \sum_{j=1, j \neq i}^N \frac{S_j O_j O_i}{d_{i,j}^\alpha},$$

где: I_i – количество социального давления, направленного на индивидуума; O_i – мнение данного i -го индивидуума (± 1) по отношению к данному вопросу, значение $+1$ соответствует поддержке и -1 соответствует сопротивлению предложению; S_i – сила социального влияния ($S_i \geq 0$); β – сопротивление к изменениям; d_{ij} – расстояние между индивидуумами i и j ; α – степень ослабления расстояния; N – общее число взаимодействующих агентов.

Значение постоянной β обычно принимается равным 2 в соответствии с величиной, использованной в исследованиях Латане. Большее значение этой постоянной означает, что для изменения мнения требуется большее давление, меньшее значение соответствует меньшему усилию. Значение постоянной α также обычно принимается равным 2. Большие значения α означают, что с ростом расстояния между источником и получателем требуется много большая величина давления.

Величина d_{ij} определяется свойствами пары индивидуумов, она может рассматриваться как показатель легкости общения (передачи информации). При задании данной величины могут учитываться возрастные, национальные, конфессиональные и другие различия. Формула для вычисления d_{ij} может включать в себя физическое расстояние. Например, между населенными пунктами, в которых находятся индивиды. Обычно учитывается факт, что легкость коммуникации подчиняется закону об обратной квадратичной зависимости от физического расстояния [4]. Для компьютерных сетей возможны различные подходы, в том числе такие, когда физическое расстояние не принимается во внимание.

Для анализа социальных сетей можно предложить модификацию формулы Латане в следующем виде

$$I_u = -\beta \cdot \sum_{i=1}^N Followers_Count(u_i) - \sum_{i=1}^N \sum_{\substack{j=2 \\ i>j}}^N \frac{Followers_Count(u_j) O_j O_i}{d^\alpha(u_i, u_j)}.$$

В этой формуле силой влияния считается количество подписчиков пользователя. Чем больше количество подписчиков, тем популярнее пользователь. Здесь $d(u_i, u_j)$ – расстояние от пользователя u_i до пользователя u_j , которое определяем по несовпадениям анкетных данных.

В нашей работе в качестве расстояния используется расстояние Хэмминга – с каждым пользователем ассоциирован вектор нечисловых характеристик. Расстояние Хэмминга равно количеству позиций, на которых значения характеристик различаются. Величина O_i равна ± 1 в зависимости от того, состоит i -й индивидуум в рассматриваемом сообществе или нет.

В работе [4] речь идет о компании вакцинации полиомиелита, которая основывалась на сообщениях (через радио, печать, телевидение, кино и плакаты), оказывающих воздействие на общество, причем телевидение и радио играло наиболее важную роль в городских областях. В нашем случае отдаленным аналогом СМИ можно считать силу влияния хэштега и силу влияния ссылки.

Внешнее влияние, например влияние СМИ, также может быть учтено [4], если в основную формулу Латане добавить дополнительный член $O_i O_M S_{Mi}$, где S_{Mi} – сила влияния внешних источников на i -й индивидуум, $S_{Mi} > 0$, O_M – мнение внешнего источника. Учитывая влияние масс медиа, получаем итоговую формулу

$$I_i = -S_i \beta - O_i O_M S_{Mi} - \sum_{j=1, j \neq i}^N \frac{S_j O_j O_i}{d_{ij}^\alpha}.$$

Обычно внешний источник также моделируется как агент, но «вне окружающей среды» и с расстоянием 1 до каждого индивидуума ввиду своей «вездесущей» природы. Величина S_{Mi} меняется в зависимости от индивидуума, так как каждый человек испытывает различное давление СМИ, эта величина аналогична иногда рассматриваемой «величине доверия» индивидуума к сообщениям, получаемым из внешних источников.

Для социальных сетей аналогом СМИ может являться сила влияния хэштега, сила влияния ссылки, либо сила влияния рекламы, имеющейся в сообществе пользователей.

Соответственно получаем формулу

$$I_u = -\beta \cdot \sum_{i=1}^N Count_Mentions_u(u_i) - \sum_{i=1}^{|Hashtags(u)|} \sum_{\substack{j=2 \\ i>j}}^{|Hashtags(u)|} Hashtags_u(h_i, h_j) - \sum_{i=1}^N \sum_{\substack{j=2 \\ i>j}}^N \frac{Mentions_u(u_i, u_j)}{\rho^\alpha(u_i, u_j)},$$

в которой учитываются все пользователи, упоминаемые u , и все хэштеги.

Аналогично получаем

$$I_u = -\beta \cdot \sum_{i=1}^N Count_Mentions_u(u_i) - \sum_{i=1}^{|Urls(u)|} \sum_{\substack{j=2 \\ i>j}}^{|Urls(u)|} Urls_u(url_i, url_j) - \sum_{i=1}^N \sum_{\substack{j=2 \\ i>j}}^N \frac{Mentions_u(u_i, u_j)}{\rho^\alpha(u_i, u_j)}.$$

В случае, когда учитывается реклама, направленная на сообщество, получаем формулу

$$I_u = -\beta \cdot \sum_{i=1}^N Followers_Count(u_i) - \sum_{i=1}^N Followers_Count(s) O_i O_s - \sum_{i=1}^N \sum_{\substack{j=2 \\ i>j}}^N \frac{Followers_Count(u_j) O_j O_i}{d^\alpha(u_i, u_j)}.$$

Здесь количество подписчиков сообщества $Followers_Count(s)$ считаем как силу влияния сообщества на данного пользователя. Если есть реклама на сообщество, то O_s принимает значение $+1$, иначе -1 .

3. Программная реализация и эксперименты

В процессе исследований был разработан программный комплекс, содержащий модули извлечения информации из социальных сетей, обработки, анализа и визуализации данных. Все модули реализованы на языке Python. Модуль извлечения данных имеет возможность извлекать данные, в первую очередь, из крупнейших социальных сетей: Twitter и vkontakte. Для доступа к каждой из них используется интерфейс прикладного программирования (API). Извлеченные данные пользователей можно разделить на три категории: 1) собственно пользовательские данные, такие как имя, ник, время регистрации; 2) сообщения пользователей; 3) связи между пользователями. Ниже в таблице представлены результаты тестирования. В последней колонке приведены уровни влияния подписчиков (друзей) рассматриваемого индивидуума.

Результаты тестирования

Друзья Moldasheva	Количество друзей	Количество подписчиков	Состоит ли в группе	Расстояние Хэмминга	Уровень влияния
Asselya Moldasheva	439	828	1	7	-118,268
Дильшат Delsh Аширов	378	67	-1	8	8,375
Бейбіт Тузелбаев	501	128	1	7	-18,2857
Сергей undertaker Коробицин	133	34	1	7	-4,85714
Балым Каратаева	267	186	-1	7	26,57143
DauletSabyr	65	65	1	7	-9,28571

Список литературы

1. Charu C. Aggarwal Social network data analytics. 2011. 520 p.
2. Батура Т. В. Методы анализа компьютерных социальных сетей // Вестн. НГУ. Серия: Информационные технологии. Новосибирск, 2012. Т. 10, Вып. 4. С. 13–28.
3. Nowak A., Szamrej J., Latané B. From private attitude to public opinion: a dynamic theory of social impact // Psychological Review. 1990. Vol. 97. P. 362–376.
4. Wragg T. Modeling the Effects of Information Campaigns Using Agent-Based Simulation. // Prep.: Command and Control Division, Defense Science and Technology Organization, Australian Government, DSTO-TR-1853. 2006. 61 p.

UDK 519.68; 681.513.7; 612.8.001.57

LATANE'S THEORY OF SOCIAL IMPACT AND ITS APPLICATION FOR THE ANALYSIS OF ENVIRONMENT OF USER IN A SOCIAL NETWORK

A. J. Akhmetova

Ph.D. student, L. N. Gumilyov Eurasian National University,
010000, Astana, Kazakhstan, Satbayev str., 2, e-mail: akhmetova_azh@mail.ru

L. L. La

Ph.D., assistant professor, L. N. Gumilyov Eurasian National University,
010000, Astana, Kazakhstan, Satbayev str., 2, e-mail: lira_la@hotmail.com

F. A. Murzin

Ph.D., deputy director, A. P. Ershov Institute of Informatics Systems SB RAS,
630090, Novosibirsk, Russia, Acad. Lavrentjev pr., 6, e-mail: murzin@iis.nsk.su

Abstract. The article is devoted to the problems of the analysis of computer social networks. Modifications of Latane's dynamic social impact theory to social networks are proposed. The formulas describing the amount of social pressure directed at an individual for different situations are modified. The article briefly reports on the developed software package that allows us to extract information from social networks, carry out processing, analyzing and visualizing data. Testing was performed on data obtained from the social network vkontakte. The investigations were carried out under the financial support of the Ministry of Education and Science of the Republic of Kazakhstan (grant 2018-2020 MES RK № AP05133550) and the Integration Project of the SB RAS (AAAA-A18-118022190008-8).

Keywords: analysis of social networks, Internet, vkontakte, data analysis, Lathane theory

References

1. *Charu C. Aggarwal* Social network data analytics. 2011. 520 p.
2. *Batura T. V.* Metody analiza komputernykh socialnykh setei [Methods of data analysis from social networks] // Vest. NGU = Vestnik Novosibirsk State University. Series: Information Technologies. Novosibirsk, 2012. Vol. 10, Iss. 4. P. 13–28. (in Russian)
3. *Nowak A., Szamrej J., Latané B.* From private attitude to public opinion: a dynamic theory of social impact // Psychological Review. 1990. Vol. 97. P. 362–376.
4. *Wragg T.* Modeling the Effects of Information Campaigns Using Agent-Based Simulation. // Prep.: Command and Control Division, Defense Science and Technology Organization, Australian Government, DSTO-TR-1853. 2006. 61 p.

Как устроены современные высокопроизводительные биржевые инфраструктуры

Д. Е. Бабурин

начальник отдела разработки ПО QUIK, ARQA Technologies,
630007, Новосибирск, Россия, ул. Коммунистическая, 2, e-mail: dbabur@arqatech.com

Аннотация. В работе представлен обзор технических решений, применяемых в современных биржевых инфраструктурах. Объясняется, как устроены биржи, какие технические требования предъявляются к торговым системам, какие программно-аппаратные решения и протоколы передачи данных используются. Приведено описание технологий, работающих на LSE (London Stock Exchange) и МБ (Московская Биржа).

Ключевые слова: биржа, обработка данных, FIX, FAST, FPGA.

Введение

Современные биржи, такие как Московская Биржа, London Stock Exchange, NYSE, Nasdaq и т. д., организованы в виде узкоспециализированных программно-аппаратных комплексов, предназначенных для проведения торгов теми или иными активами. К таким комплексам применяются строгие требования в части надежности и скорости работы. Целью данной статьи является рассмотрение биржевой инфраструктуры как средства, за счет которого достигается выполнение этих требований.

1. Структура биржи

Каждая биржа должна иметь в своем составе:

- 1) торговую систему, в которой накапливаются заявки на покупку и продажу и происходит их сведение при совпадении цены, а также производятся регистрация сделок и запись в соответствующие регистры учета;
- 2) расчетную или клиринговую палату – систему, которая ведет учет денежных средств участников торгов, осуществляет поставку денег по каждой сделке продавцам бумаг и списывает денежные средства со счетов покупателей, а также осуществляет внешние и внутренние денежные переводы;
- 3) депозитарный центр, который аналогично расчетной палате ведет учет ценных бумаг участников торгов, а также осуществляет поставку бумаг на счета покупателей, списывает бумаги со счетов продавцов и осуществляет клиринг ценных бумаг по результатам торгов в уполномоченных депозитариях.

Схематично торговая система состоит из следующих компонентов: шлюз, мэтчинговое ядро и сервис распространения рыночной информации.

Шлюз (на рисунке – Gateway) – это интерфейсная компонента, которая осуществляет подключение пользователей и прием заявок от них. Технологически шлюз обычно представляет собой распределенную ферму, позволяющую горизонтально масштабировать нагрузку от клиентских подключений. Шлюз взаимодействует с ядром торговой системы через проприетарный протокол или через шину данных и передает клиентское поручение в мэтчинговое ядро.

В мэтчинговом ядре (на рисунке – Trading Engine) происходит сбор заявок от всех подключений и выстраивание их в общую очередь. При приеме очередной заявки ядро проверяет ее на корректность, а затем выполняет поиск встречного предложения среди ранее выставленных заявок. Если находятся заявки с нужными параметрами, то между участниками

формируется сделка. Неисполненная часть заявки помещается в очередь, в которой будет ожидать встречных предложений, если они поступят позднее, или же своего снятия.

Информация о совершенной сделке и движении активов отправляется в клиринговую и депозитарную системы (на рисунке – Clearing и Depository соответственно).

Отдельным процессом является сервис распространения рыночной информации (на рисунке – Market Data). При приеме и мэтчинге заявок торговое ядро генерирует некоторое количество общезначимых рыночных показателей, которые характеризуют торговую активность или состояние торгов. За распространение такой обезличенной рыночной информации обычно отвечает отдельный сервис – Market Data Dissemination Service.



Схема устройства биржи

2. Требования к торговым системам

Выделим основные причины возникновения повышенных требований к производительности торговых систем.

Во-первых, количество данных, которыми оперируют торговые системы, является огромным и постоянно растет. Например, на торгах обычной интенсивности на американской площадке Nasdaq выставляются тысячи заявок в секунду, а количество разнообразных изменений в обезличенной рыночной информации измеряется десятками тысяч записей в секунду.

Во-вторых, с увеличением количества данных растут и аппетиты пользователей в части скорости получения данных, подстегивая конкуренцию между производителями решений для биржевой торговли.

Третий аспект, который хотелось бы здесь отметить, – это крайне неравномерный характер пользовательской активности. При появлении неожиданных новостей мгновенная интенсивность торгов может увеличиваться на несколько порядков, что в идеале не только не должно сказываться на доступности сервиса, но и должно проходить вообще незамеченным для пользователей системы.

Иллюстрацией перечисленных требований могут быть следующие вполне типичные показатели: 99,99 % надежности, 200 μ s время ответа торговой системы на действия пользователя, обработка 50 000 операций в секунду.

Возможно, для современных высокопроизводительных веб-серверов эти требования не кажутся высокими, но необходимо отметить некоторые особенности.

Так, биржевое ПО фактически является real-time системой, в которой недопустима существенная дисперсия скорости реакции на разные события. Своевременный отклик должен быть гарантированным, а пользовательские запросы должны обрабатываться в строгом порядке их поступления без возможности перестановки местами.

Второй аспект, отличающий биржи, состоит в том, что раздача биржевой информации выполняется не единственному клиенту, который ее запросил (как это часто бывает в веб-сервере), а всем подключенным системам. При этом необходимо, чтобы все системы получили данные одновременно, а не в порядке какой-либо очереди.

Третье отличие биржевой инфраструктуры от типичного веб-сервера состоит в том, что она не может быть децентрализована и разнесена по сотням платформ в облачном сервисе. Обработка всех операций в ядре системы обязана быть синхронной и последовательной.

3. Технические решения

Основные вызовы, стоящие перед производителями торговых систем: скорость распространения рыночной информации, скорость мэтчинга, отказоустойчивость и джиттер. Далее рассмотрим программно-аппаратные решения, которые используются для борьбы с этими вызовами.

3.1. InfiniBand

Протокол Ethernet, придуманный более сорока лет назад, является крайне неэффективным с точки зрения затрат на передачу информации. Именно поэтому в 1999 г. индустрией был предложен новый перспективный стандарт InfiniBand.

При проектировании стандарта InfiniBand были заложены следующие приоритеты: иерархическая приоритизация трафика, низкая латентность, масштабируемость, возможность резервирования, а также выбор любой необходимой скорости без смены самого протокола.

Например, на самом современном сетевом Ethernet-оборудовании сейчас можно получить в лучшем случае 3–5 μ s накладных расходов на обработку логики TCP стека и доставку данных в приложение. В то же время InfiniBand даже на базовом оборудовании позволяет добиться накладных расходов всего в 1,5–2 μ s. Причем в стандарте изначально была заложена прямая передача прикладных данных с сетевого уровня на уровень приложения без участия операционной системы.

Принимая во внимание приведенные соотношения, становится понятно, почему современные биржи стараются переводить свои системы на InfiniBand как минимум внутри своего периметра.

3.2. FPGA

ПЛИС (программируемая логическая интегральная схема) представляет собой электронный компонент, используемый для создания цифровых интегральных схем. В отличие от обычных цифровых микросхем логика работы ПЛИС не определяется при изготовлении, а задается посредством программирования. Для программирования используются программаторы и отладочные среды, позволяющие задать желаемую структуру цифрового устройства в виде принципиальной электрической схемы.

С одной стороны, технология ПЛИС работает существенно быстрее обычных микроконтроллеров, которые реализуют алгоритмы на программном уровне, с другой – является в равной степени универсальной за счет возможности перепрограммирования.

FPGA (field-programmable gate array) – одна из разновидностей ПЛИС, которая содержит блоки умножения-суммирования, а также логические элементы и их блоки коммутации. Уже достаточно давно эта технология используется для создания ускорителей для обычных процессоров и в суперкомпьютерах. В последние несколько лет FPGA активно внедряются также в трейдинговые и биржевые инфраструктуры.

На биржах чаще всего используют FPGA для задачи распространения рыночной информации, когда необходимо получить событие из ядра системы и сформировать

исходящее сообщение. При этом внутри FPGA реализуется не только прикладная логика, но и полноценный TCP или UDP-стек.

Однако широкому распространению FPGA мешают следующие факторы: уникальность микросхем, недостаток вентиляей, необходимость оптимизации кода по количеству используемых команд, написание кода на низкоуровневом языке, отсутствие опытных технических специалистов.

3.3. RTP

При измерении времени с высокой точностью одной из ключевых задач становится точная синхронизация времени и маркирование данных временными метками («таймстемпами») с высокой точностью. Поскольку стандартный протокол синхронизации времени NTP не обеспечивает нужной точности, в биржевых инфраструктурах применяют RTP-синхронизацию.

Основное отличие этого протокола состоит в том, что для его реализации требуется аппаратная поддержка. Устройство с поддержкой протокола RTP проставляет точную метку времени каждому входящему сообщению на соответствующем порте еще до обработки пакета. Так как проставление метки времени осуществляется на аппаратном уровне, программные задержки при обработке сообщений перестают иметь значение с точки зрения задачи синхронизации времени.

За счет проставления точных меток времени получения и отправки сообщений с портов устройств измеряется время распространения сообщений по сети. Сообщение, поступающее на конечное устройство, содержит информацию о времени его отправки с ведущих часов и о времени, потребовавшемся для поступления на ведомое устройство. Затем эти данные используются для вычисления текущего времени с погрешностью менее или равной 1 мс.

4. Протоколы

Технологически задача приема торговых поручений выглядит существенно проще, чем широковещательная рассылка рыночной информации, так как темп приема поручений всегда в несколько раз меньше и каждое сообщение нужно вычитать / обработать один раз, а не рассылать его всем подключенным пользователям. Поэтому некоторое время назад на большинстве бирж произошло разделение прикладных протоколов, используемых для приема торговых поручений и раздачи рыночной информации.

4.1. FIX

Для приема торговых поручений чаще всего используется индустриальный стандарт FIX. Это текстовый протокол с парами тег-значение, который был придуман еще в прошлом веке и постепенно стал использоваться во всей индустрии для передачи торговых приказов. Приведем пример типичного FIX-сообщения: «8=FIX.4.2 | 9=178 | 35=D | 34=123 | 49=BRK | 56=PHLX | 52=20190423-05:30:00.000 | 11=C990 | 55=MSFT | 167=FUT | 54=1 | 38=15 | 40=2 | 44=15 | 59=0 | 10=128 |».

Сообщение читается следующим образом: сообщение в FIX-формате 4.2 (8=FIX.4.2), длина сообщения 178 байт (9=178), номер сообщения 123 (34=123), тип сообщения «новая заявка» (35=D), от брокера BRK (49=BRK) на филадельфийскую биржу (56=PHLX), отправлено (тег [52]), номер заявки C990 в торговой системе брокера (11=C990) на покупку (54=1) фьючерсов (167=FUT) на акции компании Microsoft (55=MSFT) в объеме 15 лотов (38=15) по лимитированной (40=2) цене 15 \$ (44=15), заявка истекает в конце торгового дня (59=0). Тег [10] содержит контрольную сумму.

Протокол является протоколом сессионного уровня, работающим поверх TCP, и, помимо бизнес-сообщений, содержит управляющие сообщения для установки / закрытия соединения,

а также восстановления пропущенных сообщений. Протокол расширяется за счет возможности использования кастомных тегов и передачи в них произвольной информации.

Программная реализация протокола FIX чаще всего представляет собой некоторый настраиваемый «движок», позволяющий описать структуру используемых FIX-сообщений на метаязыке.

4.2. UDP

В последние годы в рамках сервисов для раздачи биржевой информации биржи стали уходить от TCP-рассылки и переходить на UDP. Это связано с тем, что система реального времени не может ждать окончания рассылки задержавшихся пакетов, и предпочтительным является сброс всех пакетов даже без гарантии их доставки.

Но поскольку принимающие системы не всегда готовы пропустить некоторые биржевые данные, приходится строить сложные системы, состоящие из резервных UDP-каналов и recovery-сервисов на базе TCP.

Как кодируется и пакуется информация, рассылаемая по UDP? Ранее достаточно часто можно было встретить подход, при котором данные рассылались в проприетарном формате, а на стороне клиента нужно было установить библиотеку, реализующую этот формат и предоставляющую API для доступа к данным. Однако сейчас индустрия уже практически избавилась от таких решений. Все протоколы являются открытыми, что позволяет клиентам самостоятельно реализовывать их разбор на своей стороне.

4.3. FAST

Основным протоколом для распространения рыночной информации является протокол FAST (FIX Adapted for Streaming) и его вариации. Это бинарный вариант протокола FIX, адаптированный для поточной передачи по сети.

Протокол FAST позволяет избавиться от избыточности с помощью шаблона, который описывает структуру сообщения целиком. Этот метод называется «неявным тегированием», поскольку FIX-теги в передаваемых данных только «подразумеваются». Таким образом, синтаксис Tag = Value меняется следующим образом:

- шаблон описывает структурированный набор полей с их операторами;
- последовательность полей в сообщении соответствует последовательности тегов в шаблоне;
- в сообщении передаются только изменения данных.

5. Примеры биржевых инфраструктур

Далее рассмотрим конкретные решения, которые применяются в настоящий момент в реальных биржевых инфраструктурах.

5.1. London Stock Exchange

LSE – одна из крупнейших и старейших бирж Европы и один из наиболее известных мировых рынков ценных бумаг. До 2010 г. система работала под Microsoft .Net, а затем на LSE завершили самый сложный многолетний проект по миграции на новую торговую платформу Millennium, которая была приобретена у индийской софтверной компании. Основной технологический стек Millennium – Linux, Sun Solaris Unix, БД Oracle.

Этот переход позволил сократить среднюю задержку обработки транзакций и устранить джиттер. При средней задержке в 126 μ s 99 % транзакций укладываются в 210 μ s, а 99,9 % – в 400 μ s.

После решения проблем с задержкой регистрации транзакций на LSE запустили проект по ускорению распространения рыночной информации и в 2013 г. отказались от связки протоколов FIX-FAST, перейдя на собственный проприетарный протокол GTP (Group Ticker Plant). В 2016 г. на LSE завершили проект по переходу на аппаратную поддержку протокола GTP внутри чипов FPGA. Потратив пять лет и большое количество средств, LSE получила внутри своего периметра ферму FPGA-устройств, способных закодировать поступающие из ядра торговые данные и отправить их «наружу» за 5 μ s.

5.2. Московская биржа

Московская биржа – крупнейшая российская площадка, имеющая сразу две торговые системы: на платформе Rebus ведется торговля акциями, облигациями и валютой, а на платформе Spectra – срочными контрактами (фьючерсами и опционами).

Основное отличие данных торговых систем от мировых аналогов в том, что они имеют отдельный встроенный блок для проверки каждой транзакции на обеспеченность (на наличие необходимых ресурсов для совершения операции на счетах участника). Поэтому скорость проверки транзакций на Московской бирже чуть хуже, чем у мировых лидеров. Среднее время обработки транзакции в системе Spectra – 200 μ s, в системе Rebus – около 300 μ s.

С точки зрения ключевых технологий можно выделить следующие. В 2011 г. платформа Rebus была переведена на real-time операционную систему Red Hat MRG, что позволило устранить джиттер. Spectra работает на Windows-ядре, но при этом много внимания уделяется задачам ручного распределения потоков по ядрам. Rebus работает на InfiniBand, который обеспечивает задержку в 2 μ s при передаче данных между хостами, а Spectra продолжает использовать Ethernet. При этом количество аппаратных платформ в одной инсталляции торговой системы – 25 для Rebus и 10 для Spectra.

Общие элементы. В качестве шины данных в обеих системах применяется IBM LLM, язык разработки C++. Основные протоколы приема транзакций и раздачи рыночной информации являются проприетарными и не мультикастными.

5.3. QUIK

В завершение обзора я позволю себе упомянуть биржевое решение, написанное внутри нашей компании, которое применяется на нескольких небольших публичных европейских площадках, а также для реализации внутреннего пула торгов у ряда российских профучастников.

Биржевое решение построено на базе системы интернет-трейдинга QUIK. QUIK – это программный комплекс, предоставляющий пользователям доступ к биржевым системам. При этом сама система уже обладает большинством функционала, присущего торговым системам:

- 1) внешние программные интерфейсы для подключения систем. Реализация FIX для приема транзакций и проприетарный протокол передачи рыночной информации;
- 2) система ведения позиций участников и проверки заявок на обеспеченность;
- 3) системы администрирования пользователей, мониторинга, отчетности и т. п.

Для создания торговой системы на базе QUIK мы лишь написали небольшой модуль, являющийся реализацией мэтчингового ядра. Это мэтчинговое ядро позволяет на стандартном Windows-сервере обрабатывать несколько тысяч транзакций в секунду с внутренним временем мэтчинга в несколько сотен микросекунд.

Технологически этого удалось достичь за счет того, что потоки исполнения привязаны к ядрам, используется неблокирующая синхронизация, система распространения информации отвязана от ядра, а также применяется отложенная запись.

Таким образом, у нас получилась относительно компактная и производительная реализация торговой системы.

Список литературы

1. InfiniBand. Wikipedia – The Free Encyclopeida. URL: <https://en.wikipedia.org/wiki/InfiniBand> (дата обращения: 30.04.2019).
2. *Leber Ch., Geib B., Litz H.* High Frequency Trading Acceleration Using FPGAs. FPL '11 Proceedings of the 2011 21st International Conference on Field Programmable Logic and Applications. 2011. P. 317–322.
3. FIX Trading Community. URL: <https://www.fixtrading.org/> (дата обращения: 30.04.2019).
4. London Stock Exchange. URL: <https://www.londonstockexchange.com/home/homepage.htm> (дата обращения: 30.04.2019).
5. Московская биржа. 2019. URL: <https://www.moex.com/> (дата обращения: 30.04.2019).

UDK 519.687.8

HOW DO MODERN HIGH-PERFORMANCE STOCK EXCHANGE INFRASTRUCTURES WORK

D. Baburin

Head of QUIK software development department, ARQA Technologies,
630007, Novosibirsk, Russia, Kommunisticheskaya st., 2, e-mail: dbabur@arqatech.com

Abstract. The paper presents an overview of the technical solutions used in modern stock exchange infrastructures. How do stock exchanges work, what technical requirements are imposed on trading systems, what software and hardware solutions and data transfer protocols are used. The article describes the technologies working on the LSE (London Stock Exchange) and MB (Moscow Exchange).

Keywords: Stock exchange, Data processing, FIX, FAST, FPGA

References

1. InfiniBand. Wikipedia – The Free Encyclopeida. URL: <https://en.wikipedia.org/wiki/InfiniBand>.
2. *Leber Ch., Geib B., Litz H.* High Frequency Trading Acceleration Using FPGAs. FPL '11 Proceedings of the 2011 21st International Conference on Field Programmable Logic and Applications. 2011. P. 317–322.
3. FIX Trading Community. URL: <https://www.fixtrading.org/>.
4. London Stock Exchange. URL: <https://www.londonstockexchange.com/home/homepage.htm>.
5. Moscow Exchange. 2019. URL: <https://www.moex.com/>. (in Russia)

Система автореферирования научно-технических текстов

Т. В. Батура

канд. физ.-мат. наук, доц., ст. науч. сотр.,
Институт систем информатики им. А. П. Ершова СО РАН,
630090, Новосибирск, пр. Акад. Лаврентьева, 6,
Новосибирский государственный университет,
630090, Новосибирск, Россия, ул. Пирогова, 2, e-mail: tatiana.v.batura@gmail.com

А. М. Бакиева

соискатель, Институт систем информатики им. А. П. Ершова СО РАН,
630090, Новосибирск, Россия, пр. Акад. Лаврентьева, 6, e-mail: m_aigerim0707@mail.ru

Аннотация. В работе описан новый метод автоматического реферирования текстов. На основе предложенного метода создана система, позволяющая получать краткие аннотации научно-технических текстов и определять их темы. Предлагаемый метод формирует аннотацию на основе наиболее значимых предложений исходного документа. Значимость предложений частично определяется в процессе риторического анализа, который выполняется с помощью дискурсивных маркеров. Также учитываются ключевые слова, многословные термины и некоторые специальные слова, которые часто встречаются в научно-технических текстах. Для извлечения ключевых слов и определения тем текста применялась аддитивная регуляризация тематических моделей (АРТМ).

Ключевые слова: автоматическое реферирование, теория риторических структур, дискурсивные маркеры, аддитивная регуляризация, тематические модели

Введение

Ввиду стремительного увеличения объемов текстовой информации в Интернете активные исследования в области компьютерной лингвистики сохраняют свою актуальность. Разработка алгоритмов и создание систем автоматического реферирования, поиска и извлечения информации, классификации и кластеризации текстовых документов по-прежнему являются сложными задачами.

Подход, основанный на применении дискурсивного анализа, используется довольно широко для решения различных задач компьютерной лингвистики. Подробный обзор литературы, представленный в работе [1], показывает, что в большинстве случаев дискурсивный анализ способен улучшить качество автоматических систем на 4–44 % в зависимости от конкретной задачи.

Система автореферирования научных статей, основанная на дискурсивном анализе, описана в [2]. В ней определены семь риторических категорий. Автор работы [3] применил теорию риторических структур для создания графического представления документа. На основе структурного анализа текста вычисляются веса предложений, из которых в итоге получается краткая аннотация. В работе [4] обсуждается создание реферата, содержащего информацию не только из одного конкретного документа, но и дополнительные знания из других, похожих на него по тематике документов.

В нашей работе описывается подход, позволяющий получать краткие аннотации научно-технических текстов и определять их темы. Предлагаемый метод формирует аннотацию на основе наиболее значимых предложений исходного документа. Значимость предложений частично определяется в процессе риторического анализа.

1. Риторический анализ и описание преобразования текста

Теория риторических структур – одна из наиболее широко используемых теорий организации текстов [5]. Согласно ей, изначально текст делится на неперекрывающиеся фрагменты, а именно на элементарные дискурсивные единицы (ЭДЕ). Последовательные ЭДЕ связаны между собой риторическими отношениями.

В теории риторических структур определяется два типа ЭДЕ. Один из них, называемый ядром, считается наиболее важной частью высказывания, другой, называемый сателлитом, поясняет ядро и считается вторичным. Ядро содержит основную информацию, тогда как сателлит содержит дополнительную информацию о ядре. Сателлит часто непонятен без ядра. Между тем, выражения, в которых сателлит удален, могут быть поняты лишь в некоторой степени. Рассмотрим следующий пример.

В предлагаемом подходе риторический анализ используется на этапе построения квазиреферата. Под квазирефератом понимается перечень наиболее значимых предложений текста. Упрощенно этот этап можно описать следующим образом. Сначала необходимо найти в тексте ядерные ЭДЕ. Далее следует преобразовать высказывания, содержащие эти ЭДЕ, чтобы получился сокращенный текст, являющийся промежуточным между исходным текстом и готовой аннотацией. В зависимости от разных маркеров и дискурсивных отношений эти преобразования будут разными. Для формального описания действий, выполняемых системой, было принято решение использовать логику предикатов первого и второго порядков.

Маркеры (дискурсивные маркеры) – это слова или фразы, которые не имеют реального лексического значения, но вместо этого обладают важной функцией формировать разговорную структуру, передавая намерения говорящих при разговоре. Примеры маркеров и действий, связанных с ними, приведены в табл. 1.

Таблица 1

Действия для маркеров

№	Риторические отношения	Маркеры	Действия
1.	Cause-Effect	Поэтому	Delete_save
2.	Contrast	Однако	Save_delete
3.	Elaboration	Кроме того	Save_delete
4.	Elaboration	Например	Save_delete
5.	Evidence	Таким образом	Delete_save
6.	Restatement	Другими словами	Save_delete

Delete_save: Это действие удаляет предстоящее предложение и сохраняет предложение с заданным маркером.

Save_delete: Это действие сохраняет предстоящее предложение и удаляет предложение с заданным маркером.

2. Общее описание системы

В нашем понимании задача реферирования состоит в том, чтобы найти преобразование текста T в реферат \tilde{T} , такое, что

$$\Phi: T \rightarrow \tilde{T}, |T| > |\tilde{T}| \approx 250 \text{ слов.}$$

Тогда алгоритм построения реферата можно записать в виде последовательных этапов.

1. Предобработка текста. На этапе предварительной обработки из исходного текста удаляются все изображения, таблицы, формулы, информация об авторах и библиографические ссылки.

2. Построение тематических моделей, извлечение ключевых слов и многословных терминов. Тематическое моделирование заключается в построении модели некоторой коллекции текстовых документов. В такой модели каждая тема представляется дискретным распределением вероятностей слов, а документы – дискретным распределением вероятностей тем. Для определения тем текстов мы применяли метод аддитивной регуляризации тематических моделей ARTM (Additive Regularization for Topic Modeling [6]). Причины выбора и результаты сравнения его с другими методами описаны в работе [7]. Для извлечения многословных терминов был использован алгоритм RAKE (Rapid Automatic Keyword Extraction [8]).

3. Риторический анализ и преобразование текста. На этом шаге обнаруживаются предложения, содержащие дискурсивные маркеры, и выполняются определенные преобразования с текстом, в результате чего формируется квазиреферат.

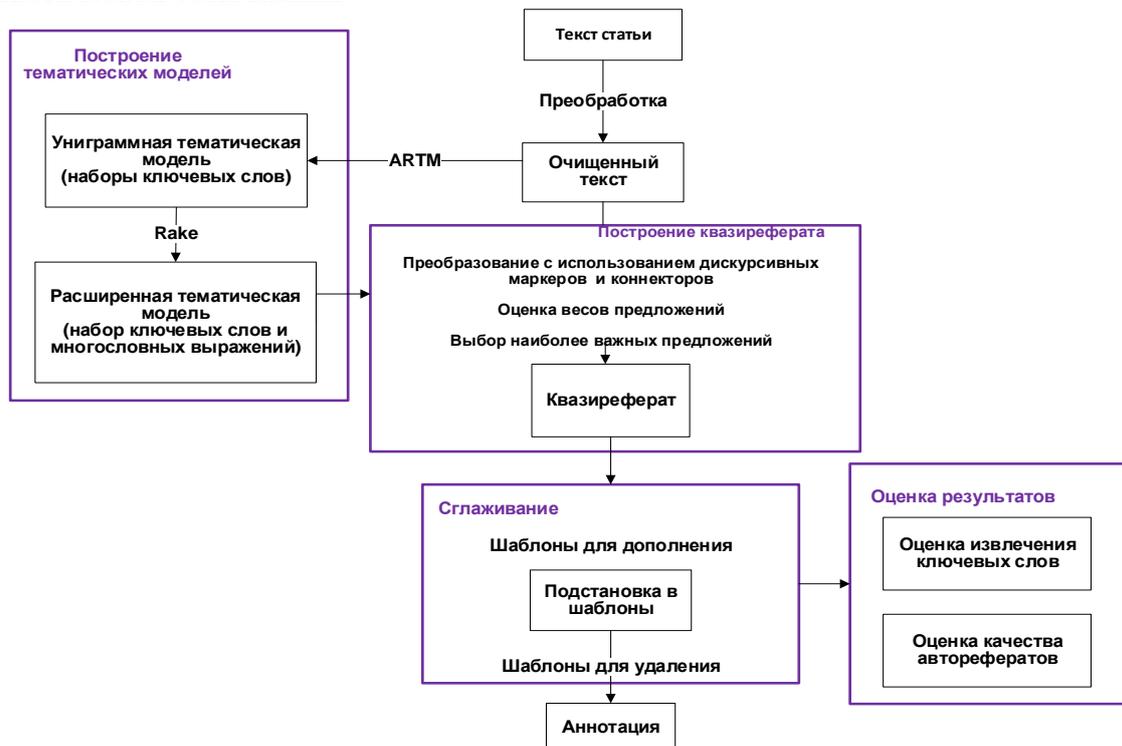
4. Оценка весов предложений. При вычислении веса каждого предложения квазиреферата учитывается наличие в этом предложении ключевых слов (или многословных терминов), дискурсивных маркеров, а также некоторых слов, которые характерны для научных текстов. В итоге вес каждого предложения вычисляется по следующей формуле:

$$SW = \frac{1}{L} \cdot \sum_{i=1}^L w_i + \frac{1}{M} \cdot \sum_{j=1}^M v_j + \frac{1}{N} \cdot \sum_{k=1}^N d_k, \text{ где}$$

$W = \{w_1, \dots, w_L\}$ веса ключевых слов и многословных выражений ($|W| = L$) определяются как частоты на этапе 2;

$V = \{v_1, \dots, v_M\}$ веса значимых глаголов и существительных, которые часто встречаются в научных текстах ($|V| = M$), определяются из лингвистической базы знаний;

$D = \{d_1, \dots, d_N\}$ веса дискурсивных маркеров ($|D| = N$) определяются из лингвистической базы знаний.



Блок-схема

5. Выбор наиболее важных предложений. Из полученного набора предложений (см. п. 3) для реферата отбираются только те предложения, вес которых (см. п. 4) превышает заданную пороговую величину β :

$$\tilde{T} = \bigcup_{k=1}^{N_1} \{S'_k : SW > \beta\},$$

где $\beta = 0,15$ является константой, которая определяется эмпирически; от нее зависит, насколько сильно будет сокращен текст.

6. Сглаживание – процедура преобразования текста, позволяющая получить связный текст из разрозненных фрагментов и при необходимости дополнительно сократить его.

В ходе данной работы была разработана система. На рисунке представлена блок-схема.

1. Оценка результатов

Автоматическая оценка осуществлялась при помощи точности, полноты и F-меры.

Таблица 2

Оценка результатов

Система	Метод	Точность, %	Полнота, %	F-мера, %
Marcu (1998)	Комбинация риторического анализа, структурного и позиционного методов	73,53	67,57	70,42
Trevgoda (2009)	Риторический анализ	67,03	64,81	66,03
Open Text Summarizer (2016)	Статистический метод	12	24,2	38,5
Scientific Text Summarizer (2018)	Предложен в данной работе	75,23	68,21	71,55

Преимуществом предложенного метода является довольно высокая точность и наличие небольшого количества данных для обучения. Недостатком является необходимость ручной настройки лингвистической базы знаний. Возможное улучшение предложенного в данной статье алгоритма, по нашему мнению, состоит в том, чтобы дополнить правила удаления менее важных предложений, увеличить количество шаблонов для сглаживания, расширить списки маркеров.

Заключение

В данной статье описан подход к автоматическому построению аннотаций научно-технических текстов на русском языке. Процесс состоит из шести шагов: предобработка, построение тематических моделей, риторический анализ, оценка весов предложений, выбор предложений и сглаживание. Проведенные эксперименты подтвердили эффективность предложенного метода. В дальнейшем планируется провести эксперименты с текстами из различных научных областей на других языках.

Исследование выполнено при финансовой поддержке РФФИ в рамках научного проекта № 19-07-01134.

Список литературы

1. *Ананьева М. И., Кобозева М. В.* Разработка корпуса текстов на русском языке с разметкой на основе теории риторических структур // Компьютерная лингвистика и интеллектуальные технологии: по материалам ежегодной Междунар. конф. «Диалог». 2016. URL: www.dialog-21.ru/media/3460/ananyeva.pdf.
2. *Teufel S., Moens M.* Summarizing scientific articles: experiments with relevance and rhetorical status // *Computational Linguistics*. 2002. Vol. 28(4). P. 409–445.
3. *Bosma W.* Query-Based Summarization using Rhetorical Structure Theory // 15th Meeting of CLIN. 2005. P. 29–44.
4. *Huspi S. H.* Improving Single Document Summarization in a Multi-Document Environment : PhD thesis. Melbourne : RMIT University, 2017. 190 p.
5. *Mann W., Thompson C.* Rhetorical structure theory: Toward a functional theory of text organization // *Text-Interdisciplinary Journal for the Study of Discourse*. 1988. Vol. 8. № 3. P. 243–281.
6. *Vorontsov K., Frei O., Apishev M., Romov P., Dudarenko M.* BigARTM: Open Source Library for Regularized Multimodal Topic Modeling of Large Collections // *International Conf. on Analysis of Images, Social Networks and Texts (AIST)*. Yekaterinburg, 2015. P. 370–384.
7. *Батура Т. В., Стрекалова С. Е.* Подход к построению расширенных тематических моделей текстов на русском языке // *Вест. НГУ. Серия: Информационные технологии*. 2018. Вып. 16. № 2. С. 5–18.
8. *Rose S., Engel D., Cramer N., Cowley W.* Automatic keyword extraction from individual documents // *Text Mining: Applications and Theory*. 2010. P. 3–20.

UDK 004.048:519.765

AUTOMATIC SUMMARIZATION SYSTEM OF SCIENTIFIC AND TECHNICAL TEXTS

T. V. Batura

Ph.D., senior researcher,

A. P. Ershov Institute of Informatics Systems, Russian Academy of Sciences, Siberian Branch
630090, Novosibirsk, Russia, Acad. Lavrentiev pr., 6,
Novosibirsk State University
6300906, Novosibirsk, Russia, Pirogova Str., 2, e-mail: tatiana.v.batura@gmail.com

A. M. Bakiyeva

external Ph.D. student A. P. Ershov Institute of Informatics Systems,
Russian Academy of Sciences, Siberian Branch

6300906, Novosibirsk, Russia, Acad. Lavrentiev pr., 6, e-mail: m_aigerim0707@mail.ru

Abstract. The paper describes a new method of automatic text summarization. Based on this method, a system has been created that makes it possible to obtain summaries of scientific and technical texts and to determine their topics. The proposed method allows to receive a summary based on important sentences of the original document. The importance of sentences is partially determined in the process of rhetorical analysis, which is performed using discourse markers. Keywords, multiword terms, and some scientific special terms that are often found in scientific and technical texts are also taken into account. We used additive regularization for topic modeling (ARTM) to extract keywords and discover the topics.

Keywords: automatic summarization, rhetorical structure theory, discourse markers, additive regularization, topic modeling

References

1. *Ananyeva M. I., Kobozeva M. V.* Development of the corpus of Russian texts with markup based on the Rhetorical Structure Theory // Proceedings of the International Conference “Dialog 2016”, Moscow, 2016, [Online]. URL: www.dialog-21.ru/media/3460/ananyeva.pdf.
2. *Teufel S., Moens M.* Summarizing scientific articles: experiments with relevance and rhetorical status // *Computational Linguistics*. 2002. Vol. 28(4). P. 409–445.
3. *Bosma W.* Query-Based Summarization using Rhetorical Structure Theory // 15th Meeting of CLIN. 2005. P. 29–44.
4. *Huspi S. H.* Improving Single Document Summarization in a Multi-Document Environment : PhD thesis. Melbourne : RMIT University, 2017. 190 p.
5. *Mann W., Thompson C.* Rhetorical structure theory: Toward a functional theory of text organization // *Text-Interdisciplinary Journal for the Study of Discourse*. 1988. Vol. 8. № 3. P. 243–281.
6. *Vorontsov K., Frei O., Apishev M., Romov P., Dudarenko M.* BigARTM: Open Source Library for Regularized Multimodal Topic Modeling of Large Collections // International Conf. on Analysis of Images, Social Networks and Texts (AIST). Yekaterinburg, 2015. P. 370–384.
7. *Batura T. V., Strekalova S. E.* An Approach to Building Extended Topic Models of Russian Texts. *Vest. NSU. Series: Information Technologies*, 2018. Vol. 16, №. 2. P. 5–18. DOI 10.25205/1818-7900-2018-16-2-5-18 (in Russ.).
8. *Rose S., Engel D., Cramer N., Cowley W.* Automatic keyword extraction from individual documents // *Text Mining: Applications and Theory*. 2010. P. 3–20.

Анализ временных и пространственных понятий, встречающихся в текстах на естественном языке

Т. В. Батура

канд. физ.-мат. наук, доц., ст. науч. сотр.,
Институт систем информатики им. А. П. Ершова СО РАН,
630090, Новосибирск, Россия, пр. Акад. Лаврентьева, 6, e-mail: tatiana.v.batura@gmail.com

Л. В. Ефимова

аспирант, Институт систем информатики им. А. П. Ершова СО РАН,
630090, Новосибирск, Россия, пр. Акад. Лаврентьева, 6, e-mail: efimova_l@ngs.ru

А. С. Еримбетова

аспирант, Новосибирский государственный университет,
630090, Новосибирск, Россия, ул. Пирогова, 1, e-mail: aigerian@mail.ru

А. Б. Касекеева

аспирант, Евразийский национальный университет им. Л. Н. Гумилева,
010000, Астана, Казахстан, ул. Сатбаева, 2, e-mail: aibike_7474@mail.ru

Ф. А. Мурзин

канд. физ.-мат. наук, зам. директора,
Институт систем информатики им. А. П. Ершова СО РАН,
630090, Новосибирск, Россия, пр. Акад. Лаврентьева, 6, e-mail: murzin@iis.nsk.su

Аннотация. В статье речь идет о создании базы знаний, содержащей информацию о временных и пространственных понятиях, встречающихся в текстах на естественном языке. База знаний содержит: наиболее важные понятия, относящиеся к времени и пространству, из толкового словаря С. И. Ожегова; перефразированные варианты ряда предложений; результаты анализа словарных статей (толкований) и примеров использования в художественной литературе соответствующих понятий из словаря С. И. Ожегова с помощью программных систем Link Grammar Parser и Dialing и т. д. Исследования выполнены при финансовой поддержке Министерства образования и науки Республики Казахстан (грант 2018-2020 MES RK No. AP05133550) и Интеграционного проекта СО РАН (АААА-А18-118022190008-8).

Ключевые слова: компьютерная лингвистика, семантика, временные и пространственные понятия, Link Grammar Parser, Dialing

Введение

Основная цель состоит в том, чтобы исследовать конструкции на естественном языке, содержащие временные и пространственные понятия, в семантическом плане.

Для этого создается база знаний, содержащая информацию о временных и пространственных понятиях. База знаний содержит: наиболее важные понятия, относящиеся к времени и пространству, из толкового словаря С. И. Ожегова; перефразированные варианты ряда предложений; результаты анализа словарных статей (толкований) и примеров использования в художественной литературе соответствующих понятий из словаря С. И. Ожегова с помощью программных систем Link Grammar Parser и Dialing и т. д.

Далее предполагается рассмотреть диаграммы, полученные на выходе систем Link Grammar Parser и Dialing средствами математической логики. Речь идет об исследовании возможностей применения в компьютерной лингвистике ряда конструкций и понятий математической логики, таких как конструкция Л. Генкина, реализация и опускание типов, модельная полнота, форсинг, а также ряда неклассических логик.

Предполагается провести анализ средствами математической логики свойств лексических функций И. А. Мельчука в контексте определения смысла текста и теоретико-множественных моделей языка, предложенных С. Маркусом.

В будущем работа будет также включать исследование комбинаторных свойств лингвистических определений и конструкций, возможностей применения в компьютерной лингвистике понятий из теоретико-множественной и алгебраической топологии и исследование географических понятий.

1. Формальные методы исследования семантики текстов

Семантика – раздел лингвистики, изучающий смысловое значение единиц языка: отдельных слов, словосочетаний, предложений, фрагментов текста. На данный момент существует ряд машинно-ориентированных методов отображения смысла высказываний.

Например, И. А. Мельчук ввел понятие лексической функции, развил понятия синтаксических и семантических валентностей и рассмотрел их в контексте толково-комбинаторного словаря [1]. В. Ш. Рубашкин и Д. Г. Лахути ввели иерархию синтаксических связей для более эффективной работы семантического анализатора. Подход И. А. Мельчука поддержан в программной системе Dialing [2].

Появилось понятие универсального языка представления знаний. Он может быть удобным инструментом для осуществления вывода новых знаний из уже имеющихся. Вполне возможно, что именно в направлении создания подобных семантических языков будут развиваться исследования в будущем. Например, в настоящее время система Knowledge Vault содержит 1,6 миллиарда фактов. Система NELN, разрабатываемая в рамках проекта ReadTheWeb университетом Карнеги-Меллона, содержит более 50 миллионов утверждений, дополнительно характеризующихся различными степенями доверия.

Еще один подход – это использование синтаксического анализатора Link Grammar Parser [3], разработанного в университете Корнеги-Меллона, базирующегося на некоторой специальной теории синтаксиса. Отметим, что данная теория, вообще говоря, отличается от классической теории синтаксиса. Получив предложение, система приписывает к нему синтаксическую структуру, которая состоит из множества помеченных связей (коннекторов), соединяющих пары слов.

Получаемые диаграммы, по сути, являются аналогами, так называемых деревьев подчинения предложений. В деревьях подчинения от главного слова в предложении можно задать вопрос к второстепенному. Таким образом, слова выстраиваются в древовидную структуру. Главной причиной, по которой анализатор называют семантической системой, можно считать уникальный по полноте набор связей. Имеется около 100 основных связей, при этом некоторые из них дополнительно имеют 3–4 варианта.

2. Источники информации

Наши исследования проводятся в соответствии со следующими пунктами.

1. Выборка наиболее важных понятий, относящихся к времени и пространству, из толкового словаря С. И. Ожегова.

2. Создание массива перефразированных вариантов различных предложений и методов оценивания их похожести.

3. Анализ словарных статей (толкований) и примеров использования в художественной литературе соответствующих понятий из словаря С. И. Ожегова с помощью программных систем Link Grammar Parser и Dialing.

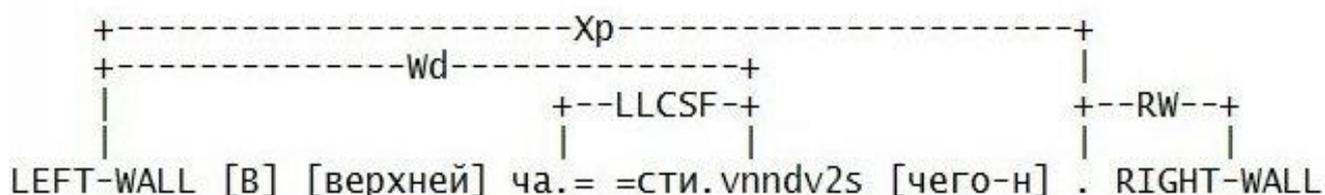
4. Анализ диаграмм, полученных на выходе систем Link Grammar Parser и Dialing средствами математической логики. Исследование возможностей применения к географическим понятиям.

5. Интеграция результатов исследований в базу знаний.

Ниже приведены два примера.

Пример 1. Анализ словарной статьи из словаря С. И. Ожегова в системе Link Grammar Parser.

```
linkparser> В верхней части чего-н.
No complete linkages found.
Found 4 linkages (4 had no P.P. violations) at null count 4
Linkage 1, cost vector = (UNUSED=4 DIS= 2.00 LEN=10)
```



Пример 2. Анализ словарной статьи из словаря С.И. Ожегова в системе Dialing.



На задворках – на самом последнем, невидном и невыгодном месте.

3. Перефразирования предложений

В ряде случаев в базе данных целесообразно иметь перефразированные варианты предложений и методы оценивания их близости, или, иначе говоря, схожести. Это особенно важно для предложений, содержащих глаголы каузации движения и изменения положения в пространстве, ввиду сложности предложений их содержащих. В этот класс входят глаголы, выражающие такие понятия, как: перемещать, приближать, удалять, нести, давать, брать, класть, поднимать, опускать, бросать, ловить, посылать и др. Определенный интерес представляют предложения, содержащие некоторые наречия: впереди, позади, сбоку, раньше, позже, еще, уже; именно с точки зрения их перефразирования.

В работах [4] дано описание, каким образом можно сравнивать перефразированные предложения, для случая использования анализатора Link Grammar Parser. Предположим, что

L – множество слов некоторого естественного языка. Для любого слова $x \in L$ обозначим $Norm(x)$ его нормализованную форму. Запись $Syn(x, y)$ обозначает, что x, y – синонимы.

Возникают два вида эквивалентностей:

- 1) $x_1 \approx x_2 \leftrightarrow x_1 = x_2 \vee Syn(x_1, x_2)$;
- 2) $x_1 \equiv x_2 \leftrightarrow Norm(x_1) = Norm(x_2)$.

Предложение рассматриваем, как вектор с компонентами из слов $\bar{x} = \langle x_1, \dots, x_n \rangle$. Функция $Norm$ может быть естественно распространена на предложения $Norm(\bar{x}) = \langle Norm(x_1), \dots, Norm(x_n) \rangle$. Текст $T = \langle \bar{x}_1, \dots, \bar{x}_n \rangle$ есть последовательность предложений.

Пусть запись $\bar{x} \models P(x_i, x_j)$ обозначает, что в схеме разбора предложения $\bar{x} = \langle x_1, \dots, x_n \rangle$ посредством анализатора Link Grammar Parser имеется коннектор типа P , идущий от слова x_i к слову x_j . Знак \models означает, что фактически мы имеем дело с моделью. Основным множеством модели является множество пар $\{ \langle 1, x_1 \rangle, \dots, \langle n, x_n \rangle \}$. Так как одно и то же слово может входить в предложение два и более раз, то это приводит к необходимости рассмотрения именно пар, а не отдельных слов. В силу сказанного выше корректным является даже обозначение $\bar{x} \models \varphi$, где φ – формула, например исчисления предикатов первого порядка. Фактически \bar{x} одновременно является обозначением и для вектора, и для модели.

Предположим, что даны два предложения $\bar{x} = \langle x_1, \dots, x_n \rangle$, $\bar{y} = \langle y_1, \dots, y_m \rangle$. Интерес представляют функции f такие, что $dom(f) \subseteq \{1, \dots, n\}$, $range(f) \subseteq \{1, \dots, m\}$ с дополнительными свойствами типа: $f(i) = j \rightarrow x_i \approx y_j$, $f(i) = j \rightarrow x_i \equiv y_j$ и другие подобные им.

При сопоставлении двух предложений, точнее при анализе их на близость, осуществляется проверка ряда логических свойств. Например, пусть $f(i_1) = j_1$, $f(i_2) = j_2$. Теперь приведены примеры такого рода свойств.

1. Инвариантность коннектора
 $\bar{x} \models P(x_{i_1}, x_{i_2}) \rightarrow \bar{y} \models P(y_{j_1}, y_{j_2})$.
2. Замена коннектора на дизъюнкцию других
 $\bar{x} \models P(x_{i_1}, x_{i_2}) \rightarrow \bar{y} \models \bigvee_t Q_t(y_{j_1}, y_{j_2})$.
3. Расщепление коннектора на два коннектора
 $\bar{x} \models P(x_{i_1}, x_{i_2}) \rightarrow \exists k (\bar{y} \models Q(y_{j_1}, y_k) \wedge R(y_k, y_{j_2}))$.
4. Расщепление коннектора на два коннектора с инверсией
 $\bar{x} \models P(x_{i_1}, x_{i_2}) \rightarrow \exists k (\bar{y} \models Q(y_{j_2}, y_k) \wedge R(y_k, y_{j_1}))$.

Принимая во внимание, что \bar{y} является обозначением для соответствующей модели, формула из третьего пункта может быть переписана в виде $\bar{x} \models P(x_{i_1}, x_{i_2}) \rightarrow \bar{y} \models \exists y Q(y_{j_1}, y) \wedge R(y, y_{j_2})$. По аналогии может быть записана формула из четвертого пункта.

Резюмируя, можно сказать, что в нашем распоряжении имеются правила вида

$$R_i : \bar{x} \models \varphi_i(x_1, x_2) \rightarrow \bar{y} \models \psi_i(y_1, y_2).$$

Отметим, что для английского языка такого рода правил нами фиксировано более тридцати. Для русского и других языков этот вопрос менее изучен. Для системы Dialing тоже могут быть сформулированы аналогичные правила, но это более сложный вопрос.

Далее строится функция f и проводится анализ по поводу выяснения того, встречаются ли индексы $i_1, i_2, j_1 = f(i_1), j_2 = f(i_2)$ такие, что на конкретных словах из предложений \bar{x}, \bar{y} выполнено правило R_i , т. е. $\bar{x} \models \varphi_i(x_{i_1}, x_{i_2}) \rightarrow \bar{y} \models \psi_i(y_{j_1}, y_{j_2})$. Для простоты можно говорить, что правило выполняется на паре $\langle i_1, i_2 \rangle$.

Теперь рассмотрим множество всех таких пар $\langle i_1, i_2 \rangle$, на которых выполнено одно из правил. Обозначим это множество I , и пусть его мощность $|I| = n$. Отметим, что анализатор Link Grammar Parser допускает между двумя словами наличие только одного коннектора. Поэтому будет выполняться не более чем одно правило.

Далее пусть n_1, n_2 – количество коннекторов, получающихся в результате анализа предложений \bar{x}, \bar{y} соответственно. В качестве меры схожести двух предложений можно ввести $\mu_0(\bar{x}, \bar{y}) = n / \max(n_1, n_2)$ или $\mu_1(\bar{x}, \bar{y}) = 2n / (n_1 + n_2)$.

В заключение отметим, что аналогичные идеи могут быть применены для анализа перефразированных вариантов предложений, а именно оценивания их близости, на основе диаграмм, получаемых на выходе системы Dialing.

Исследование выполнено при финансовой поддержке РФФИ в рамках научного проекта № 19-07-01134.

Список литературы

1. Мельчук И. А. Опыт теории лингвистических моделей типа «Смысл \Leftrightarrow Текст». М. : Наука, 1974. 315 с.
2. Сокирко А. В. Семантические словари в автоматической обработке текста: дис. ... канд. техн. наук. М. : МГПИИЯ, 2000. 108 с.
3. Link Grammar Documentation. 2015. URL: <http://www.abisource.com/projects/link-grammar>.
4. Батура Т. В., Бакиева А. М., Еримбетова А. С., Мурзин Ф. А., Сагнаева С. К. Грамматика связей, релевантность и определение тем текстов. Новосибирск : Изд-во СО РАН, 2018. 91 с.

UDK 519.68; 681.513.7; 612.8.001.57

ANALYSIS OF TEMPORAL AND SPATIAL CONCEPTS OCCURRING IN NATURAL LANGUAGE TEXTS

T. V. Batura

Ph.D., senior researcher, A. P. Ershov Institute of Informatics Systems SB RAS,
630090, Novosibirsk, Russia, Acad. Lavrentjev pr., 6, e-mail: tatiana.v.batura@gmail.com

L. V. Efimova

Ph.D. student, A. P. Ershov Institute of Informatics Systems SB RAS,
630090, Novosibirsk, Russia, Acad. Lavrentjev pr., 6, e-mail: efimova_l@ngs.ru

A. S. Yerimbetova

Ph.D. student, Novosibirsk State University,
630090, Novosibirsk, Russia, Pirogov str., 1, e-mail: aigerian@mail.ru

A. B. Kasekeeva

Ph.D. student, L. N. Gumilyov Eurasian National University,
010000, Astana, Kazakhstan, Satbayev str., 2, e-mail: aibike_7474@mail.ru

F. A. Murzin

Ph.D., deputy director, A. P. Ershov Institute of Informatics Systems SB RAS,
630090, Novosibirsk, Russia, Acad. Lavrentjev pr., 6, e-mail: murzin@iis.nsk.su

Abstract. The article is devoted to the creation of a knowledge base containing information on temporal and spatial concepts found in natural language texts. The knowledge base contains: the most important concepts relating to time and space from the explanatory dictionary S.I. Ozhegov;

rephrased versions of some sentences; the results of the analysis of dictionary entries (interpretations) and examples of the use in literature of corresponding concepts from the S.I. Ozhegov dictionary with the help of software systems Link Grammar Parser and Dialing; results of the analysis of diagrams obtained at the output of Link Grammar Parser, etc. The investigations were carried out under the financial support of the Ministry of Education and Science of the Republic of Kazakhstan (grant 2018-2020 MES RK № AP05133550) and the Integration Project of the SB RAS (AAAA-A18-118022190008-8).

Keywords: computational linguistics, semantics, temporal and spatial concepts, Link Grammar Parser, Dialing

References

1. *Melchuk I. A.* Opyt teorii lingvisticheskikh modelei tipa «Smysl \Leftrightarrow Text» [Experience of the linguistic models theory «Meaning \Leftrightarrow Text»]. M. : Nauka, 1974. 315 p.
2. *Sokirko A. V.* Semanticheskiye slovari v avtomaticheskoi obrabotke teksta [Semantic dictionaries in automatic text processing] : PhD Thesis. M. : The Maurice Thorez Institute of Foreign Languages, 2000. 108 p.
3. Link Grammar Documentation. 2015. URL: <http://www.abisource.com/projects/link-grammar>.
4. *Batura T. V., Bakiyeva A. M., Yerimbetova A. S., Murzin F. A., Sagnayeva S. K.* Grammatika svyazei, relevantnost i opredeleniye tem textov [Link grammar, relevance and definition of themes of texts]. Novosibirsk : SB RAS, 2018. 91 p.

Использование графических ускорителей для выявления функциональных сигналов в регуляторных районах генов прокариот

А. В. Бочарников

аспирант, Институт систем информатики им. А. П. Ершова СО РАН,
630090, Новосибирск, Россия, пр. Акад. Лаврентьева, 6, e-mail:
andrey.bocharnikov@gmail.com

Е. В. Игнатьева

канд. биол. наук, ст. науч. сотр., Институт цитологии и генетики СО РАН,
630090, Новосибирск, Россия, пр. Акад. Лаврентьева, 10;
доц., Новосибирский государственный университет,
630090, Новосибирск, Россия, ул. Пирогова, 1, e-mail: eignat@bionet.nsc.ru

О. В. Вишневский

канд. биол. наук, науч. сотр., Институт цитологии и генетики СО РАН,
630090, Новосибирск, Россия, пр. Акад. Лаврентьева, 10;
ст. преподаватель, Новосибирский государственный университет,
630090, Новосибирск, Россия, ул. Пирогова, 1, e-mail: oleg@bionet.nsc.ru

Аннотация. Нами предложен новый алгоритм оценки представленности вырожденных олигонуклеотидных мотивов в выборке нуклеотидных последовательностей. Показана его высокая производительность в сравнении с ранее предложенным подходом. С его использованием были проанализированы промоторы генов, экспрессия которых достоверно отличалась в AGRP-нейронах гипоталамуса голодающих и сытых мышей, и выявлены значимые олигонуклеотидные мотивы, ассоциированные с голоданием.

Ключевые слова: олигонуклеотидный мотив, GPGPU

Введение

Выявление функциональных сигналов в регуляторных районах генов является важной задачей современной биоинформатики и необходимо для понимания базовых механизмов регуляции транскрипции и трансляции.

Выявление олигонуклеотидных мотивов *de novo* является одним из наиболее ранних и широко используемых биоинформатических подходов обнаружения контекстных сигналов в регуляторных районах генов, но их использование затруднено гигантским разнообразием возможных вариантов. Так, мотив длиной в 8 нуклеотидов, записанный в пятнадцатибуквенном IUPAC коде (табл. 1) составляет $15^8 \sim 2,5 \times 10^9$ различных вариантов записи в четырехбуквенном коде. Это заставляет исследователей использовать различные эвристические подходы, основанные на анализе частот k -плетов, деревьев суффиксов, локальном множественном выравнивании, EM методе (Expectation–Maximization) и т. д. В то же время такие подходы не гарантируют нахождения наиболее достоверного мотива. Решением может являться использование полнопереборного строкового подхода, что требует применения высокопроизводительных массивно-параллельных вычислительных систем, таких как GPU ускорители.

Ранее нами был предложен такой полнопереборный подход, реализованный в виде программной системы Argo_CUDA [1] как на CPU, так и на GPU устройствах и продемонстрирована его высокая эффективность в сравнении с существующими подходами.

В данной работе мы предлагаем новый подход, основанный на использовании дерева префиксов мотивов и позволяющий существенно увеличить производительность.

Предложенный подход был использован для анализа промоторов генов, экспрессия которых достоверно отличалась в AGRP-нейронах гипоталамуса голодающих и сытых мышей. В промоторах генов, повышающих экспрессию в ответ на голод и понижающих ее, были найдены достоверные олигонуклеотидные мотивы. Анализ полученных мотивов выявил достоверное сходство с рядом известных сайтов связывания транскрипционных факторов. Классификация генов, промоторы которых содержали найденные сигналы с использованием системы DAVID, выявила их достоверную ассоциацию с голоданием.

1. Материалы и методы

1.1. Описание алгоритма поиска мотивов, основанного на дереве префиксов мотивов

Для полнопереборной оценки представленности каждого из 15^k вырожденных мотивов длины k в выборке из N нуклеотидных последовательностей длины L необходимо сравнить каждый мотив длины k с каждой k -символьной подстрокой этой выборки, что потребует $N \times (L - k + 1)$ сравнений. Для ускорения этого процесса удобно записывать нуклеотидные последовательности и мотивы в виде бинарных хэшей (табл. 1).

Таблица 1

Бинарное представление вырожденных олигонуклеотидов записанных в пятнадцатibuквенном IUPAC-коде

IUPAC code	Нукл.	Хэш
A	A	0001
T	T	0010
W	A/T	0011
G	G	0100
R	A/G	0101
K	T/G	0110
D	A/T/G	0111
C	C	1000
M	A/C	1001
Y	T/C	1010
H	A/T/C	1011
S	G/C	1100
V	A/G/C	1101

IUPAC code	Нукл.	Хэш
B	T/G/C	1110
N	A/T/G/C	1111

В этом случае для записи одного символа требуется 4 бита, а сравнение мотива с 8-нуклеотидной подстрокой возможно с использованием одной операции AND (табл. 2).

Таблица 2

Сравнение мотива WWWWAAAA и олигонуклеотида ATATAAAA

boolean match = (hash & motif_hash) == hash		
motif_hash	WWWWAAAA	0011 0011 0011 0011 0001 0001 0001 0001
hash	ATATAAAA	0001 0010 0001 0010 0001 0001 0001 0001
match	true	0001 0010 0001 0010 0001 0001 0001 0001

Если хотя бы в одной позиции мотив не совпадает с олигонуклеотидом, то результат побитового AND будет равен нулю. Таким образом, сложность сравнения мотива с подстрокой снижается с 8 операций посимвольного сравнения при стандартной записи до одной операции побитового AND при записи в виде хэша.

Полный перебор всех 15^k вариантов мотивов можно представить в виде обхода дерева. Корневому узлу соответствует пустая строка. У каждого узла, кроме листовых, имеется 15 потомков, в каждом из которых хранится символ пятнадцатибуквенного IUPAC-алфавита. В каждом узле текущий префикс мотива определяется набором символов из всех узлов на пути от корня к текущему узлу. Для каждой вершины поддерево со всеми узлами-потомками содержит все возможные варианты мотивов с заданным префиксом. В листовых вершинах префиксом является полный мотив.

Мотив совпадает с подстрокой входной выборки только в том случае, если их префиксы совпадают. Таким образом, каждому возможному префиксу мотива соответствует множество подстрок входной выборки нуклеотидных последовательностей.

Для сокращения числа сравнений каждого мотива со всеми подстроками входной выборки был предложен метод построения диапазонов подстрок по префиксу мотива длины l_p , состоящий из трех шагов.

Шаг 1. Представление входной выборки в виде набора отсортированных хэшей

Входные последовательности представляются в виде множества всех входящих в них k -буквенных подстрок. После чего подстроки преобразуются в единый список хэшей и сортируются в лексикографическом порядке.

Шаг 2. Построение диапазонов для нуклеотидных префиксов (в четырехбуквенном коде)

На втором шаге перебираются все возможные олигонуклеотидные префиксы заданной длины (строка 3). Для каждого префикса с помощью бинарного поиска определяется диапазон во входной выборке, где он представлен (строки 5–6).

```

1 total_prefixes = pow(4, prefix_size)
2 Ranges ranges_by_olig_prefix
3 for (i = 0; i < total_prefixes; i++)
4   olig_prefix_hash = olig_idx_to_hash(i)
5   low_bound = lower_bound(sequences.begin(), sequences.end(), olig_prefix_hash)
6   up_bound = upper_bound(sequences.begin(), sequences.end(), olig_prefix_hash |
7     0xFFFFFFFF)
8   if (low_bound != up_bound)
9     ranges_by_olig_prefix[olig_prefix_hash].add({ low_bound, up_bound })

```

Где *prefix_size* – длина префикса мотива l_p , *olig_idx_to_hash* – функция, которая по индексу олигонуклеотида возвращает его представление в виде хэша (табл. 2), *lower_bound* – позиция первого элемента, который не меньше заданного, *upper_bound* – позиция первого элемента, который больше заданного, *sequences* – отсортированный список хэшей всех подстрок длины k входной выборки.

Шаг 3. Построение диапазонов для префиксов мотивов (в пятнадцатибуквенном IUPAC-коде)

На третьем шаге перебираются все возможные варианты префиксов мотивов заданной длины (строка 3). Для каждого из префиксов мотива рассматриваются все варианты его записи в виде олигонуклеотида (строка 5) и для них объединяются диапазоны, посчитанные в шаге 2 (строка 8).

```

1 total_prefixes = pow(16, prefix_size)
2 Ranges ranges_by_motif_prefix
3 for (i = 1; i < total_prefixes; i++)
4   motif_prefix_hash = motif_idx_to_hash(i, prefix_size)
5   for (j = 0; j < motif_variants(i); j++)
6     olig_hash_mask = motif_variant(motif_prefix_hash)
7     range = ranges_by_olig_prefix[olig_hash_mask]
8     ranges_by_motif_prefix[motif_prefix_hash].add(range)

```

Где *motif_idx_to_hash* – функция, которая по индексу мотива возвращает его представление в виде хэша.

После того как получены диапазоны для всех возможных префиксов мотивов, они могут быть использованы для оценки представленности мотивов в выборке:

```

1 for (i = 0; i < pow(15,8); i++)
2   motif_hash = motif_idx_to_hash(i)
3   motif_prefix = motif_hash >> (32 - 4*prefix_size)
4   motif_ranges = ranges.get(motif_prefix)
5   for (j = 0; j < ranges.size; j++)
6     for (k = motif_ranges[j].start; k < motif_ranges[j].end; k++)
7       match = (sequence_hashes[k] & motif_hash) == sequence_hashes[k]

```

Таким образом, число сравнений мотива с подстроками входной выборки существенно снижается. Проведенный нами анализ показал, что количество перебираемых подстрок уменьшается, а время расчета диапазонов растет в зависимости от длины префикса. Оптимальным размером префикса является 5 или 6 в зависимости от размеров входной выборки.

1.2. Реализация алгоритма поиска мотивов на GPU

Современные GPU-ускорители содержат тысячи вычислительных ядер, позволяющих одновременно обчитывать десятки тысяч потоков. При этом наиболее эффективно с помощью GPU могут быть распараллелены задачи, где отсутствуют зависимости по данным. К таким задачам может быть отнесен и расчет представленности мотива в выборке нуклеотидных последовательностей.

```

1 motif_index = blockIdx.x * blockDim.x + threadIdx.x
2 motif_hash = idx_to_hash(motif_index)
3 extern __shared__ unsigned char matched_sequences[]
4 motif_prefix = motif_hash >> (8 - prefix_size)*4
5 Range range = ranges[motif_prefix]
6 offset = threadIdx.x * sequences_count
7
8 for (i = range.start_pos; i < range.end_pos; i++)
9   hash = sequence_hashes[i]
10  match = (hash & motif_hash) == hash

```

```

11   if (match)
12       seq_id = hash_to_sequence_id[i]
13       matched_sequences[offset + seq_id] = 1
14
15 for (i = offset; i < (offset + sequences_count); i++)
16     if (matched_sequences[i]) presense += 1
17 motif_presense[motif_index] = presense

```

При расчетах на GPU каждый поток в блоке формирует очередной индекс мотива (строка 2) и с помощью функции *idx_to_hash* преобразует его в хэш. Префикс вычисляется с помощью битового сдвига хэша мотива на $(8\text{-}prefix_size)*4$ бит вправо (строка 4). Полученный префикс используется для получения соответствующего ему диапазона подстрок входной выборки (строка 5). По индексу каждой подстроки, совпадающей с мотивом (строка 10), в массив результатов *matched_sequence* записывается единица (строка 13). Каждому потоку из блока выделяется свой участок памяти для результатов размером *sequence_count* (строка 6, 13, 15). Встречаемость мотива во входной выборке оценивается количеством единиц в массиве результатов (строки 16–17). Для ускорения сохранения результатов сравнений массив *matched_sequences* расположен в разделяемой памяти GPU (строка 3). Размер *S* разделяемой памяти ограничен (48–96 кб на блок), поэтому максимальное число последовательностей, обрабатываемых за 1 запуск вычислительного ядра, составляет *S/THREAD_PER_BLOCK* .

2. Результаты и обсуждение

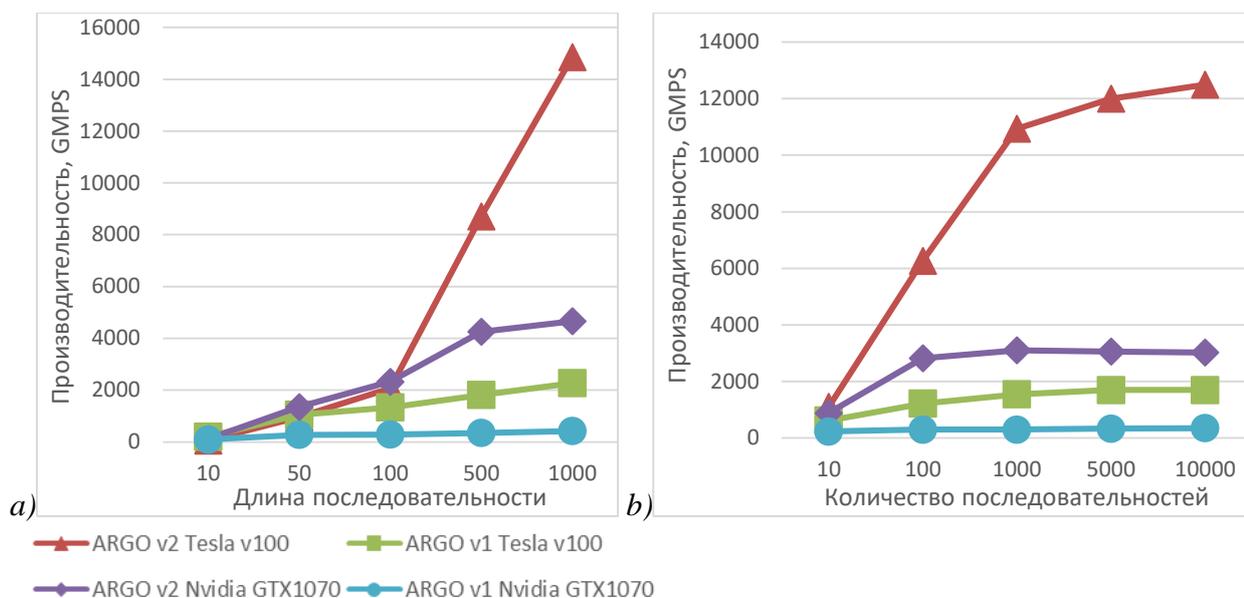
2.1. Оценка производительности работы программы на разных вычислительных устройствах.

Для того чтобы получить оценки производительности работы программы, не зависящие ни от длины и количества анализируемых последовательностей ни от длины мотивов, будем измерять производительность в количестве сравнений позиций мотивов, с позициями выборки последовательностей за единицу времени. Для набора мотивов *M* и выборки последовательностей *D* производительность *G* вычисляется следующим образом:

$$G = \frac{|M| \times |D|}{t \times 10^9} = \frac{k \times N_{mot} \times N_{seq} \times (L - k + 1)}{t \times 10^9}$$

Где $|M|$ – суммарное количество всех букв в наборе вырожденных олигонуклеотидных мотивов, $|D|$ – суммарное количество всех позиций в выборке последовательностей, с которыми возможно сравнение с мотивами, t – время работы в секундах.

С использованием предложенной меры мы провели оценку эффективности использования разработанного метода на двух графических ускорителях: NVidia GTX 1070 и NVidia Tesla v100. На рисунке приведены сравнительные оценки производительности *G* на различных вычислительных платформах на выборке в зависимости *a)* от длины последовательностей и *b)* от их количества. Длина префикса 5.



Оценка производительности GMPS (ось Y) на различных устройствах в зависимости a) от длины последовательностей (ось X). Количество последовательностей 100; b) от количества последовательностей (ось X). Длина последовательностей 128 нт

Рисунок демонстрирует, что NVidia Tesla v100, обладающая 5120 процессорами и более современной архитектурой, имеет в среднем в четыре раза более высокую производительность по сравнению с NVidia GTX 1070 (2048 ядер). Отметим, что оба графические ускорителя имеют худшую производительность при малых длинах анализируемых последовательностей и их малом количестве. Это можно объяснить тем, что внутренний цикл становится достаточно коротким и запуск вычислительного ядра происходит чаще. Можно отметить, что при количестве последовательностей от 1000 производительность обеих GPU стабилизируется. При этом производительность системы растет с увеличением размера выборки.

Рисунок показывает, что на всех рассмотренных GPU-устройствах при всех используемых параметрах анализируемых выборок предложенный нами новый алгоритм оценки представленности мотивов обеспечивает в среднем десятикратный прирост производительности.

2.2. Выявление олигонуклеотидных мотивов в промоторах генов голодающих мышей.

Список дифференциально экспрессирующихся генов мышей был получен из [2]. Он содержал 968 генов, повышающих экспрессию в ответ на голодание, и 610 генов, понижающих экспрессию. Из базы данных Ensembl (<https://www.ensembl.org/index.html>) с использованием системы BioMart были получены последовательности промоторов в районе [-200;+1] относительно старта транскрипции генов повышающих (выборка **Neg**) экспрессию в ответ на голодание и понижающих экспрессию (выборка **Pos**). Обе выборки были проанализированы с использованием предложенного нами нового метода.

Анализ полученных олигонуклеотидных мотивов с известными регуляторными мотивами проводился с помощью системы Tomtom (<http://meme-suite.org/doc/tomtom.html>). Функциональная аннотация генов, промоторы которых содержали найденные сигналы проводилась с использованием системы DAVID (<https://david.ncifcrf.gov>).

Проведенный анализ показал, что мотивы выборки **Neg** имеют достоверное сходство с такими сайтами связывания транскрипционных факторов (ССТФ), как: NFYA, FOXI1, SEBPZ, NFYB, NFYC, PBX3, SP4, ZN335, SP2, а мотивы выборки **Pos** имеют достоверное

сходство с такими ССТФ, как: SP1, E2F4, EGR1, SALL4, GLI2, KLF8, SRBP2, KLF5, ARNT2, ZIC1, HTF4, E2F7, PLAG1.

Функциональная аннотация генов, промоторы которых содержали мотивы, найденные в выборке **Neg**, показала их достоверную ассоциацию с **GO**-терминами ответа на эндоплазматический ретикулярный стресс (ERS). В их число входили такие хорошо известные гены – регуляторы или эффекторы ответа на ERS, как синовалин 1 (Syvn1), hypoxia up-regulated 1 (Hyoul) и белки теплового шока (Hspa5, Hsp90b1).

Мотивы из выборки **Pos** найдены в генах, достоверно ассоциированными с широким кругом биологических процессов, связанных с нормальным развитием и функционированием нейронов.

Заключение

Нами предложен новый высокопроизводительный полнопереборный алгоритм для выявления вырожденных олигонуклеотидных мотивов в больших выборках нуклеотидных последовательностей, основанный на дереве префиксов мотивов. С его помощью в промоторах дифференциально экспрессирующихся генов мышей были выявлены контекстные сигналы, достоверно ассоциированные с голоданием.

Работа поддержана бюджетным проектом № 0324-2019-0040.

Список литературы

1. *Henry F. et al.* Cell type-specific transcriptomics of hypothalamic energy-sensing neuron responses to weight-loss // *Elife*. 2015. Vol. 4. doi: 10.7554/eLife.09800.
2. *Vishnevsky O. V., Bocharnikov A. V., Kolchanov N. A.* ARGO_CUDA: Exhaustive GPU based approach for motif discovery in large DNA datasets // *Journal of Bioinformatics and Computation Biology*. 2018. Vol. 16(1), Epub 2017, Dec 10.

UDK 577.218

USING GRAPHICS ACCELERATORS FOR IDENTIFYING FUNCTIONAL SIGNALS IN THE REGULATORY REGIONS OF PROKARYOTIC GENES

A. V. Bocharnikov

Ph.D. student, A. P. Ershov Institute of Informatics Systems
Siberian Branch of the Russian Academy of Sciences
630090, Novosibirsk, Russia, Acad. Lavrentjev pr., 6, e-mail: andrey.bocharnikov@gmail.com

E. V. Ignatieva

Ph.D., senior researcher, Institute of Cytology and Genetics
Siberian Branch of the Russian Academy of Sciences
630090, Novosibirsk, Russia, Acad. Lavrentjev pr., 10, e-mail: eignat@bionet.nsc.ru

O. V. Vishnevsky

Ph.D., researcher, Institute of Cytology and Genetics
Siberian Branch of the Russian Academy of Sciences
630090, Novosibirsk, Russia, Acad. Lavrentjev pr., 10, e-mail: oleg@bionet.nsc.ru

Abstract. We have proposed a new algorithm for assessing the representation of degenerate oligonucleotide motifs in a sample of nucleotide sequences. Its high performance in comparison with

the previously proposed approach is shown. With its use, gene promoters were analyzed, the expression of which reliably differed in AGRP neurons of the hypothalamus of starving and well-fed mice, and revealed significant oligonucleotide motifs associated with starvation.

Keywords: Oligonucleotide motif, GPGPU

References

1. *Henry F. et al.* Cell type-specific transcriptomics of hypothalamic energy-sensing neuron responses to weight-loss // *Elife*. 2015. Vol. 4. doi: 10.7554/eLife.09800.
2. *Vishnevsky O. V., Bocharnikov A. V., Kolchanov N. A.* ARGO_CUDA: Exhaustive GPU based approach for motif discovery in large DNA datasets // *Journal of Bioinformatics and Computation Biology*. 2018. Vol. 16(1), Epub 2017, Dec 10.

Система поддержки прогнозирования развития опорной транспортной сети России**М. А. Бульонков**

канд. физ.-мат. наук., доц. Новосибирский государственный университет,
зав. лабораторией смешанных вычислений,
Институт систем информатики им. А. П. Ершова СО РАН,
630090, Новосибирск, Россия, пр. Акад. Лаврентьева, 6, e-mail: mike@iis.nsk.su

Т. В. Нестеренко

доц., Новосибирский государственный университет,
научный сотрудник, Институт систем информатики им. А. П. Ершова СО РАН,
630090, Новосибирск, Россия, пр. Акад. Лаврентьева, 6, e-mail: nest@iis.nsk.su

Аннотация. Рассматривается система автоматизации научных исследований в применении к задаче прогнозирования развития опорной транспортной сети России. Особое внимание уделяется визуальному и интерактивному представлению результатов моделирования. Для поддержки моделирования типа «как-чтобы» используются стохастические методы, основанные на массовом решении оптимизационной транспортной задачи для множества варьируемых параметров.

Ключевые слова: транспортная задача, моделирование, система автоматизации научных исследований

Введение

В 2009 году творческий коллектив, состоящий из сотрудников СИМОиР, ИСИ СО РАН, ИЭиОПП СО САН и НГУ начал инициативный проект МИКС (Модельная Информационная Картографическая Система) [1], предусматривающий создание информационной среды, направленной на поддержку задач управления регионами и их взаимодействия между собой. В работе [2] была представлена система МИКС-ПРОСТОР, основным назначением которой являлась автоматизация экономических экспериментов, связанных с прогнозированием развития опорной транспортной сети. Эта система позволяет задать прогноз изменения параметров, описывающих услуги по перевозке грузов и их обработке в транспортных узлах, и решить задачу оптимальной перевозки в условиях конкуренции нескольких видов транспорта. Таким образом, мы получаем вариант развития опорной транспортной сети, обеспечивающий рациональное взаимодействие входящих в нее видов транспорта. Иными словами, мы решаем задачу прогнозирования путём имитационного моделирования типа «что-если» [3].

1. Моделирование типа «что-если»

На первом этапе задача состояла в основном в автоматизации процесса научных исследований. В тот момент у исследователей уже имелся весь необходимый аппарат для проведения экспериментов. Транспортная задача формулировалась как задача линейного программирования [4], в которой величинами являются объемы и направление перевозки продуктов по плечам транспортной сети, коэффициентами – стоимость обработки грузов и длины плеч, а ограничения формулируются как неравенства вида:

– суммарный объем перевозимых по плечу продуктов вне зависимости от направления не должен превышать пропускную способность;

– объем ввозимого продукта должен быть равен сумме вывозимого и разгружаемого;

– объем потребляемого продукта должен быть не меньше объема выгружаемого и т. п.

Задача состоит в минимизации суммарных затрат на обработку грузов.

Исходные данные задавались в виде текстовых файлов стандартизованного формата и поступали на вход решателю, который представлял собой совокупность консольных программ [5]. Результат выдавался также в виде текстового файла. Таким образом, проведение единичного эксперимента могло занимать часы и даже дни.

Первоочередной задачей стала создание такого пользовательского интерфейса, который бы обеспечивал следующее:

1) формирование транспортной сети: задание перечней видов транспорта и продуктов, узлов и плеч (дорог), их пропускной способности и длины;

2) задание количества производимых и потребляемых в узлах продуктов, а также базовых тарифов перевозки продуктов по плечам и обработки в узлах (например, транзита или перегрузки с одного вида транспорта на другой);

3) запуск решателя [6];

4) визуализацию полученного результата.

Если для подготовки входных данных естественно использовать стандартные диалоговые возможности, то отображение результатов потребовало разработки специальных средств. Наиболее естественным представляется отображение транспортной сети на карте. По ряду соображений, обсуждение которых выходит за рамки данной работы, мы решили не привязываться к известным ГИС, даже если они и предоставляли программные интерфейсы для встраивания новой функциональности. Для наших целей и качественных требований оказалось достаточно решить проблему координатной привязки к «карте-подложке». Таким образом, система предоставляет на выбор одну из предобработанных карт, поверх которой отображаются узлы, плечи и прочие элементы транспортной сети.

На практике оказалось, что пространственное расположение на карте далеко не всегда удобно, например, в случае, когда «концентрация» узлов сильно отличается для разных регионов. Кроме того, дополнительная информация легче воспринимается, если плечи ортогональны и без изломов. Поэтому в дополнение к картам было разработано схематическое отображение сети, при котором приблизительно сохраняется взаимное расположение узлов. Грубо говоря, достаточно, чтобы Северный морской путь находился сверху, а Москва – левее Якутска, но необязательно, чтобы Диксон оказался правее Новосибирска. Два представления одной и той же транспортной сети показаны на рис. 1.

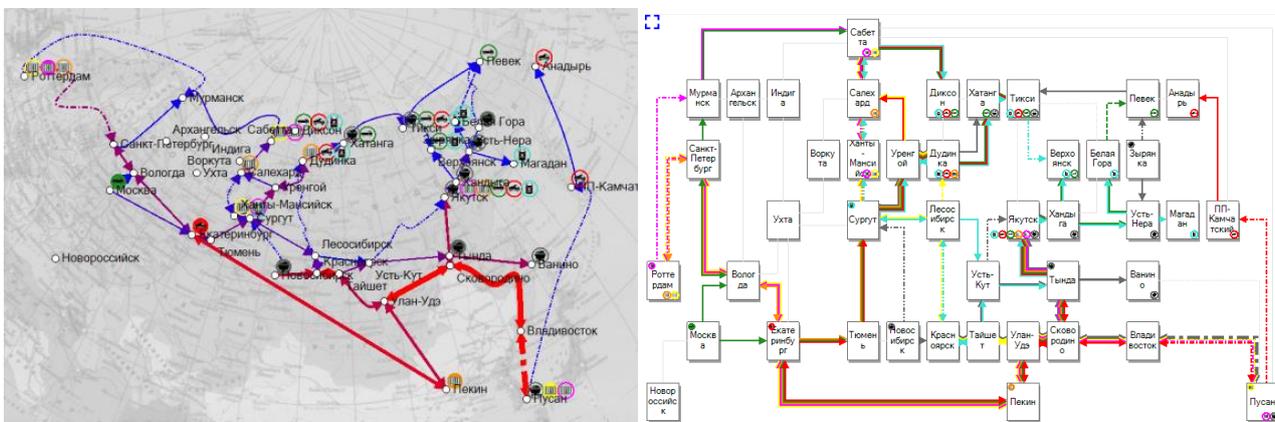


Рис. 1. Пространственное отображение транспортной сети

Для каждого плеча – на карте или схеме – требуется отобразить большое число параметров, таких как вид транспорта, пропускная способность, перевозимый объем и

стоимость перевозки (в единицах на километр, для каждого вида продукта), загруженность плеча и т. д. При этом количество визуальных средств весьма ограничено: тип линии (сплошная, пунктирная и т. п.), ее цвет и толщина. Использование всплывающих окон здесь не помогает, поскольку они не дают возможность получить общее представление о решении задачи. В текущей версии системы пользователь может выбрать один параметр, например, объем, который будет определять толщину и цвет линии, либо для всех продуктов вместе, либо для одного выбранного продукта.

Естественно, пространственное отображение не является единственно возможным. Иногда полезным оказывается обычное табличное представление, особенно, если его дополнить упорядочением строк, соответствующим транспортным коридорам, типичными примерами которых являются Северный морской путь (СМП) или Транссибирская магистраль. Следует отметить, что понятие транспортного коридора может возникать не только априорно, но и исходя из конкретного способа перевозки, возникающем в данном решении транспортной задачи для данного продукта. Причем в этот момент может возникать и ориентация плеч: по одному и тому же плечу уголь везут в одну сторону, а контейнеры – в другую. Кроме того, клетки таблицы могут быть раскрашены теми же цветами, что и дороги на карте.

Плечо	Вид трансг	Средний объём	Объём	Сред. цена за 1км.	Цена за 1км.	Сред. загрузе	Загруже	Зависим от Новосибир - Краснояр
Дудинка-Песосибир...	Река	2.22	2.00	1.44	1.30	0.22	0.20	0.58
Сургут-Новосибирск	Река	6.72	2.00	3.44	1.20	0.67	0.20	-0.55
Хандыга-Верхоянск	Авто	2.00	2.00	5.00	5.00	0.20	0.20	
Якутск-Усть-Кут	Река	2.00	2.00	3.00	3.00	0.20	0.20	
Мурманск-Роттердам	Море	3.22	3.00	0.64	0.60	0.03	0.03	0.58
ПП-Камчатский-Пус...	Море	3.00	3.00	0.60	0.60	0.03	0.03	
Усть-Нера-Белая Го...	Авто	4.00	4.00	9.00	9.00	0.40	0.40	
Мурманск-Сабетта	Пед	4.22	4.00	1.27	1.20	0.11	0.10	0.58
Хатанга-Тикси	Пед	5.00	5.00	13.00	13.00	0.13	0.13	
Ханты-Мансийск-Са...	Река	5.00	5.00	2.48	2.48	0.50	0.50	
Сургут-Ханты-Манси...	Река	6.00	6.00	3.00	3.00	0.60	0.60	
Дудинка-Хатанга	Авто	6.00	6.00	16.00	16.00	0.60	0.60	
Хандыга-Усть-Нера	Авто	6.00	6.00	10.50	10.50	0.60	0.60	
Диксон-Сабетта	Пед	6.00	6.00	17.00	17.00	0.15	0.15	

Рис. 2. Табличное представление решения

Следующий вопрос, который возникает в задаче типа «что-если» – сравнение двух решений. Предположим, что определен некоторый базовый вариант транспортной сети, для которого было получено решение, и теперь исследователь хочет выяснить, что будет, например, если повысить в два раза пропускную способность по Трансибу или снизить на 15 % тариф перевозки по СМП. Для этого необходимо просчитать данную ситуацию и сравнить ее с базовым решением. Мы опробовали несколько способов сравнения двух решений: от простого табличного представления до пространственно-анимационного, при котором отображение на карте одного решения плавно сменяется на другое, и обратно. Это позволяет «мгновенно» оценить места, в которых решения различаются, и определить степень отличия.

2. Моделирование типа «как-чтобы»

Нередко с точки зрения эксперта, ответ на вопрос «что-если» носит вспомогательный характер. На самом деле его интересует некоторая цель и значения параметров транспортной сети, при которых эта цель достигается. При этом часто как сама цель, так и возможные изменяемые параметры локальны. Примерами вопросов, соответствующих таким целям, являются следующие:

- при каких тарифах данное плечо будет загружено по крайней мере на 25 %?
- насколько имеет смысл увеличивать пропускную способность данного транспортного коридора?

Подобного рода задачи можно исследовать путем проведения множества экспериментов, отслеживая, как изменяется решение в зависимости от варьируемых параметров. Процесс этот трудоемкий и ненадежный, поскольку зависимость объемов перевозки от тарифов и пропускной способности при их одновременном изменении нелинейна и даже не непрерывна. Например, если тариф перевозки по плечу превосходит некоторое значение, то грузопоток может полностью переключиться на другое плечо.

Для автоматизации этих исследований мы используем стохастический подход следующим образом:

- 1) эксперт выбирает «ключевые» узлы или плечи транспортной сети и задает для них интервалы варьируемых параметров – тарифов обработки или пропускную способность;
- 2) эксперт указывает количество вариантов, которые нужно просчитать;
- 3) система генерирует и рассчитывает указанное количество вариантов, равномерно распределяя значения варьируемых параметров в указанных интервалах;
- 4) эксперт исследует результаты расчетов с помощью предоставляемых системой визуальных средств и методов статистического анализа.

Естественным расширением пространственной визуализации единичного эксперимента является отображение средних, а не конкретных объемов перевозки и загруженности. Однако можно таким же образом отобразить и другие статистические характеристики: например, зависимость объема перевозки по плечу от данного варьируемого параметра. Таким образом, пользователь сразу получает представление о том, на что влияет данный параметр. На рис. 3 приведен пример такого отображения. Положительная корреляция представлена красными линиями, а отрицательная – синими.

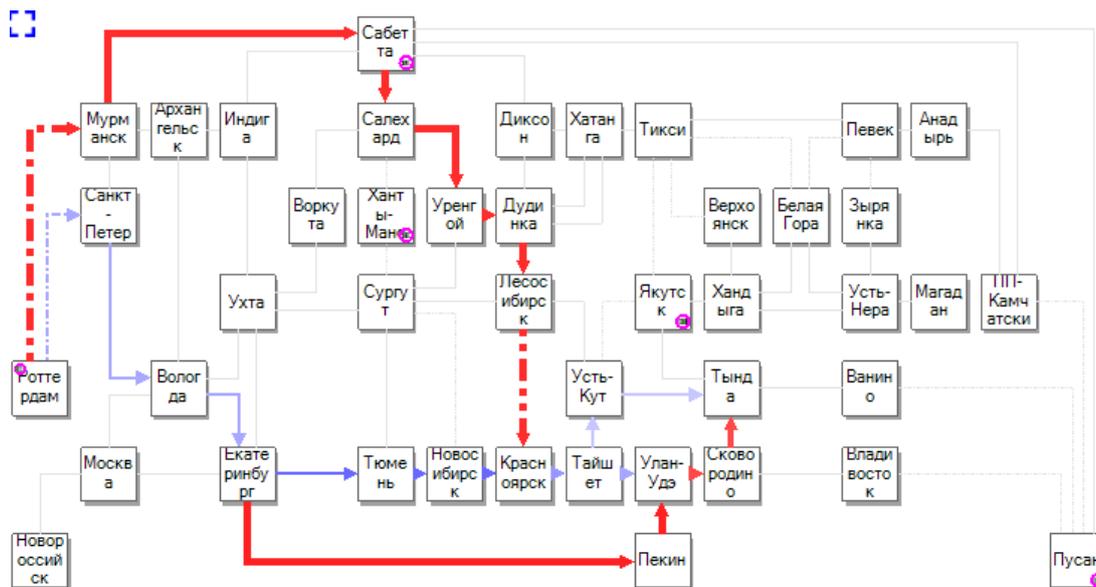


Рис. 3. Пространственное отображение корреляции

Рассматривая объемы перевозки по плечам как характеристики конкретного решения, можно ставить вопрос о схожести вариантов и, следовательно, ставить задачу кластеризации множества рассчитанных вариантов. Вопрос о качестве кластеризации и выборе наиболее подходящего метода – тема отдельного исследования.

Имея разбиение множества вариантов решений на кластеры, можно выделить их наиболее типичных представителей, а также показать частотные характеристики распределения значений варьируемых параметров в каждом кластере. Пример такого отображения представлен на рис. 4, из которого очевидно, что для выделенного кластера характерно малое значение по первым двум варьируемым параметрам.

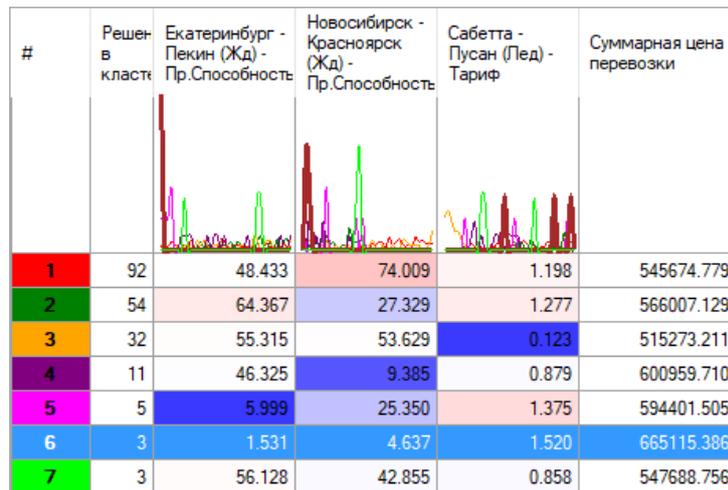


Рис. 4. Кластеризация множества решений

Транспонирование характеристической матрицы позволяет вычислять корреляцию не между вариантами решений, а между плечами. Знание этой корреляции дает возможность выделения транспортных коридоров: некоторая совокупность плеч с определенной степенью вероятности либо одновременно задействована, либо одновременно не задействована в перевозке. Следует отметить, что для разных продуктов коридоры могут отличаться. Если в матрице корреляции раскрасить клетки, градиентно выбирая цвет от красного (для положительной корреляции) до синего (для отрицательной корреляции), а затем переставить строки и столбцы матрицы так, чтобы красный цвет сосредоточился около главной диагонали, то возникающие красные блоки и будут означать транспортные коридоры. Пример такого отображения показан на рис. 5.

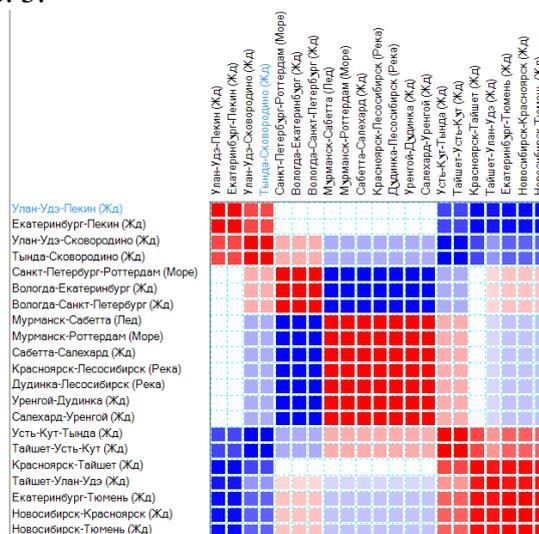


Рис. 5. Корреляция плеч – транспортные коридоры

Если варьируемых параметров несколько, то естественным образом возникает необходимость многомерной визуализации. Пусть, например, эксперт одновременно меняет тарифы для двух плеч. Отдельно 2D-графики зависимости объема перевозки по некоторому третьему плечу могут не давать адекватного полного представления (рис. 6, а, б). Оставаясь в двумерном пространстве, мы можем отобразить объем цветом и размером точек на плоскости, каждая из которых соответствует конкретному варианту, как показано на рис. 6, в. Такой способ визуализации можно назвать 2D+.

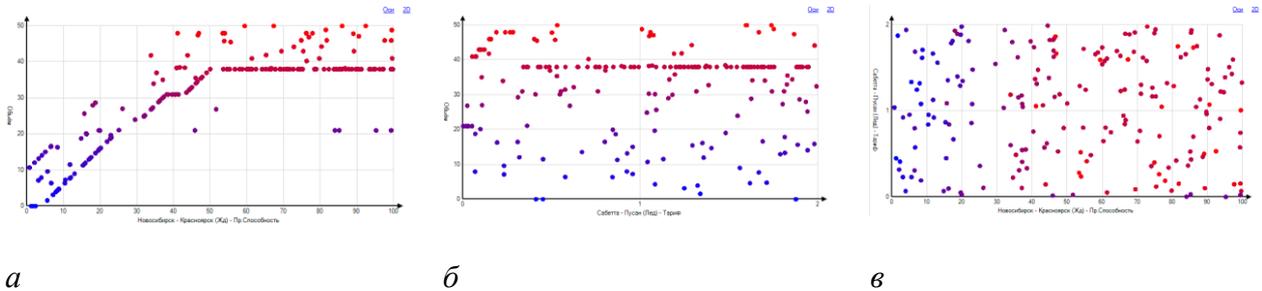


Рис. 6. Двумерная визуализация: а – график зависимости объема перевозки от пропускной способности, б – график зависимости объема перевозки от тарифа, в – график зависимости тарифа от пропускной способности

Возможна также и 3D-, и 3D+-визуализация, причем осям могут соответствовать как значения варьируемых параметров, так и любые вычисляемые числовые величины, а отображение отдельных точек не обязательно должно напрямую определяться перевозимым объемом. Так, на рис. 7 осям соответствуют два варьируемых параметра и суммарная стоимость перевозки, а цвет точки определяется цветом кластера, к которому отнесен данный вариант. Естественно, что в трехмерном случае необходима возможность покрутить картинку в пространстве.

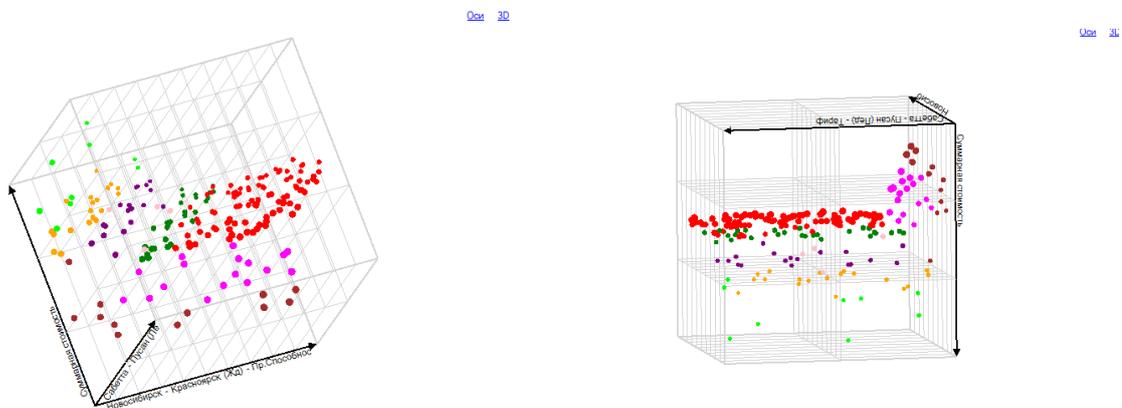


Рис. 7. Трехмерная визуализация

На этом примере становится очевидным предположение, что разбиение на кластеры связано с суммарной стоимостью перевозки.

Заключение

Мы рассмотрели средства автоматизации прогнозирования развития опорной транспортной сети, предоставляемых системой МИКС-ПРОСТОР. Можно заметить, что по мере развития системы, разработка новых средств визуализации позволяет не столько облегчить восприятие исследователем / экспертом полученного решения, сколько осознать и

сформулировать запросы на новые средства анализа, которые, в свою очередь, требуют новых средств визуализации и т. д.

Список литературы

1. Бульонков М. А., Карпан В. В., Малов В. Ю., Марусин В. В., Радченко В. В. Концептуальные вопросы построения Модельно-Информационно-Картографической Системы (МИКС) // Моделирование производственных и региональных систем на основе ГИС и информационных технологий : сб. науч. тр. / под ред. Ю. Ш. Блама, В. В. Радченко. Новосибирск : ИЭОПП СО РАН, 2011. С. 5–28.
2. Бульонков М. А., Филаткина Н. Н. Ситуационный анализ в системе транспортного прогнозирования МИКС-ПРОСТОП // Информационные технологии. 2013. № 8. С. 43–52.
3. Воробьёва В. В., Малов В. Ю., Радченко В. В., Поттер М. В., Серебрянников И. Е. Модель прогнозирования развития опорной транспортной сети // Моделирование производственных и региональных систем на основе ГИС и информационных технологий : сб. науч. тр. / под ред. Ю. Ш. Блама, В. В. Радченко. Новосибирск : ИЭОПП СО РАН, 2011. С. 68–96.
4. Гранберг А. Г. Оптимизация территориальных пропорций народного хозяйства. М. : Экономика, 1973.
5. Забиняко Г. И. Пакет программ целочисленного линейного программирования // Дискрет. анализ и исслед. опер. Серия. 2. № 6:2. 1999. С. 32–4.
6. Google OR-Tools. URL: <https://developers.google.com/optimization/> (дата обращения: 22.03.2019).

UDK 004.94

SUPPORT SYSTEM FOR FORECASTING THE DEVELOPMENT OF THE BASIC TRANSPORT NETWORK OF RUSSIA

M. A. Bulyonkov

Ph.D., associate professor, Novosibirsk State University,
630090, Novosibirsk, Russia, Pirogova str., 1,
Head of Mixed Computation Laboratory,
A. P. Ershov Institute of Informatics Systems,
Siberian Branch of the Russian Academy of Sciences,
630090, Novosibirsk, Russia, Acad. Lavrentjev pr., 6, e-mail: mike@iis.nsk.su

T. V. Nesterenko

associate professor, Novosibirsk State University,
630090, Novosibirsk, Russia, Pirogova str., 1,
researcher, A. P. Ershov Institute of Informatics Systems,
Siberian Branch of the Russian Academy of Sciences,
630090, Novosibirsk, Russia, Acad. Lavrentjev pr., 6, e-mail: nest@iis.nsk.su

Annotation. The system of scientific research automation is considered as applied to the task of forecasting the development of the supporting transport network in Russia. Special attention is paid to the visual and interactive presentation of simulation results. To support “how-to” modeling, stochastic methods are used based on the mass solution of the optimization transport problem for a variety of variable parameters.

Keywords: transportation problem, modeling, research automation system

References

1. *Bulyonkov M. A., Karpan V. V., Malov V. Y., Marusin V. V., Radchenko V. V.* Conceptual issues of building a Model Information-Mapping System (MIMS) [Modeling production and regional systems based on GIS and information technology]. Novosibirsk : IEIE SB RAS, 2011. P. 5–28. (in Russian)
2. *Bulyonkov M. A., Filatkina N. N.* Situational analysis in the MICS-PROSTOR transport forecasting system = Information Technology, 2013. № 8. P. 43–52. (in Russian)
3. *Vorobyova V. V., Malov V. Y., Radchenko V. V., Potter M. V., Serebryannikov I. E.* Prediction model of the core transport network development [Modeling production and regional systems based on GIS and information technology]. Novosibirsk: IEIE SB RAS, 2011. P. 68–96. (in Russian)
4. *Granberg A. G.* Optimization of the territorial proportions of the national economy. M. : Ekonomika Publ., 1973. (in Russian)
5. *Zabinyako G. I.* Software package of the integer linear programming = Discrete analysis and operations research. 1999. Vol. 2, № 6:2. P. 32–41. (in Russian)
6. Google OR-Tools. URL: <https://developers.google.com/optimization/> (date of the application 22.03.2019).

Модель управления вычислительным комплексом, реализованным на основе концепции БСМ

Л. А. Голубева

канд. физ.-мат. наук, доц., науч. сотр., Институт вычислительной математики и математической геофизики СО РАН,
630090, Новосибирск, Россия, пр. Акад. Лаврентьева, 6,
Новосибирский государственный университет,
630090, Новосибирск, Россия, ул. Пирогова, 1, e-mail: golubeva@labchem.sccc.ru

В. С. Горшунов

инженер, Институт вычислительной математики и математической геофизики СО РАН,
630090, Новосибирск, Россия, пр. Акад. Лаврентьева, 6,
аспирант, Новосибирский государственный университет,
630090, Новосибирск, Россия, ул. Пирогова, 1, e-mail: basil.gorshunov@gmail.com

В. П. Ильин

д-р физ.-мат. наук, проф., глав. науч. сотр., Институт вычислительной математики и математической геофизики СО РАН,
630090, Новосибирск, Россия, пр. Акад. Лаврентьева, 6,
Новосибирский государственный университет,
630090, Новосибирск, Россия, ул. Пирогова, 1, e-mail: ilin@sccc.ru

Аннотация. Вычислительный комплекс, базирующийся на концепции базовой системы моделирования (БСМ), включает в себя три важных модуля для построения геометрической модели, сеточной модели и решателя алгебраических уравнений. Цель настоящей работы – разработать модель управления данным вычислительным комплексом с помощью базы знаний, представленной в виде семантической сети.

Ключевые слова: Математическая физика, геометрическая структура данных, сеточная структура данных, функциональная структура данных, БСМ, семантические сети, базы знаний.

Введение

База знаний — это особого рода база данных, разработанная для оперирования знаниями (метаданными). База знаний содержит структурированную информацию, покрывающую некоторую область знаний. Современные базы знаний работают совместно с системами поиска информации, имеют классификационную структуру и формат представления знаний.

Представление знаний — вопрос, связанный с подбором представления конкретных и обобщённых знаний, сведений и фактов для накопления и обработки информации в ЭВМ. Основная проблема — научиться хранить знания таким образом, чтобы программы могли эффективно обрабатывать их.

Для представления знаний можно использовать семантические сети. Каждый узел такой сети представляет концепцию, а дуги используются для определения отношений между концепциями.

Одна из проблем в представлении знаний — как хранить и обрабатывать знания в информационных системах формальным способом так, чтобы машины могли использовать их

для достижения поставленных задач. Примеры применения — экспертные системы, Машинный перевод, компьютеризированное техническое обслуживание и системы извлечения и поиска информации (включая пользовательские интерфейсы баз данных). Одна из самых выразительных и детально описанных парадигм представления знаний, основано на семантических сетях [8].

В настоящей статье описывается подход к построению базы знаний для управления программным комплексом для решения трехмерных задач математической физики на основе концепции БСМ [7].

1. Описание семантической сети, моделирующей управление вычислительным процессом

При решении задач математической физики можно выделить три важных этапа: построение геометрической модели, построение сеточной модели, решение уравнений [1–6]. Составим семантическую сеть [8], которая будет включать в себя в качестве подсетей 3 семейства узлов, связанных последовательно, каждое из которых ответственно за один из следующих этапов вычислительного процесса:

- «Вороной» – задание геометрической и функциональной структуры данных;
- «Делоне» – построение сеточной структуры данных;
- «Чебышёв» – аппроксимация уравнений.

Каждая трех из подсетей, в свою очередь, так же является семантической сетью. Связи между узлами в этой семантической сети предоставляют возможность организовать конкретный вычислительный процесс. Таким образом, на каждом из трех этапов, представленных в виде подсети, решения задачи возможны различные варианты развития событий в зависимости от того. Можем использовать различные методы построения геометрии, сетки и различные методы решения уравнений – все это будет между собой связано. Получается путь в сети, который из себя представляет вычислительный процесс, пройдя по которому получим решение задачи, обладающее определенными качествами. В зависимости от того, какие условия накладываются на качество решения, мы можем либо принять такой подход, либо выбрать иной путь построения решения в любой из трех подсетей.

2. Описание узлов семантической сети

Каждый узел в семантической сети представляет из себя отдельную программу, в которой реализованы определенные алгоритмы. Узлы в сети могут быть одного из трех типов, в зависимости от того, к какой подсети они принадлежат. От типа узла зависит набор входных и выходных данных.

Все три подсети последовательно связаны между собой: «Вороной» – «Делоне» – «Чебышёв». В свою очередь, узлы внутри каждой из подсетей тоже могут между собой быть связаны, такое утверждение основывается на принципе декомпозиции расчетной области на подобласти. Цель декомпозиции – использование наиболее подходящего алгоритма в каждой из подобластей с точки зрения качества получаемого решения задачи во всей области. Опишем каждую из трех подсетей.

2.1. «Вороной»

Подсеть «Вороной» ответственна за построение, визуализацию геометрии и формирование геометрической структуры данных (ГСД) и функциональной структуры данных (ФСД). Входные данные для узлов подсети – описание расчетной области для конкретной решаемой задачи, которое может быть задано при помощи диалогового окна либо в виде

Подсеть текста на соответствующем языке описания геометрии. Выходные данные – различные ФСД, ГСД и их графическое представление. ФСД включает в себя данные, необходимые для функционирования узлов других подсетей, например, граничные значения, начальные разбиения границ для построения сеток и т. д.

Узлы могут иметь различное представление ГСД. Однако для каждой из ГСД можно построить общее внутреннее представление на основе какой-либо платформы с возможностью визуализации геометрии. В качестве такой платформы в настоящей работе используется «Гербарий» [4] – инженерно-исследовательская программная платформа. В «Гербарии» реализована возможность построения и визуализации геометрии. Геометрическая модель в «Гербарии» представляет из себя набор примитивов, к которым применены булевы операции объединения, вычитания. Под набором примитивов в данном случае понимаются объемные фигуры, полученные путём выдавливания или вращения криволинейной плоской замкнутой линии.

2.2. «Делоне»

Задачей узлов этой подсети ставится построение сеточной модели. Для каждого такого узла существует определенный набор входных значений – некоторые ГСД и ФСД, сформированные в подсети «Вороной», и набор выходных данных – сеточная структура данных (ССД).

ССД, которая представляет сеточную модель, может включать в себя набор сеточных вершин, сеточных ребер, сеточных граней и т. д. ССД каждого из узлов может различаться, но также, как и в случае с ГСД, может быть сведена к некоторой общей ССД, а значит, и иметь одно графическое представление на выбранной платформе, в нашем случае в «Гербарии».

Также сеточная модель обладает различными свойствами: качество сетки, структурированность, адаптивность [7]. Качественными характеристиками могут служить максимальный и минимальный угол по всем элементам, их отношение; максимальная и минимальная длина ребра; радиус вписанной или описанной окружности. В зависимости от условий, накладываемых на качество, можно выбрать тот или иной путь, при управлении вычислительным процессом.

2.3. «Чебышёв»

Целью функционирования подсети служит получение решения исходной задачи. Решением задачи будем называть алгебраическую структуру данных (АСД). Входными данными служат результаты, полученные на предыдущих этапах (в предыдущих подсетях): ССД и ФСД.

В зависимости от постановки исходной задачи ее решением может быть решение некоторой системы дифференциальных, интегральных или линейных алгебраических уравнений в узлах сеточной модели. [1,7]

3. Использование БЗ для управления вычислительным комплексом

Опишем ГСД, ФСД, необходимые для функционирования некоторых сеточных генераторов, а также ССД, полученные в результате работы таких генераторов и пример использования описанной концепции.

SubDat – сеточный генератор, способный строить структурированные сетки с равномерным шагом в трёхмерных областях, состоящих из параллелепипедов.

Базовые координаты (БК) – упорядоченный набор уникальных значений координат для каждой из координатных осей.

ГСД и ФСД SubDat выглядит следующим образом:

$nx\ ny\ nz$ – число БК для каждой из осей координат;

$k_1 k_2 \dots k_{nk}$ – список БК по каждой из трёх осей. Здесь и далее $k = x | y | z$;
 $sk_1 sk_2 \dots sk_{nk-1}$ – список числа шагов разбиения между базовыми координатами, sk_i – задает число шагов разбиения между БК k_i и k_{i+1} ;
 $nmat$ – число материалов в расчетной области, не может быть меньше 1;
 nsd – число подобластей (параллелепипедов) в расчетной области, не может быть меньше одного;
 $sd_1.t$ – номер материала подобласти;
 $sd_1.p$ – номер приоритета подобласти;
 $sd_1.kb sd_1.ke$ – номера БК оси O_k начала и конца подобласти соответственно, нумерация ведется с 0;
 $sd_1.bc_{kb} sd_1.bc_{ke}$ – номера граничных условий на гранях подобласти $k = k_{sd_1.kb}$ и $k = k_{sd_1.ke}$ соответственно;
 $ndbc_i$ – номер значения условия Дирихле узлу сетки;
 n_dbc – число значений условий Дирихле;
 $Vdbc_i$ – значения условия Дирихле.
 В ССД CubDat содержится описание узлов:
 $nnode$ – число узлов в расчетной области;
 $x_i y_i z_i$ – декартовы координаты узлов;
 $ntet$ – число тетраэдров в расчетной области;
 $n_i n_j n_k n_l$ – номера узлов – вершин тетраэдра, нумерация узлов идет с 0;
 $bc_i bc_j bc_k bc_l$ – номера граничных условий на гранях тетраэдра.

NetGen – универсальный сеточный генератор. Входные данные для NetGena должны быть записаны в формате .stl, .geo, .step. Выходной формат данных – .vol. Описание этих форматов доступно в открытых источниках [9].

В программном комплексе [7] посредством «Гербария» формируется внутренняя ГСД, входные данные для которой задаются с использованием графического интерфейса в виде диалогового окна. ГСД «Гербария» может быть преобразована на уровне программного кода в формат, понятный другим узлам семантической сети: CubDat и NetGen. Более того, соответствующая ГСД будет иметь визуальное представление в Демонстраторе «Гербария». То же справедливо и для ССД.

Приведем пример работы программного комплекса [7] на основе вышеописанной концепции:

3.1 Задание геометрии и формирование входных данных и ФСД для сеточного генератора CubDat. Модуль «Вороной»

Добавляем два тела кубической формы, указав начальные и конечные координаты диагонали каждого из тел (рис. 1).

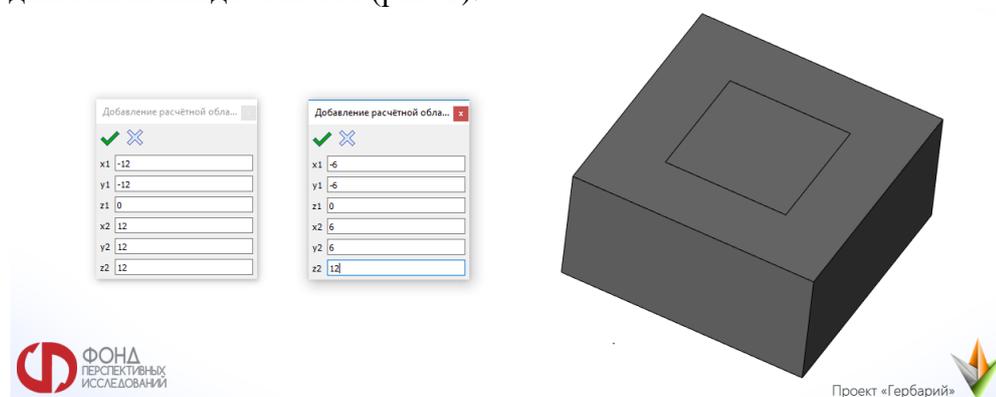


Рис. 1. Задание геометрии

Задаем отношение между телами с помощью булевых операций, ФСД и параметры генерации сетки (рис. 2).

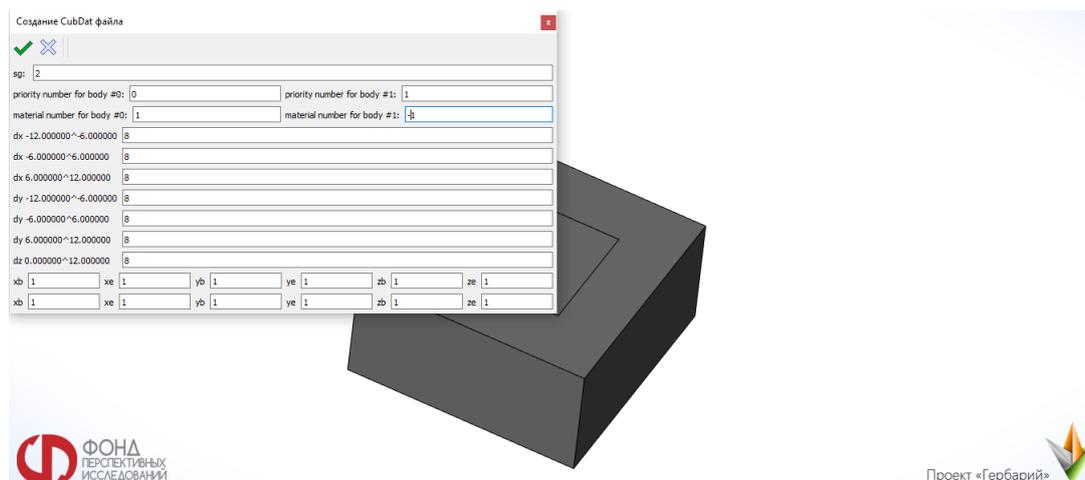


Рис. 2. Параметры генерации

3.2. Генерация сетки. Модуль «Делоне»

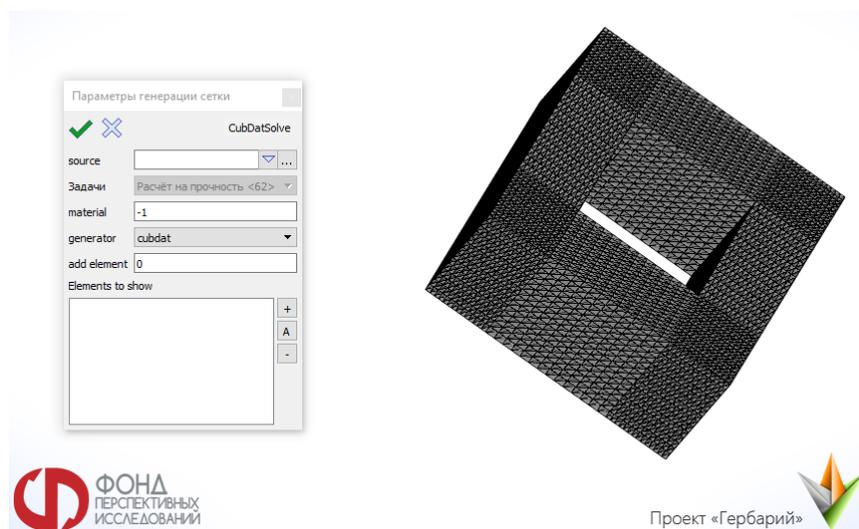


Рис. 3. Сеточная модель

Указав источник генерации сетки, получим визуальное представление построенной сеточной модели (рис. 3). Далее вызываем программу построения решения CubDatSolve (модуль «Чебышёв»). Визуализация полученного решения в данный момент находится в разработке.

Заключение

В настоящей работе модель управления вычислительным комплексом на основе семантической сети. Узлы этой семантической сети описаны и классифицированы по трем типам семейств: «Вороной», «Делоне», «Чебышёв». Такой способ опробован на программном комплексе для решения трехмерных задач математической физики на основе концепции БСМ [7]. В дальнейшем предполагается расширение этой сети путем добавления новых узлов в каждую подсеть.

Список литературы

1. Ильин В. П. Фундаментальные вопросы математического моделирования // Вестн. Российской Академии Наук. 2016. Т. 86, № 4. С. 26–36.
2. Голубева Л. А., Ильин В. П., Козырев А. Н. О программных технологиях в геометрических аспектах математического моделирования // Вестн. НГУ. Серия: Информационные технологии. 2012. Т. 10, № 2. С. 25–33.
3. Ильин В. П., Гладких В. С. Базовая система моделирования (БСМ): концепция, архитектура и методология «Современные проблемы математического моделирования, обработки изображений и параллельных вычислений 2017»(СПММОИиПВ-2017). Ростов-на-Дону: Изд-во ДГТУ, 2017. С. 151–158.
4. URL: <http://tflex.ru/about/publications/detail/index.php?ID=3846>.
5. Бутюгин Д. С., Гурьева Я. Л., Ильин В. П. и др. Функциональность и технологии алгебраических решателей в библиотеке Krylov // Вестн. ЮУрГУ. Серия: Вычислительная математика и информатика. 2013. Т. 2, № 3. С. 92–105.
6. Gurieva Ya. L., Il'in V. P. Program package for 3D boundary-value elliptic Problem // Bull. NCC. Series: Num. Anal. 2002. Iss. 11. P. 35–52.
7. Голубева Л. А., Горшун В. С., Ильин В. П., Эрдыниева Э. Б. Программный комплекс для решения 3-мерных задач математической физики на основе концепции БСМ // Труды Междунар. конф. «Вычислительная математика и математическая геофизика», посвящ. 90-летию со дня рождения акад. А. С. Алексеева. 2018. С. 126–132.
8. Basic Research Needs Workshop for Scientific Machine Learning Core Technologies for Artificial Intelligence. Prepared for Department of Energy Advanced Scientific Computing Research. 2019. Vol. February, 10.
9. URL: <https://ngsolve.org/docu/latest/>.

UDK 519.632

THE MANAGEMENT MODEL OF THE COMPUTING COMPLEX, IMPLEMENTED ON THE BASIS OF THE CONCEPT OF BCM

L. Golubeva

Ph.D., associate professor, researcher, Institute of Computational Mathematics and Mathematical Geophysics of SB RAS, 630090, Novosibirsk, Russia, Acad. Lavrentieva pr., 6, Novosibirsk State University, 630090, Novosibirsk, Russia, Pirogova str., 1, e-mail: golubeva@labchem.sccc.ru

V. Gorshunov

Engineer, Institute of Computational Mathematics and Mathematical Geophysics of SB RAS, 630090, Novosibirsk, Russia, Acad. Lavrentieva pr., 6, graduate student, Novosibirsk State University 630090, Novosibirsk, Russia, Pirogova str., 1, e-mail: basil.gorshunov@gmail.com

V. Il'in

Ph.D., Professor, chief researcher, Institute of Computational Mathematics and Mathematical Geophysics of SB RAS, 630090, Novosibirsk, Russia, Acad. Lavrentieva pr., 6, Novosibirsk State University, 630090, Novosibirsk, Russia, Pirogova str., 1, e-mail: ilin@sccc.ru

Annotation. A computer complex based on the concept of a basic modeling system (BSM) includes three important modules for the construction of a geometric model, a grid model and an algebraic equation solver. The purpose of this work is to develop a model for managing this computing complex using a knowledge base presented in the form of a semantic network.

Keywords: Mathematical physics, geometric data structure, grid data structure, functional data structure, BCM, semantic networks, knowledge bases

References

1. *Il'in V. P.* Fundamental'nye voprosy matematicheskogo modelirovanija // Vestn. Rossijskoj Akademii Nauk. 2016. T. 86, № 4. P. 26–36. (in Russian)
2. *Golubeva L. A., Il'in V. P., Kozyrev A. N.* O programmnyh tehnologijah v geometricheskikh aspektah matematicheskogo modelirovanija // Vestn. NGU, Serija : Informacionnye tehnologii. 2012. T. 10, № 2. P. 25–33. (in Russian)
3. *Il'in V. P., Gladkih V. S.* Bazovaja sistema modelirovanija (BSM): koncepcija, arhitektura i metodologija “Sovremennye problemy matematicheskogo modelirovanija , obrabotki izobrazhenij i parallel'nyh vychislenij 2017”(SPMMOIiPV-2017). Rostov-na-Donu : Izd-vo DGTU, 2017. P. 151–158. (in Russian)
4. URL: <http://tflex.ru/about/publications/detail/index.php?ID=3846>.
5. *Butjugin D. S., Gur'eva Ja. L., Il'in V. P. et al.* Funkcional'nost' i tehnologii algebraicheskikh reshatelej v biblioteke Krylov // Vestn. JuUrGU. Serija : Vychislitel'naja matematika i informatika. 2013. T. 2, № 3. P. 92–105. (in Russian)
6. *Gur'eva Ya. L., Il'in V. P.* Program package for 3D boundary-value elliptic Problem // Bull. NCC. Series: Num. Anal. 2002. Iss. 11. P. 35–52.
7. *Golubeva L. A., Gorshunov V. S., Il'in V. P., Jerdyniev Je. B.* Programmnyj kompleks dlja reshenija 3-mernyh zadach matematicheskoy fiziki na osnove koncepcii BSM // Trudy Mezhdunar. konf. “Vychislitel'naja matematika i matematicheskaja geofizika” posvjashhennaja 90-letiju so dnja rozhdenija akad. A. S. Alekseeva. 2018. P. 126–132. (in Russian)
8. Basic Research Needs Workshop for Scientific Machine Learning Core Technologies for Artificial Intelligence. Prepared for Department of Energy Advanced Scientific Computing Research. 2019. Vol. February, 10.
9. URL: <https://ngsolve.org/docu/latest/>.

О классификации парадигм и языков программирования

Л. В. Городня

канд. физ.-мат. наук, ст. науч. сотр.,
Институт систем информатики им. А. П. Ершова СО РАН,
630090, Новосибирск, Россия, пр. Акад. Лаврентьева, 6,
доц., Новосибирский государственный университет,
630090, Новосибирск, Россия, ул. Пирогова, 1, e-mail: lidvas@gmail.com

Аннотация. Статья представляет результаты анализа наиболее известных парадигм программирования и подход к навигации в современном расширяющемся пространстве языков программирования, основанный на классификации парадигм по особенностям постановок задач и парадигмальную характеристику языков программирования в терминах семантических систем. Работа выполнена при поддержке Российского фонда фундаментальных исследований, проект № 18-07-01048-а.

Ключевые слова: парадигмы программирования, семантические системы, понятийная сложность программирования, реализационная прагматика, определение языков программирования, декомпозиция определений, критерии декомпозиции

Введение

К настоящему времени в разных источниках упоминается от двадцати до семидесяти парадигм программирования (ПП), поддержанных десятками тысяч языков программирования (ЯП), число которых быстро растет, особенно в области проблемно ориентированных языков [1]. Для нового ЯП обычно указывают несколько повлиявших на него предшественников и пять-семь поддержанных ПП [2]. Такие, бесспорно важные, данные не обладают точностью для понимания места ЯП, его потенциала и сферы успешного применения. Остаются без ответа вопросы:

- почему стремительно растет число ЯП?
- как обосновывать выбор ЯП для решения конкретной задачи?
- можно ли оценить зависимость трудоёмкости программирования от выбора ЯП?

В данной статье рассматривается несколько форм, нацеленных на решение проблемы навигации в расширяющемся пространстве ЯП с помощью представления семантических систем, примеров программ и численных характеристик используемых и создаваемых средств и методов. Предполагается, что, используя такие формы, можно выстроить метрическое пространство, полезное при измерении понятийной сложности конструкций, поддержанных в определениях ЯП, а также информационных систем и компьютерных средств. Такое пространство может быть применено при сравнении ПП, потенциала используемых при разработке программ схем и моделей, оценки уровня новизны создаваемых ЯП, а также при выборе критериев декомпозиции программ для накопления типовых программных компонентов, нацеленных на смягчение перехода от доминирующей парадигмы императивно-последовательного программирования к парадигмам организации асинхронных параллельных взаимодействующих процессов над общей памятью.

1. Результаты анализа постановок задач и парадигм программирования

Анализ и сравнение ряда характеристик наиболее известных ПП показывает их зависимость от особенностей описания постановок задач, от исторически сложившихся подходов к решению задач и уровня их изученности. Для показа особенностей ПП обычно выделяются концептуальные монопарадигмальные ЯП и приводятся критерии успешного применения ПП с оценкой результатов на примерах программ, что подтверждает признание практикой программирования.

Переход к парадигмальной характеристике ЯП, поддерживающих более одной ПП, требует выделения монопарадигмальных подязыков в его определении и описания схемы постановок целевых задач. Уровень реализационной прагматики системы программирования (СП) для анализируемого ЯП подчинен требованиям эффективности и надежности кода программы, обычно препятствующим сохранению исходной структуры определения. Поэтому при парадигмальном анализе ЯП возникает необходимость в механизмах независимого представления семантических систем, позволяющих демонстрировать расслоение средств ЯП желательно на простых примерах отлаженных программ с комментариями.

Парадигма программирования как образ мышления связана с компромиссом между особенностями решаемых задач и методами их решения с помощью программ. Постановки задач существенно различаются, кроме разницы между научно-исследовательскими и производственными задачами, по следующим направлениям:

- стабильная постановка задачи, решаемой с помощью конечного автомата;
- развивающаяся постановка задачи, требующая для решения расширяемого автомата;
- задача бесперебойного функционирования многопроцессорной конфигурации, решаемая как комплекс взаимодействующих процессов;
- усложненные задачи, связанные с трудно достоверяемыми критериями качества обработки большеобъемных массивов, обеспечения надежности и безопасности информационного обслуживания.

Заметно, что большинство ПП являются сужениями или обобщениями четырех-шести базовых ПП, называемых **основными** в сфере производственной практики, реализующей достаточно изученные решения востребованных задач, или **фундаментальными** в академических кругах, решающих новые задачи и исследующих полноту их решения. Внешне разница сводится к приоритетам в оценке важности понятий ПП для решения задач разработки программ или изучения методов программирования. Это позволяет дать формальное описание типовых категорий семантических систем, характерных для поддержки конкретных **базовых** ПП. Описанию **производных** ПП в таком случае можно сопоставлять нечто вроде семантического индекса, отражающего разницу с базовой ПП. К базовым парадигмам имеет смысл относить императивно-процедурное (ИПП), функциональное (ФП), логическое (ЛП) и объектно-ориентированное (ООП) программирование, а также параллельные вычисления (ПВ).

Постановки задач ИПП начинаются с определенного алгоритма решения актуальной задачи. Необходимо получить программу реализации алгоритма с практичными пространственно-временными характеристиками на доступном оборудовании. Производные ИПП выделяют разные методы представления и организации порождаемых программой последовательных процессов:

- автоматное — без процедур;
- скалярное — без структур данных;
- структурное — регулярная логика и типы данных;
- сборочное — крупноблочность;
- диаграммно-графовое — визуальность;
- сценарное — одноуровневое, поверхностное;
- операционное — управление заданиями;

- языки действий — управление акторами;
- векторно-ориентированные — обработка массивов;
- синхронизация над общей памятью и т. п.

Постановка задач ООП основана на доступе к иерархии классов объектов с готовыми работоспособными методами решения ряда задач некоторой сферы приложения ИТ. Требуется без лишних трудозатрат уточнить эту иерархию, чтобы приспособить ее к решению новых востребованных задач этой области, ее расширения или ей подобной. Производные ООП дают разнообразные конкретизации понятия «класс объектов»:

- аспектно-ориентированное — разделение слоев (вертикальное слоение А. Л. Фуксмана);
- прототипная — без классов;
- табличная — с таблицами в роли классов;
- обобщенная – с АД в роли классов;
- DSL — языково-ориентированная настройка над gсс;
- субъектно-ориентированное — объекты с поведением;
- агентно-ориентированное — выделение исполнителей;
- сервис-ориентированное — выделение обслуживания и т. п.

Задачи ФП ориентированы на известную предметную область, в рамках которой следует выбрать символьное представление данных и отладить систему универсальных функций, пригодных для создания прототипов решения актуальных задач из этой области. Производные ФП представляют вариации в методах организации вычислений:

- ленивые вычисления — откладывание действий;
- комбинаторное — сведение к функциям;
- гомоиконное — подобию структур данных и обрабатывающих программы;
- аппликативное — подгрузка любого базиса;
- потоковый параллелизм — множество результатов;
- правило переписывания — подстановка;
- бесточечное — форма записи;
- алгебраическое — выделение семантических систем;
- инсерционное — динамические вставки-замены (точки роста С. С. Лаврова);
- грамматико-ориентированная обработка (BNF);
- стек-ориентированное (Forth) и проч.

Решения задач ЛП исходят из заданной коллекции фактов и отношений, характеризующей актуальную задачу, которая пока не имеет эффективного решения. Надо привести эту коллекцию, возможно пополнить, к форме, демонстрирующей возможность ответов на практические запросы относительно данной задачи. Производные ЛП используют разные подходы к смягчению зависимости получения результатов от избыточного или недостаточного детерминизма:

- откаты-бэктрекинг — перебор вариантов;
- программирование в ограничениях — границы определенности;
- представление знаний — привлечение экспертов;
- вопрос-ответное — задачи поиска;
- индуктивное — вывод решения;
- естественно-языковое — представление запросов на языковом подмножестве;
- недетерминированное — без преждевременного упорядочения и др.

Постановки задач ПВ используют доступные программы решения конкретной задачи, скорость получения результатов по которым недостаточна. Необходимо построить решение в

виде семейства процессов, дающее заметный выигрыш в скорости. Определение парадигм этого направления находится в стадии формирования.

Кроме того, существует заметное число комбинированных ПП, нацеленных на сложные постановки задач с разнотипными подзадачами, поддерживаемые и мультипарадигмальными ЯП (Lisp 1.5, Planner, Merlin, F#, C#, Scala и др.). Известны отвергнутые ПП, не получившие признания программистским корпусом, несмотря на авторитетную формулировку принципов.

Редко упоминаются достаточно важные вспомогательные **системообразующие** ПП, существование которых нацелено на общий прогресс программирования, решение его собственных задач, инструментальную поддержку программирования как профессии. Эксплуатационная прагматика таких ПП, в отличие от парадигм прикладного программирования, требует создания специальной обстановки, включая элементы образовательной системы и комплекс мероприятий по проявлению и накоплению знаний и по обмену опытом. Результативность такой обстановки ярко показана Ершовскими школами юных программистов, ежегодно проводимыми ИСИ СО РАН. Постановки задач здесь нацелены на улучшение программистского корпуса, эстафету программистского знания и совершенствование технологий программирования. Следует отметить достаточно сложившиеся системообразующие ПП:

- системное программирование (СП, БД, ОС);
- оптимизационное программирование;
- метапрограммирование (Рефал);
- трансформационное программирование (смешанные вычисления);
- учебное программирование;
- экспериментальное программирование;
- олимпиадное программирование;
- теоретическое программирование — раздел дискретной математики;
- доказательное программирование;
- компонентное программирование;
- непрофессиональное программирование (Basic);
- генетическое программирование — вероятность.

В последние годы проявляются основания обуславливать конструирование средств верификации программ формализацией используемых ПП, а программистские проекты сопровождать обоснованием выбора не только инструментария, но и ПП, чтобы избежать межпарадигмальных конфликтов, чреватых трудно уловимыми ошибками.

2. Семантические модели

Следует отметить при любом анализе зависимость понимания смысла от владения оттенками профессиональной терминологии и вариациями определения понятий. Современная практика программирования имеет тенденцию к использованию примеров текстов или шаблонов программ без выделения понятийной структуры. Желательно, чтобы такие тексты допускали ясное сопоставление и сведение к общим интуитивным понятиям. Наиболее объективные понятия такого рода связаны с архитектурными моделями, с методами реализации ЯП и с классификацией решаемых задач.

В качестве отправной точки для выстраивания пространства принятия решений по декомпозиции программ можно принять классическое понятие «алгебраическая система», расширенное С.С. Лавровым до понятия «семантическая система» заданием явного правила применения функций к значениям [3].

Алгебраическая система — это $\langle A, F \rangle$, где:

A — основное множество значений,

F — конечный набор операций над множеством A.

Примерно так можно формализовать библиотечные модули в СП и классы объектов в ООП до тех пор, пока при реализации ЯП не возникает необходимость использовать разные модели вычислений по одним и тем же формулам.

Семантическая система — это $\langle V, F, R \rangle$, где:

V — основное множество значений,

F — конечный набор функций над значениями из основного множества,

R — правило применения функций к значениям.

Наиболее известные компьютерные архитектуры обычно представляют как конструкцию из вычислительного устройства (E), памяти (M), устройства управления процессами (C) и средств коммуникации между устройствами и их элементами (S). Это дает первое приближение для выбора основных семантических систем в языках программирования [3]. Рассматривая любые семантические системы, важно отметить разницу в характере выполнения эффективно реализуемых функций таких систем.

Для любого множества значений **V** реализационно различимы категории функций для методов вычислений **E**: $(V^* \rightarrow V^+)$, средств доступа к памяти **M**: $(T : N \rightarrow V)$, особенностей управления вычислениями **C**: $(F \rightarrow \{0, 1\})^*$ и обратимой комплексации данных **S**: $(A \leftrightarrow K)$. При сравнительном анализе языков программирования может быть использована понятийная матрица, представленная в таблице, в которой RE — обычные арифметические вычисления, отображающие произвольный ряд значений аргументов в не менее чем один результат, RM — символьные вычисления, подставляющие представления аргументов без их предварительного вычисления, RC — частичные или смешанные вычисления, лавирующие между вычислением и подстановкой в зависимости от разных условий, RS — обобщенные и параллельные вычисления, оперирующие организацией процессов как множеством потоков над комплексами из разных устройств.

Понятийная матрица для сравнительного анализа ЯП

Категория семантической системы	Константы	Вычисления	Хранение	Процессы	Структуры
Виды функций	V	FE: $V^* \rightarrow V^+$	FM: $(T : N \rightarrow V)$	FC: $\{F \rightarrow \{0, 1\}\}$	FS: $A \leftrightarrow K$
RE: Ядро	Значение	Операции	Память	Управление	Вектор
RM: Макро	Данное	Функции	Задание	Блоки	Стек
RC: Границы	Исключения	Предикаты	Типы данных	Логика	Варианты
Общность	Неопределенность	Мультиопера- ции	Внешний мир	Отображени- я	Ввод-вывод

Ядро — семантический базис ЯП. Полное определение языка можно получать как консервативное расширение его ядра. Обычно ядро приспособлено и к неконсервативному расширению пополнением набора библиотечных функций, реализуемых на уровне аппаратуры. Это позволяет в реализации систем программирования для любого ЯП поддерживать разные парадигмы программирования, необходимые для поддержки полного жизненного цикла программ, чтобы достигать результата независимо от исходных возможностей ЯП.

Макро — пополнение ядра средствами обработки представлений, используемых с целью укрупнения любых конструкций, что позволяет выполнять консервативное расширение ЯП. Кроме того, оно способствует лаконизму текстов программ. Макротехника позволяет наследовать отлаженность фрагментов программ. Бывает важным исключать дубли частей

текста, кода и структур данных. Простейший механизм макрогенерации обычно присутствует в СП как препроцессор. Также бывает устроена техника кодогенерации и обработки шаблонов при компиляции программ. Реализация укрупнений может быть функционально эквивалентна вызову подпрограмм. Взаимозаменяемость макроподстановки и вызова подпрограмм нередко используется при оптимизации программ.

Границы — методы проверки вычислимости функций и выполнимости заданий. Цель представления границ — снижение трудоемкости отладки программ упрощением поиска ошибок. При отсутствии ошибок проверка воспринимается как накладные расходы. Встречаются механизмы установки ловушек на нежелательные ситуации и программирования обработки исключений с возможностью продолжения вычислений.

Общность — дополнительные средства обеспечения отладки, **улучшения** и применения программ, поддерживающие возможность разумного продолжения вычислений при любых исходных данных, возможно неподходящих, и неожиданных или аварийных ситуациях.

Такую таблицу можно использовать как критерии для декомпозиции определения ЯП на разные категории семантических систем, допускающие их сравнение в рамках монопарадигмальных подязыков [4]. Парадигмальная характеристика ЯП может быть представлена в виде перечня заполненных клеток понятийной таблицы.

Заключение

Описанные решения можно рассматривать как конкретизацию понятийной сложности по Колмогорову [5]. Близкие работы начаты еще в середине 1960-х годов, когда возникла задача определения ЯП, поддерживающих процессы разработки СП при создании новых машин и расширении сфер приложения программ. В те годы был выполнен ряд серьезных попыток создания языков системного программирования, позволяющих конструировать системы программирования [6,7]. Предложенная формализация может быть полезной при оценке сложности и трудоемкости программирования, особенно при учете разделения требований к постановкам задач по сферам применения на академические и производственные, а по уровню изученности – на четкие, развиваемые и усложненно трудоемкие.

Список литературы

1. Computer Languages History. URL : <https://www.levenez.com/lang/>.
2. Энциклопедия языков программирования. URL : <http://progopedia.ru/>.
3. *Лавров С. С.* Методы задания семантики языков программирования // Программирование. 1978. № 6. С. 3–10
4. *Городняя Л. В.* О представлении результатов анализа языков и систем программирования // Научный сервис в сети Интернет: труды XX Всеросс. науч. конф. (17–22 сентября 2018 г., г. Новороссийск). М. : ИПМ им. М. В. Келдыша, 2018.
5. *Колмогоров А. Н.* Три подхода к определению понятия «количество информации» // Проблемы передачи информации. 1965. № 1 (1). С. 3–11.
6. *Фуксман А. Л.* Технические аспекты создания программных систем. М. : Статистика, 1979. 180 с.
7. *Koster Cornelis H. A.* Compiler Description Language. CDL3 manual // The Netherlands. 2004. August 18. URL : <http://www.cs.ru.nl/cdl3/cdl3.pdf>.

PAPER FORMATTING IN MICROSOFT WORD FORMS OF REPRESENTATION OF RESULTS OF PARADIGMAL ANALYSIS OF PROGRAMMING LANGUAGES

L. V. Gorodnyaya

Ph.D., associate professor, researcher A. P. Ershov Institute of Informatics Systems,
Siberian Branch of the Russian Academy of Sciences,
630090, Novosibirsk, Russia, Acad. Lavrentjev pr., 6,
Novosibirsk State University,
630090, Novosibirsk, Russia, Pirogova str., 1, e-mail: lidvas@gmail.com

Annotation. The purpose of the article is the choice of presenting the results of a comparison of programming languages, convenient for assessing the expressive power of languages and the complexity of the implementation of programming systems. Forms of such a presentation should be adapted to substantiate practical criteria for program decomposition, which can be viewed as an approach to solving the problem of factorization of very sophisticated definitions of programming languages.

Keywords: programming systems, program decomposition, decomposition criteria, semantic systems, implementation pragmatics, programming paradigms, definition of programming languages

References

1. Computer Languages History. URL : <https://www.levenez.com/lang/>.
2. Encyclopedia of programming languages. URL : <http://progopedia.ru/>. (in Russian)
3. Lavrov S. S. Methods for specifying the semantics of programming languages // Programming. 1978. № 6. P. 3–10. (in Russian)
4. Gorodnyaya L.V. Paradigms of programming: analysis and comparison / Rus. Acad. of Sciences, Sib. Branch, A. P. Ershov Institute of Informatics Systems. Novosibirsk : Izdat. SB RAS, 2017. 232 p. (in Russian)
5. Kolmogorov A. N. Three approaches to the definition of “amount of information” // Problems of information transfer. 1965. № 1 (1). P. 3–11. (in Russian)
6. Fuksman A. L. Technical aspects of creating software systems. M. : Statistics, 1979. 180 p. (in Russian)
7. Koster Cornelis H. A. Compiler Description Language. CDL3 manual // The Netherlands. 2004. August 18. URL: <http://www.cs.ru.nl/cdl3/cdl3.pdf>.

Разработка МАС для имитации процесса реализации проекта трансконтинентальной магистрали через Берингов пролив (моделирование информационного пространства)

Т. Н. Есикова

канд. экон. наук., ведущий науч. сотр.,

Институт экономики и организации промышленного производства СО РАН,
630090, Новосибирск, Россия, пр. Акад. Лаврентьева, 17, e-mail: T.N.Yesikova@gmail.com

С. В. Вахрушева

аспирант, Новосибирский государственный университет,

630090, Новосибирск, Россия, ул.Пирогова, 1, e-mail: s.vakhr@gmail.com

Аннотация. В работе ставится проблема моделирования процессов, сопряженных со реализацией крупномасштабных инфраструктурных проектов на примере трансконтинентальной магистрали через Берингов пролив (развитие транспортной сети на территориях Крайнего севера). Исследование проводится на базе такого метода имитационного моделирования как мультиагентное моделирование. В статье приводится описание основных характеристик информационной среды взаимодействия агентов системы применительно к данной задаче, требований к разработке протоколов взаимодействия со средой в рамках МАС и другие аспекты проектирования окружения агентов в соответствии с процессами, протекающими в реальном мире.

Ключевые слова: мегапроекты, мультиагентные системы (МАС), агенты, среда функционирования агентов, архитектура МАС

Введение

Проектирование и реализация крупномасштабных инфраструктурных проектов сопряжена с пересечением интересов экономических акторов различной природы (в том числе, социальных групп), имеющих разный вес и обладающих разными уровнями влияния при принятии окончательных решений. В связи с этим представляет особый интерес такой инструмент имитационного моделирования, как мультиагентный подход [1–8], который позволяет не только детально представить каждого экономического актора (с его интересами, механизмами влияния на управленческие решения, особенностями поведения и др.), но и саму гетерогенную среду окружения.

Особое место в этом контексте занимает моделирование информационного пространства [9], предопределяющего аналитическую основу формирования решений акторами разной природы (экономическими, управленческими и др.). Это обусловило необходимость отдельного рассмотрения данного этапа проектирования мультиагентной системы для имитации различных вариантов реализации инфраструктурных мегапроектов.

1. Некоторые аспекты разработки мультиагентной системы

Предметная область данного исследования связана с крупномасштабными транспортными проектами – акцент сделан на проект строительства трансконтинентальной магистрали (ТКМ) через Берингов пролив.

В качестве метода исследования выступает такой инструмент имитационного моделирования как мультиагентный подход, который заключается в имитации процесса взаимодействия интеллектуальных агентов. Мультиагентное моделирование позволяет отразить неоднородность, уникальность, многообразие и динамику взаимодействия конкретных экономических акторов, а значит и структуру моделируемого процесса наиболее приближенно к реальности.

Специфика выбранного метода исследования предопределяет идентификацию представительного множества акторов в качестве агентов и декомпозицию общего процесса реализации проекта на ключевые компоненты [1].

В общем случае модель разрабатываемой МАС имеет вид [2]:

$M = (V_n, E_m, I_k, P_l, R_s)$, где

V – множество различных групп агентов. На текущем этапе исследования выделены следующие типы агентов: представители высших органов власти; руководители территориальных субъектов в зоне реализации проекта, эксперты, инвесторы, подрядные организации, поставщики оборудования и общественные организации. $|V| = n = 7$;

$E \subset V \times V$ – множество различных видов отношений между агентами (ключевых подпроцессов), например, согласование проекта ТКМ (в том числе, при внесении изменений в проект); взаимодействие власти и компаний-инвесторов; проведение тендеров по отдельным этапам работ по проекту, закупка оборудования и т. д. $|E| = m = 6$;

$I_k = I_field_p \cup I_flow_q$ – множество компонентов внешней информационной среды функционирования агентов – информационного пространства, которое состоит из информационных полей ($I_field_p, |I_field| = p$) и информационных потоков ($I_flow_q, |I_flow| = q$), в роли которых могут выступать, к примеру, СМИ; $|I| = k = p + q$;

$P \subset V \times I$ – множество взаимосвязей агентов и внешней среды их функционирования (восприятие агентами преобразований среды, формирование изменений в информационных потоках) $|P| = l$;

$R \subset I \times V$ – множество взаимосвязей среды и агентов (оповещение агентов информационной средой о каком-либо событии, изменение атрибутов агентов и т. д.) $|R| = s$.

Поскольку при протекании процессов, связанных с развитием транспортной сети, информационное пространство занимает особое место (т.к. влияет на исходы взаимодействий различных участников этих процессов), рассмотрим более подробно некоторые аспекты проектирования окружения агентов на примере реализации проекта ТКМ через Берингов пролив.

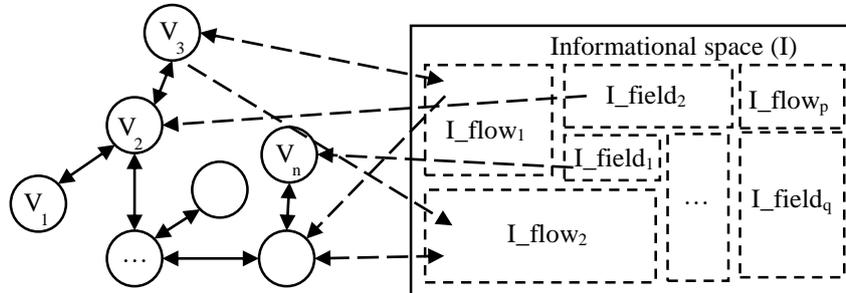
2. Построение концептуальной модели информационного пространства

В рамках разрабатываемой МАС окружение агентов (I) – абстрактное представление агрегации различных составляющих информационного пространства (как информационных полей, так и потоков) [10, 11]. В контексте программной системы это база данных, причем под информационными полями (I_field_p) понимаются данные, в которые не вносятся изменения агентами (это базы знаний о погодных условиях региона; справочники, содержащие информацию о составе тех или иных строительных работ, информацию о существующих транспортных проектах и т. д.); а под информационными потоками (I_flow_q) – данные, которые формируют и изменяют агенты (сообщения СМИ о событиях, связанных с реализацией проекта, сообщения иностранных СМИ, упоминания о проекте в социальных медиа, информация о ходе строительства и др.) [11]. Таким образом, в каждый момент времени один и тот же информационный поток может содержать различную информацию.

В реальном мире перечисленные объекты информационных пространства формируют специалисты в различных областях, представители определенных профессий, лидеры мнений и т. д. [10, 11], но для моделирования хода реализации проекта существенными для модели свойствами данные субъекты не обладают, с участниками процесса напрямую не

взаимодействуют, а имеет значение лишь результат их работы – производимые ими артефакты, которые и будут являться компонентами модели данных информационной среды разрабатываемой МАС.

На рисунке представлена схема взаимодействия агентов (V_n) и информационного пространства (Informational space, I), логически разделенного на области ($I_field_1, \dots, I_field_p; I_flow_1, \dots, I_flow_q$).



Взаимодействие агентов и информационной среды в моделируемой МАС

В данном случае агенты являются как потребителями, так и производителями информации, но при этом каждый тип агентов имеет доступ лишь к определенной ее части, в то же время не все области информационного пространства могут быть подвержены изменениям со стороны агентов.

Как и в реальном мире, информационное пространство влияет на исход взаимодействий между различными агентами – участниками реализации проекта, поскольку может содержать как справочную информацию, на основе которой агенты принимают решения, так и срочные сообщения, к примеру, информацию о произошедшей аварии и т. д.

Пример: Агент V_1 (представители высших органов власти) имеет среди своих атрибутов ($a_{11} \dots a_{1x}$) атрибут a_{12} , с помощью которого выражается лояльность агента к проекту. Информационный поток I_flow_3 (иностранные СМИ) оповещает агентов о появлении транспортного проекта, подобного реализуемому (справочник I_field_6). Информационное поле I_field_6 в контексте МАС – уже имеющая значения таблица базы данных, столбцами которой являются: project_name (наименования транспортных проектов) и countries (ссылки на страны из I_field_{11} , которые положительно относятся к соответствующему проекту). В свою очередь, I_field_{11} – также заполненная таблица базы данных, столбцами которой являются: country (название страны) и relation (показатель влияния страны на мнение агентов, $[0, \dots, 100]$). Таким образом, при появлении сообщения от I_flow_3 агент V_1 инициирует процесс формирования оценки нового транспортного проекта: считывает список стран, оставивших мнение о данном проекте из I_field_6 , а затем вычисляет интегральный критерий, выражающий вес нового проекта в мире с учётом показателя влияния этих стран на агентов (из I_field_{11}): $j = \sum_{i=1}^{countries} relation_i$. После этого V_1 суммирует значение a_{12} с j . Каждый раз при смене a_{12} агент фиксирует значение в I_flow_6 – информационном потоке, содержащем информацию о лоббистах и противниках проекта, разработку которого имитируют агенты.

В силу того, что в некоторых случаях и при формировании и восприятии информации речь идёт о субъективном взгляде на те или иные события, несвоевременности получения каких-либо сведений, а также о других влияниях человеческого фактора, было принято при некоторых случаях восприятия информации агентами искусственно внести некоторое ее искажение (по аналогии с реальными процессами: агент неверно интерпретирует считанную информацию или ошибается при ее обработке). Вероятность возникновения искажений будет варьироваться [9].

3. Основные признаки информационной среды

Среды взаимодействия агентов в МАС можно классифицировать по различным параметрам [12]:

- по характеру восприятия среды агентами среды делятся на дискретные и непрерывные;
- по характеру проявляемой агентом реакции на среду – на детерминированные и не детерминированные;
- по признаку изменяемости среды за время между восприятием агентами преобразования среды и проявлением их реакции на данное преобразование – на статические и динамические.

С позиции описанных выше факторов, в рамках данного исследования разрабатывается дискретная недетерминированная динамическая среда. Иными словами, предполагается, что: а) агенты воспринимают события с достаточно высокой, но ограниченной точностью (информация, воспринятая агентом может отличаться от информации, сформированной средой), поскольку имеет место искусственное искажение информации; б) в ответ на преобразование среды агент формирует неограниченное число реакций – от инициирования взаимодействия с различными другими агентами (или же формирования изменений в другой области информационной среды) до отсутствия какой-либо реакции; в) в момент между восприятием изменения среды и реакцией агента на это изменение информационная среда может преобразоваться вновь. Проектирование среды, обладающей данными характерными чертами, обусловлено целью достижения наибольшего сходства модели с процессами, протекающими в реальном мире.

4. Разработка протоколов взаимодействия агентов со средой

Агенты обладают сенсорами (датчиками) для восприятия окружающей среды и впоследствии изменяют её (реагируют) при помощи эффекторов (исполнительных органов), а значит, важной задачей также является определение способов восприятия агентами изменений информационной среды (и обратно) [17]. Таким образом, существенными аспектами разработки информационной среды являются:

- 1) определение границ чувствительности каждого агента;
- 2) рассмотрение возможности управления режимами работы сенсоров каждого агента;
- 3) определение искажений при восприятии информации из окружающей среды;
- 4) определение характера и степени влияния среды на агентов;
- 5) определение характера и степени влияния агентов на среду.

На данном этапе исследования поставлена цель анализа задачи с позиции этих аспектов и последующей разработки протоколов взаимодействия агентов с их общей информационной средой, которые содержат указания порядка доступа к общему пространству данных. Наблюдается очевидная связь с классическими подходами обеспечения мультидоступа к данным, в связи с этим настоящие методы послужат основой для разработки коммуникационных протоколов для обеспечения передачи информации между средой и агентами [13-16]. В случае, когда агент вносит изменения в общую информационную среду, на данные накладываются ограничения (по атомарности операций, производимых агентами; по сериализуемости расписаний доступа к данным; по консистентности данных, но не во всех областях одновременно и по изолированности вносимого изменения). В различных информационных областях пространства, которые являются прототипом информационных потоков (например, сообщений СМИ), консистентность данных не запрашивается, так как в реальном мире сведения об одном и том же событии из разных источников также могут отличаться и быть не согласованными.

Заключение

На данном этапе исследования предложен прототип мультиагентной системы для моделирования процессов, сопряженных с реализацией транспортных мегапроектов. Особое внимание уделено описанию информационной среды функционирования агентов, определению состава источников данных для среды, а также способов формирования и изменения ее компонентов агентами и обратно, методов воздействия среды на агентов.

В перспективе стоит задача уточнения агентной модели: состава агентов, алгоритмов их взаимодействия, развития топологии МАС, а также усовершенствования прототипа разрабатываемой информационной среды. Кроме того, представляется необходимой разработка коммуникационных протоколов для организации корректного взаимодействия агентов и их общего информационного пространства.

Работа выполнена по плану НИР ИЭОПП СО РАН N АААА-А17-117022250123-0.

Список литературы

1. *Лычкина Н. Н.* Информационные системы управления производственной компанией: учебник и практикум для академического бакалавриата. М. : Изд. Юрайт, 2018.
2. *Жмурко С. А.* Обобщенная модель агента и многоагентной системы // Известия ЮФУ. Технические науки. 2008. № 4.
3. *Городецкий В. И., Грушинский М. С., Хабалов А. В.* Многоагентные системы // Новости искусств. интеллекта. 1998. № 2. С. 64–116.
4. *Хорошевский В. Ф.* Методы и средства проектирования и реализации мультиагентных систем // Матер. семинара «Проблемы искусственного интеллекта». ИПУ РАН, 1999.
5. *Маслобоев А. В.* Механизмы взаимодействия и координации агентов в открытой мультиагентной системе информационной поддержки региональных инновационных структур. Теория и практика системной динамики : труды II-й Всерос. науч. конф. Апатиты : КНЦ РАН, 2007.
6. *Маслобоев А. В.* Мультиагентная технология информационной поддержки инновационной деятельности в регионе. Прикладные проблемы управления макросистемами : труды Ин-та системного анализа РАН. М. : КомКнига УРСС, 2009. Т. 39.
7. *Замятина Е. Б.* Современные теории имитационного моделирования : спец. курс. Пермь : ПГУ, 2007
8. *Аристов С. А.* Имитационное моделирование экономических процессов : учеб. пособие. Екатеринбург, 2003.
9. *Пантелеев М. Г., Кохтенко Н. В., Лебедев С. В.* Среда имитационного моделирования агентных систем реального времени // Научно-технический вестник информационных технологий, механики и оптики. 2012. № 1 (77).
10. *Чайковский Д. В.* Информационное пространство: анализ определений // Вестн. БГУ. 2010. № 14.
11. *Манойло А. В.* Государственная информационная политика в особых условиях: Монография. М. : МИФИ, 2003.
12. *Девятков В. В.* Системы искусственного интеллекта: учеб. пособие для вузов. М. : Изд-во МГТУ им. Н.Э. Баумана, 2001.
13. *Дейт К. Дж.* Введение в системы баз данных, М. : Изд.«Вильямс», 2001.
14. *Гарсиа-Молина Г., Ульман Д., Уидом Д.* Системы баз данных. Полный курс. М. : Изд. дом «Вильямс», 2003.
15. *Малыхина М. П.* Базы данных: основы, проектирование, использование. СПб. : БХВ-Петербург, 2004.

16. *Дуго С. М.* Базы данных: проектирование и использование : учеб. пособие. М. : Финансы и статистика, 2005.

UDK 004.942

PAPER FORMATTING IN MICROSOFT WORD

T. N. Yesikova

Ph.D. in Economics, leading researcher
Institute of Economics and Industrial Engineering
Siberian Branch of the Russian Academy of Sciences
630090, Novosibirsk, Russia, Acad. Lavrentyev pr., 17, e-mail: T.N.Yesikova@gmail.com

S. V. Vakhrusheva

Graduate student, Novosibirsk State University
630090, Novosibirsk, Russia, Pirogova Str., 1, e-mail: s.vakhr@gmail.com

Annotation. The article poses the problem of modeling processes associated with the construction of large-scale infrastructural projects in the context of the Bering Strait tunnel. The study is based on such a simulation method as multi-agent modeling. The article describes the basics of building the environment of a multi-agent system in relation to this task, contains description of the interactions of the environment, and identified agents.

Keywords: large-scale infrastructural projects, simulation modelling, multi-agent system (MAS), agents, interaction environment, MAS architecture

References

1. *Lychkina N. N.* Информационные системы управления производственной компанией: учебник и практикум для академического бакалавриата. М. : Изд. Jurajt, 2018.
2. *Zhmurko S. A.* Obobshhennaja model' agenta i mnogoagentnoj sistemy // Izvestija JuFU. Tehnicheskie nauki. 2008. № 4.
3. *Gorodeckij V. I., Grushinskij M. S., Habalov A. V.* Mnogoagentnye sistemy // Novosti iskusstv. intellekta. 1998. № 2. P. 64–116.
4. *Horoshevskij V. F.* Metody i sredstva proektirovanija i realizacii mul'tiagentnyh sistem // Mater. ceminara «Problemy iskusstvennogo intellekta». IPU RAN, 1999.
5. *Masloboev A. V.* Mehanizmy vzaimodejstvija i koordinacii agentov v otkrytoj mul'tiagentnoj sisteme informacionnoj podderzhki regional'nyh innovacionnyh struktur. Teorija i praktika sistemnoj dinamiki : trudy II-j Vseros. nauch. konf. Apatity : KNC RAN, 2007.
6. *Masloboev A. V.* Mul'tiagentnaja tehnologija informacionnoj podderzhki innovacionnoj dejatel'nosti v regione. Prikladnye problemy upravlenija makrosistemami : trudy In-ta sistemnogo analiza RAN. M. : KomKniga URSS, 2009. T. 29.
7. *Zamjatina E. B.* Sovremennye teorii imitacionnogo modelirovanija. Perm': PGU, 2007.
8. *Aristov S. A.* Imitacionnoe modelirovanie jekonomicheskikh processov : ucheb. posobie. Ekaterinburg, 2003.
9. *Panteleev M. G., Kohtenko N. V., Lebedev S. V.* Sreda imitacionnogo modelirovanija agentnyh sistem real'nogo vremeni // Nauchno-tehnicheskij vestn. inf. tehnologij, mehaniki i optiki. 2012. № 1 (77).
10. *Chajkovskij D. V.* Informacionnoe prostranstvo: analiz opredelenij // Vestn. BGU. 2010. № 14.
11. *Manojlo A. V.* Gosudarstvennaja informacionnaja politika v osobyh uslovijah. M. : MIFI, 2003.
12. *Devjatkov V. V.* Sistemy iskusstvennogo intellekta : ucheb. posobie dlja vuzov. M. : Izd-vo MGTU im. N. Je. Baumana, 2001.

13. *Dejt K. Dzh.* Vvedenie v sistemy baz dannyh. M. : Izd.«Vil'jams», 2001.
14. *Garsia-Molina G., Ul'man D., Uidom D.* Sistemy baz dannyh. Polnyj kurs. M. : Izd. dom «Vil'jams», 2003.
15. *Malyhina M. P.* Bazy dannyh: osnovy, proektirovanie, ispol'zovanie. SPb, 2004.
16. *Digo S. M.* Bazy dannyh: proektirovanie i ispol'zovanie : ucheb. posobie dlja vuzov. M. : Finansy i statistika, 2005.

Разработка интеллектуальной СППР по угрозам энергетической безопасности**Г. Б. Загорулько**

науч. сотр., Институт систем информатики имени А. П. Ершова СО РАН,
630090, Новосибирск, Россия, пр. Акад. Лаврентьева, 6, e-mail: gal@iis.nsk.su

Л. В. Массель

д-р техн. наук, проф., зав. лабораторией информационных технологий в энергетике,
Институт систем энергетики им. Л. А. Мелентьева СО РАН,
664033, Иркутск, Россия, ул. Лермонтова, 130, e-mail: massel@isem.irk.ru

Аннотация. В статье описывается интеллектуальная система поддержки принятия решений (ИСППР) по угрозам энергетической безопасности. Данная система создана средствами методологии, объединяющей современные подходы и принципы разработки систем такого класса. ИСППР предоставляет доступ к систематизированной информации об угрозах энергетической безопасности и решает две конкретные задачи: моделирование угрозы похолодания и моделирование аварии на электростанции.

Ключевые слова: энергетическая безопасность, интеллектуальные системы поддержки принятия решений, методология разработки интеллектуальных СППР, ИСППР по угрозам энергетической безопасности

Введение

Понятие энергетической безопасности (ЭнБ) является одной из основных составляющих Концепции национальной безопасности Российской Федерации. Вопросами ЭнБ занимаются многие научные коллективы и государственные структуры. Правительством РФ принята Энергетическая стратегия России на период до 2030 г. В институте систем энергетики им. М. А. Мелентьева СО РАН (ИСЭМ СО РАН) разработана Доктрина энергетической безопасности России.

Существует много определений ЭнБ, отражающих разные аспекты этого понятия. Наиболее полным является понимание ЭнБ как «состояние защищенности страны, ее граждан, общества, государства и экономики от угрозы дефицита в обеспечении их обоснованных потребностей в энергии экономически доступными энергетическими ресурсами приемлемого качества, от угроз нарушения бесперебойности энергоснабжения». При этом состояние защищенности понимается в двух аспектах: как состояние, соответствующее в нормальных условиях обеспечению в полном объеме обоснованных потребностей в энергии, а в экстремальных условиях – гарантированному обеспечению минимального объема потребностей.

Моделирование угроз ЭнБ (УЭнБ) и разработка превентивных и ликвидационных мероприятий являются важными актуальными задачами, решением которых занимаются в ИСЭМ СО РАН. Создана онтология угроз ЭнБ, разработан ряд моделей и систем поддержки принятия решений (СППР) задач ЭнБ.

Описанная в статье интеллектуальная СППР (ИСППР) является совместной разработкой коллективов ИСЭМ СО РАН и института систем информатики им. А. П. Ершова (ИСИ СО РАН). Для ее создания была использована методология разработки ИСППР (ИСИ СО РАН), которая объединяет современные подходы и принципы создания систем такого класса [1], и

онтология УЭнБ (ИСЭМ СО РАН) [2]. ИСППР предоставляет доступ к систематизированной информации о своей предметной области и решает две конкретные задачи – моделирование угрозы похолодания и моделирование аварии на электростанции.

1. Методология разработки интеллектуальных СППР

Методология разработки ИСППР объединяет современные подходы и принципы создания систем такого класса. На рис. 1 показаны основные элементы методологии. Разработка ИСППР основывается на таких принципах, как Максимальное использование готовых решений, Информативность, Доступность, Модульность, Открытость, Масштабируемость, Независимость от предметной области (ПО). При разработке используются Онтологический, Фрактально-стратифицированный [3], Сервис-ориентированный подходы, подходы Быстрого прототипирования и Гибкой разработки, Технология Semantic Web, и Технология разработки информационно-аналитических интернет-ресурсов (ИАИР) [4].

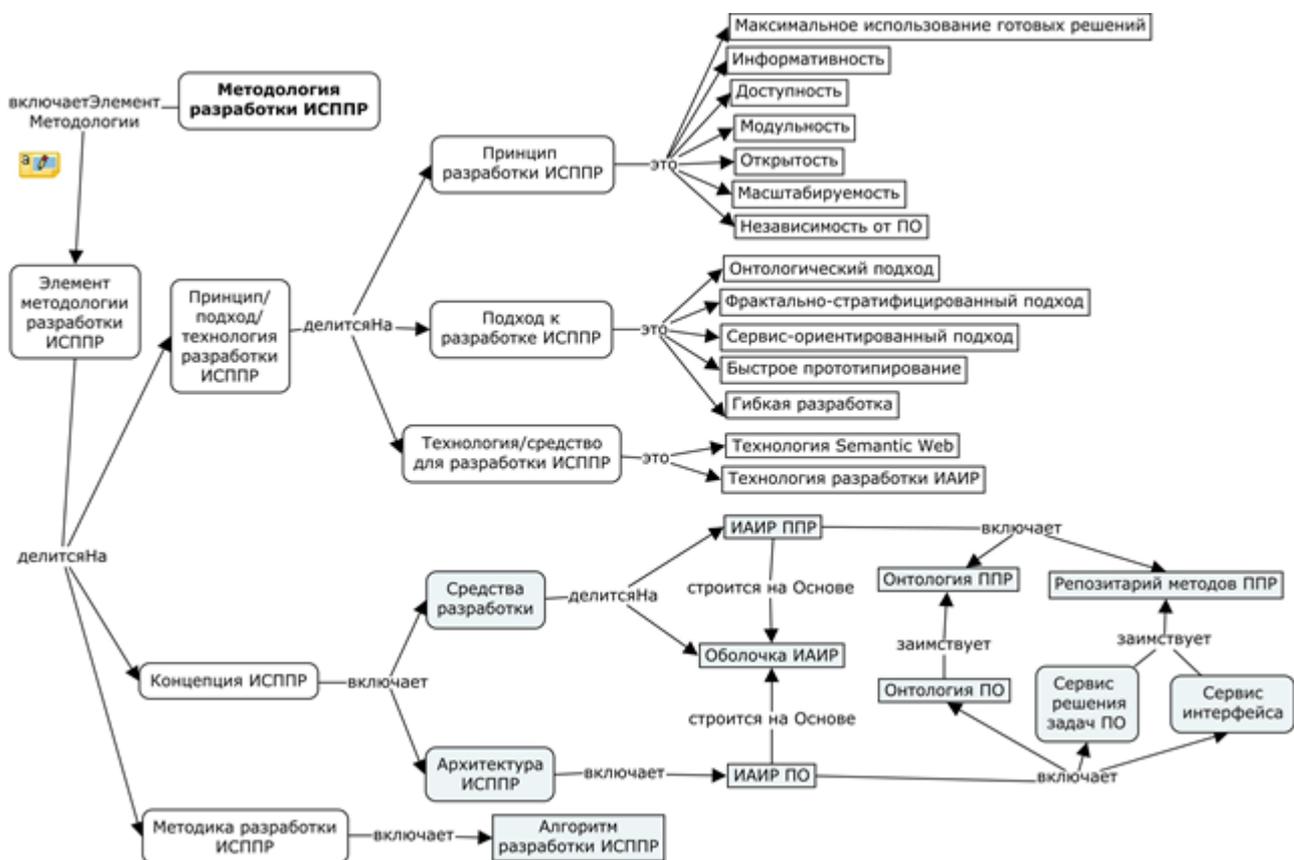


Рис. 1. Элементы методологии разработки ИСППР

Предлагаемые методологией средства позволяют получать быстрые прототипы СППР, легко осуществлять их модификацию за счет использования онтологии области знаний «Поддержка принятия решений» (ППР), построенного на ее основе ИАИР, репозитория методов ППР и оболочки ИАИР [1]. Онтология ППР предоставляет концептуальный базис и позволяет организовать взаимодействие всех типов разработчиков (экспертов, инженеров знаний, программистов). ИАИР ППР предоставляет доступ к систематизированной на основе онтологии информации об области знаний (ОЗ) ППР. Методы ППР, входящие в репозиторий, реализованы в виде сервисов, что обеспечивает масштабируемость репозитория и позволяет легко интегрировать методы между собой. ИАИР ППР является каркасом разрабатываемой

ИСППР, к которому будут подключаться необходимые сервисы, обеспечивающие ее функциональность.

Предлагаемые средства для разработки ИСППР не зависят от их предметных областей, однако для облегчения представления знаний об этих областях разработчики могут воспользоваться набором базовых онтологий и паттернов онтологического проектирования [5], методикой построения онтологии ПО. Ресурс, созданный на основе этой онтологии и оболочки ИАИР, является информационной системой, систематизирующей знания о своей ПО и предоставляющий к ним содержательный доступ. Подключенные к ресурсу сервисы из репозитория делают его полноценной ИСППР и позволяют решать поставленные перед ней задачи.

2. Разработка ИСППР УЭнБ

Рассматриваемая ИСППР строится на основе онтологии ПО «Угрозы ЭНБ». Был выполнен системный анализ данной ПО, выделены ее цели, задачи, объекты, предметы, субъекты, методы исследования [1,2,6]. Построена иерархия угроз ЭНБ (рис. 2).

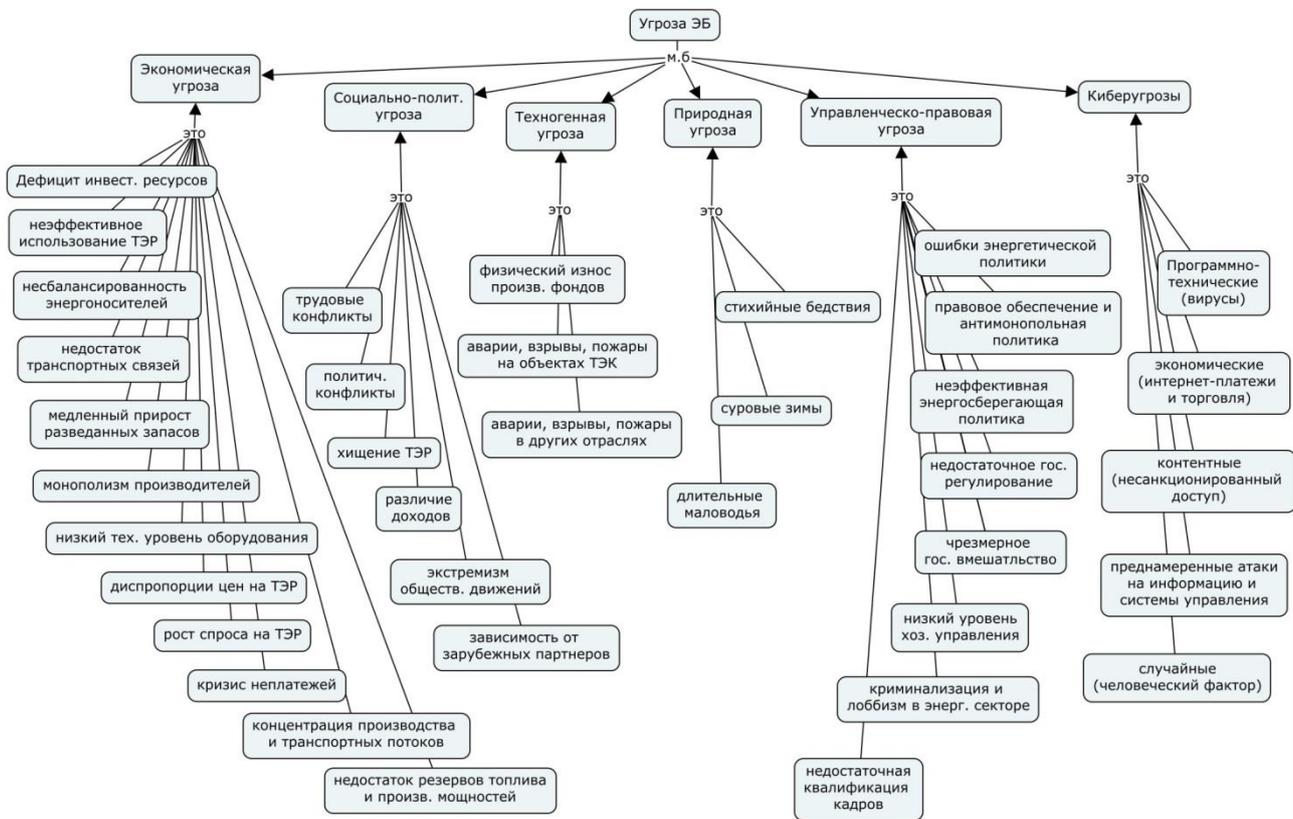


Рис. 2. Иерархия угроз ЭНБ

ИСППР УЭнБ должна решать конкретные задачи. Были построены модели этих задач. С помощью ресурса ИАИР ППР выбраны методы и средства их решения. В онтологии ППР выделен фрагмент, описывающий классы, к которым принадлежат решаемые ИСППР конкретные задачи и их связи с другими классами онтологии. Выделенный фрагмент был встроен в Онтологию УЭнБ. В репозитории методов ППР выделены методы, решающие задачи ИСППР УЭнБ, и встроены в Библиотеку сервисов (рис. 1).

На основе Онтологии УЭнБ был разработан ресурс ИАИР УЭнБ (рис. 3). В его контент добавлены информационные объекты, описывающие решаемые ИСППР задачи. Каждый такой информационный объект содержит неформальное описание задачи, связи с другими объектами онтологии – предметами и объектами исследования разрабатываемой ИСППР,

методами ППР и программными разработками, подходящими для решения данной задачи, необходимыми входными данными и способами их представления, сервисами, позволяющими задать входные данные, запустить процесс решения задачи и посмотреть результат.

ИСППР УЭнБ решает две конкретные задачи: моделирование угрозы похолодания и моделирование аварии на электростанции. Подробно задачи и их аналитические модели описаны в публикациях [7] и [6] соответственно.

В первой задаче моделируется ситуация, когда из-за снижения температуры наружного воздуха относительно среднесуточной в определенные месяцы отопительного периода повышается потребление тепло- и электроэнергии. Модель задачи отражает аналитические зависимости между этими параметрами и позволяет оценить, насколько нужно увеличить выработку указанных видов энергии, чтобы не допустить ее дефицита.

Главная Онтология О СППР

УГРОЗЫ ЭНЕРГЕТИЧЕСКОЙ БЕЗОПАСНОСТИ

СИСТЕМА ПОДДЕРЖКИ ПРИНЯТИЯ РЕШЕНИЙ

Табличное представление Графовое представление

Свойства объекта

Название	Авария на электростанции
Описание	/threats/uploads/40/b3/d659ad18a7019b9eee39bf177608.pdf

Связи объекта

- ликвидирует последствия
- Угроза Энергетической безопасности
- ЧС на объектах ТЭК
- связана с Задачей
- Задача Энергетической безопасности
- Разработка и реализация превентивных мероприятий
- Разработка ликвидационных мероприятий

Обратные связи объекта

- описывает Задачу
- Публикация
- Определение и угрозы энергетической безопасности, 2016, статья
- решает Конкретную Задачу
- Метод / средство
- Метод недоопределенных вычислений

Инструкция	/threats/uploads/75/83/1554f7f6be9cb89f724302899608.pdf
Ввод данных	http://uniserv.iis.nsk.su/threats-files/web/accident/Client.html
Модель Задачи	/threats/uploads/0a/1f/8913665556823c11ecfb7110e923.uni
Решатель	http://uniserv.iis.nsk.su/unicalc

© ИСИ, ИСЭМ 2018 Ресурс разработан при финансовой поддержке РФФИ (проект №16-07-00569)

Рис. 3. ИСППР «Угрозы ЭнБ»

Во второй задаче моделируется реакция сети энергоснабжения на угрозы технологического типа. В случае аварийной ситуации у какого-либо производителя подключаются пункты резервного хранения энергии. Суммарное содержание ресурса в таких пунктах должно компенсировать его сокращение у производителей в результате аварии. Любой потребитель может получать ресурс из любого пункта резервного хранения, т. е. потенциально между ними можно установить связь. В модели задачи задаются веса таких связей. Цель решаемой задачи – оптимизировать потоки ресурсов из резервных хранилищ потребителям с учетом весов их связей.

Для решения этих задач в ИСППР УЭнБ был выбран метод недоопределенных вычислений. Данный метод позволяет представить модель решаемой задачи в виде набора параметров, связанных математическими соотношениями, без разделения параметров на входные и выходные. Параметры могут задаваться интервальными значениями, которые должны сужаться в процессе вычислений. Для каждой задачи были реализованы сервисы, позволяющие пользователям ИСППР задавать значения параметров модели, запускать модели на счет и отображать полученные результаты.

Заключение

В статье описана методология разработки ИСППР, которая объединяет современные подходы и принципы построения систем данного класса. Предлагаемые ею средства позволяют быстро строить и легко использовать на практике ИСППР в различных слабоформализованных областях. Методология была использована для построения ИСППР по угрозам ЭНБ. Каркасом данной ИСППР является ИАИР, построенный на основе онтологии УЭнБ. ИСППР предоставляет доступ к систематизированной информации об угрозах ЭНБ и решает две конкретные задачи: моделирование угрозы похолодания и моделирование аварии на электростанции.

Работа выполнена при частичной финансовой поддержке РФФИ (грант № 19-07-00762).

Список литературы

1. *Загорулько Г. Б.* Методические аспекты разработки интеллектуальных СППР в слабоформализованных предметных областях на основе информационно-аналитических ресурсов // Шестнадцатая Нац. конф. по искусственному интеллекту с междунар. участием КИИ-2018 (24–27 сентября 2018 г., г. Москва, Россия) : труды конф. : в 2 т. М. : РКП, 2018. Т. 1. С. 36–43.
2. *Пяткова Н. И., Массель Л. В., Массель А. Г.* Методы ситуационного управления в исследованиях проблем энергетической безопасности // Известия Академии наук. Энергетика. 2016. № 4. С. 156–163.
3. *Массель Л. В.* Фрактальный подход к структурированию знаний и примеры его применения // Онтология проектирования. 2016. Т. 6, № 2 (20). С. 149–161.
4. *Загорулько Ю. А., Загорулько Г. Б., Боровикова О. И.* Технология создания тематических интеллектуальных научных интернет-ресурсов, базирующаяся на онтологии // Программная инженерия. 2016. Т. 7, № 2. С. 51–60.
5. *Zagorulko Y., Zagorulko G., Borovikova O.* Implementation of Content Patterns in the Methodology of the Development of Ontologies for Scientific Subject Domains // Communications in Computer and Information Science. 2018. Vol. 934. P. 260–272.
6. *Карнов В. В., Симанчев Р. Ю.* Определение и угрозы энергетической безопасности // Вестн. Омского ун-та. Серия: Экономика. 2016. № 4. С. 30–38.
7. *Пяткова Е. В.* Методика моделирования угроз энергетической безопасности с помощью байесовских сетей доверия // Современные технологии. Системный анализ. Моделирование. 2013. № 3 (39). С. 133–139.
8. *Нариньяни А. С., Телерман В. В., Ушаков Д. М., Швецов И. Е.* Программирование в ограничениях и недоопределенные модели // Информационные технологии. 1998. № 7. С. 13–22.

DEVELOPMENT OF INTELLIGENT DSS ON THREATS TO ENERGY SECURITY**G. B. Zagorulko**

Researcher, A. P. Ershov Institute of Informatics Systems
Siberian Branch of the Russian Academy of Sciences
630090, Novosibirsk, Russia, Acad. Lavrentjev pr., 6, e-mail: gal@iis.nsk.su

L. V. Massel

D. Tech., Professor, Head. Laboratory "Information Technology in Energetics"
Melentiev Energy Systems Institute
Siberian Branch of the Russian Academy of Sciences
664033, Irkutsk, Russia, Lermontov Str., 130, e-mail: massel@isem.irk.ru

Annotation. The paper describes an intelligent decision support system (IDSS) on threats to energy security. This system is created by means of a methodology that combines modern approaches and principles for developing systems of this class. IDSS provides access to systematized information about the threats to energy security and solves two specific tasks: modeling the threat of cooling and modeling the accident at a power plant.

Keywords: energy security, intelligent decision support systems, the methodology for the development of intelligent DSS, IDSS on threats to energy security

References

1. *Zagorulko G. B.* Metodicheskie aspekty razrabotki intellektual'nyh SPPR v slaboformalizovannyh predmetnyh oblastyah na osnove informacionno-analiticheskikh resursov [Methodical aspects of development of intelligent DSS in weakly formalized subject domains based on information-analytical resources] // Sixteenth National Conference on Artificial Intelligence with International Participation RCAI-2018 (24–27 September, 2018, Moscow, Russia). Conference proceedings. M. : RKP, 2018. Vol. 1. P. 36–43. (in Russian)
2. *Pyatkova N. I., Massel L. V., Massel A. G.* Metody situacionnogo upravleniya v issledovaniyah problem energeticheskoy bezopasnosti [Methods of situational management in researching energy security problems] // "Proceedings of the Russian Academy of Sciences. Power Engineering" Journal. 2016. № 4. P. 156–163. (in Russian)
3. *Massel L. V.* Fraktal'nyj podhod k strukturirovaniyu znaniy i primery ego primeneniya [Fractal approach to knowledge structuring and examples of its application] // Ontology of designing. 2016. Vol. 6, № 2 (20). P. 149–161. (in Russian)
4. *Zagorulko Yu. A., Zagorulko G. B., Borovikova O. I.* Tekhnologiya sozdaniya tematicheskikh intellektual'nyh nauchnyh internet-resursov, baziruyushchayasya na ontologii [Technology for building subject-based intelligent scientific internet resources based on ontology] // Software Engineering. 2016. Vol. 7. № 2. P. 51–60. (in Russian)
5. *Zagorulko Y., Zagorulko G., Borovikova O.* Implementation of Content Patterns in the Methodology of the Development of Ontologies for Scientific Subject Domains // Communications in Computer and Information Science. 2018. Vol. 934. P. 260–272.
6. *Karpov V. V., Simanchev R. YU.* Opredelenie i ugrozy energeticheskoy bezopasnosti [Identification and threats to energy security] // Herald of Omsk University. Series : ECONOMICS. 2016. № 4. P. 30–38. (in Russian)
7. *Pyatkova E. V.* Metodika modelirovaniya ugroz energeticheskoy bezopasnosti s pomoshch'yu bajesovskikh setej doveriya [Energy Security Threat Modeling Technique Using Bayesian Confidence Networks] // Modern technologies. System analysis. Modeling. 2013. № 3(39). P. 133–139. (in Russian)

8. *Narinyani A. S., Telerman V. V., Ushakov D. M., Shvetcov I. E.* Programmirovaniye v ogranicheniyah i nedoopredelyonnye modeli [Constraint programming and subdefinite models] // Information Technology. 1998. № 7. P. 13–22. (in Russian)

Применение клеточных автоматов для моделирования движения группы людей

М. А. Кантуреева

Аспирант, Евразийский национальный университет имени Л. Н. Гумилева,
010000, Астана, Казахстан, ул. Сатбаева, 2, e-mail: monsiko@mail.ru

Аннотация: В данной работе речь идет о моделировании движения множества людей. Используется конечно-автоматная (КА) модель. Проведен анализ методов моделирования поведения толпы, в том числе изучалась зависимость плотности, скорости толпы и пропускной способности различных сооружений от различных внешних параметров. Основным инструментальным средством для выполнения данной работы является программная система AnyLogic. Автором была реализована модель небольшого аэропорта.

Ключевые слова: клеточные автоматы, моделирование движения людей, полевая модель движения людей, КА-модель, FF-модель, модель аэропорта, AnyLogic

Введение

Клеточные автоматы являются частным случаем конечных автоматов. В частности, они используются для моделирования динамического поведения однородных сред. Поведение клеточных автоматов определяется в терминах локальных зависимостей. Каждая клетка представляет собой конечный автомат, состояние которого в очередной момент времени определяется состояниями ее соседей в некоторой окрестности. Обычно рассматриваются так называемая окрестность фон Неймана (четыре соседа, расположенные ортогонально клетке) и окрестность Мура (четыре соседа, примыкающие по диагоналям).

Первые конечно-автоматные модели (КА-модели) движения людей основывались на моделях дорожного трафика. Динамика таких моделей в основном была детерминистическая. Например, изучалось двунаправленное движение в длинном коридоре. Обновление положения частиц происходило последовательно. Частицам задавалось только направление для перемещения. Если «желаемая» ячейка была занята другой частицей, двигающейся в противоположном направлении, то частица могла случайным образом выбрать себе другую незанятую ячейку для перемещения. Фактически, движение частиц рассматривалось как аналог многополосного дорожного движения.

В дальнейшем появились расширения и вариации данных моделей. Например, различные специальные типы обновления положения частиц, возможность обратного движения. Исследовалось влияние формы частиц, другие геометрии, обобщения для полноценного двухмерного движения, когда передвижение частиц не ограничивалось только близлежащими клетками, частицы могли передвигаться на две, три или четыре клетки за один шаг.

Чтобы избегать конфликтов между частицами, которые пытались двигаться в одну и ту же позицию, обновление положения частиц происходило последовательно [1,2]. В каждой ячейке пространства назначались очки на основе ее близости к другим частицам. Очки описывали «отталкивающие» взаимодействия, а передвижение определялось посредством конкуренции между этими силами отталкивания, и их рост приближал частицу к цели.

1. Полевая модель движения людей

Широкое распространение получила полевая модель Floor Field (FF). В ее основе лежит использование так называемых полей [3] для моделирования движущей силы и

взаимодействия с другими частицами. В модели используются два поля: динамическое D и статическое S . Эти поля имеют такую же дискретную структуру, как и само пространство, по которому передвигаются частицы в КА-моделях. Динамическое поле D соответствует виртуальному следу, который создается движением частиц и оказывает влияние на движение других. Оно имеет свою собственную динамику, а именно: рассеивание и забывание. Статическое поле S или поле расстояний не изменяется со временем, представляет своего рода карту местности, где каждая ячейка содержит обратное расстояние до ближайшего выхода с учетом всех неподвижных препятствий. Поле S не зависит от наличия или отсутствия частиц в рассматриваемой области. Значение S увеличивается с приближением к выходу.

Переходные вероятности для всех частиц зависят от значений полей S и D в соседних клетках. Формула вычисления вероятностей устроена так, что наибольшая вероятность придается направлениям с наибольшими значениями поля. Влияние обоих полей на движение частицы контролируется константами k_s и k_D соответственно. Чем больше значение k_s , тем более направленное движение к выходу и более короткий путь выбирают частицы. При сильном привязывании к динамическому полю, т. е. при увеличении k_D наблюдается все более выраженное «стадное» поведение, когда индивидуумы пытаются следовать за другими.

Классическая формула переходных вероятностей (например, в направлении φ) имеет вид

$$\rho_{\varphi} = Norm^{-1} e^{k_D D_{\varphi}} e^{k_s S_{\varphi}} (1 - f_{\varphi}) \omega_{\varphi},$$

где $Norm$ – нормализующий коэффициент (сумма всех переходных вероятностей), k_s, k_D – параметры модели, S_{φ}, D_{φ} – значения статического и динамического полей в ячейке по направлению φ . Величина f_{φ} задает занятость или свободность ячейки. Если $f_{\varphi} = 0$, то ячейка свободна, и если $f_{\varphi} = 1$, то ячейка занята. Величина ω_{φ} указывает на наличие ($\omega_{\varphi} = 1$) или отсутствие ($\omega_{\varphi} = 0$) стены и других неподвижных препятствий. Перемещение в занятую ячейку или ячейку, относящуюся к препятствию, запрещено. Частица может остаться на прежнем месте только в случае, когда все соседние клетки заняты или относятся к препятствиям.

Правила переходов устроены следующим образом: вычисляется статическое поле S , динамическое поле D в начальный момент времени имеет нулевое значение, для каждой частицы и для каждого направления перемещения определяются переходные вероятности и на основе полученного распределения возможных направлений выбирается направление для перемещения. Если в модели используется синхронное обновление положения частиц, то возникают конфликты, связанные с тем, что на одну ячейку могут претендовать несколько частиц [4]. После разрешения таких конфликтов частицы перемещаются на новые позиции, динамическое поле обновляется.

В настоящее время динамическое поле D практически не учитывается при моделировании ($k_D = 0$). Но идея использования поля расстояний S для моделирования движущей силы явилась существенным улучшением по сравнению с другими моделями и позволила легко воспроизводить направленное движение людей.

На основе FF-подхода было разработано много конкретных моделей. Анализ работ показывает, что FF-модели различаются по следующим особенностям:

- 1) способ дискретизации пространства, размеры и форма ячеек;
- 2) шаблон соседства;
- 3) скорость движения частиц (перемещение на одну ячейку или на несколько);
- 4) способ вычисления поля S ;

- 5) режимы работы КА, синхронный или асинхронный;
- 6) способ вычисления вероятностей переходов;
- 7) правила переходов;
- 8) способ разрешения конфликтов.

В FF-моделях обычно используется обратное поле расстояний, т. е. значения поля S увеличиваются при приближении к выходу. Также в FF-моделях обычно используется синхронный режим работы КА. Все частицы перемещаются на новые позиции одновременно. При таком режиме возникают конфликтные ситуации, когда две или более частицы претендуют на одну ячейку. Асинхронный режим исключает возникновение конфликтных ситуаций. В этом случае частицы для перемещения, как правило, выбираются случайным образом [2,3].

В работах А. Кирхнера, К. Нишинари и А. Шадшнайдера [5] исследуются способы разрешения конфликтов. Основная идея состоит в следующем. С вероятностью $\mu \in [0,1]$ одной из частиц, которая выбирается случайно из всех кандидатов, разрешается переместиться в спорную ячейку, остальные остаются на месте. Это позволяет описывать эффект затруднения движения между частицами. Параметр μ можно интерпретировать как некоторый вид локального давления между частицами. Если значение μ близко к единице, тогда соседние частицы могут сильно мешать друг другу достичь желаемых ячеек. Величину μ называют фрикционным параметром. В простейшем случае при возникновении конфликтной ситуации случайным образом выбирается одна частица для перемещения, остальные остаются на прежних местах [4,5]. Формула для вычисления вероятностей переходов – самая вариативная часть в моделях движения людей.

Приведем некоторые более точные определения, касающиеся КА-моделей [1].

Клеточный автомат обычно определяется множеством клеток, плотно заполняющих дискретное D -мерное пространство. Паре $\langle i, j \rangle, (0 < i < N, 0 < j < M)$ соответствует ячейка, левый нижний угол, который имеет данные координаты [4]. Считаем, что в каждой точке задан $\varphi(i, j)$ – потенциал, который отражает некоторое поле, действующее на людей. Потенциал можно задать посредством непрерывной функции $\varphi(x, y)$. Далее производим дискретизацию функции. Как обычно с потенциалом можно связать «напряженность поля» в узле сетки:

$$E_{ij}^x = \varphi(i, j + 1) - \varphi(i, j - 1),$$

$$E_{ij}^y = \varphi(i + 1, j) - \varphi(i - 1, j).$$

Следующий шаг – это задание вероятностей перехода объекта в соседние ячейки.

Сначала определяем шаблон соседства $T(i, j)$. Например, если перемещение возможно в четырех направлениях (по горизонтали и по вертикали), то шаблон соседства имеет вид:

	$(i - 1, j)$	
$(i, j - 1)$	(i, j)	$(i, j + 1)$
	$(i + 1, j)$	

С движением частицы можно связать (ассоциировать) случайную величину σ . При этом имеет место формула $\sigma = u_{ij-1} + u_{ij+1} + u_{i-1,j} + u_{i+1,j}$, а следующие величины

$$\rho_1 = \frac{u_{ij-1}}{\sigma}, \quad \rho_2 = \frac{u_{ij+1}}{\sigma}, \quad \rho_3 = \frac{u_{i-1j}}{\sigma}, \quad \rho_4 = \frac{u_{i+1j}}{\sigma}$$

трактуются как вероятности перехода объекта налево, направо, вниз и вверх соответственно, очевидно, что $\sum_{i=1}^4 \rho_i = 1$.

Рассмотрение только четырех направлений не означает серьезного ограничения. Переходы по диагональным направлениям могут быть легко осуществлены [5]. Направление движения частицы определяется на основе вероятностей переходов в каждом направлении для каждой частицы на каждом дискретном шаге времени и набором правил переходов [6,7].

Способы задания потенциалов. Традиционная схема пошаговых вычислений для дискретного подхода, основанного на теории КА, заключается в следующем. Сначала вычисляются переходные вероятности, в соответствии с этим выбираются направления движения для каждой частицы, затем происходит разрешение конфликтов, если две или более частицы оказались претендентами на одну ячейку, и непосредственно перемещение.

Если $i_0 j_0$ – некоторая точка, то можно задать потенциал, принимающий максимально значение в этой точке, имеющей форму «шапочки». «Шапочку» можно задать с помощью нормального распределения:

$$f(x, y) = \frac{1}{\sigma_x \sigma_y \sqrt{2\pi}} e^{-\frac{(x-\mu_x)^2}{\sigma_x^2} - \frac{(y-\mu_y)^2}{\sigma_y^2}}$$

Обычно рассматривают частный случай $\sigma_x = \sigma_y, \mu_x = \mu_y$.

По нашему мнению, более интересно рассмотреть функции, исследуемые в размытой логике Заде [8, 9]. Например, функции, задающие нечеткие числа.

Нечетким числом называют нормализованное и выпуклое нечеткое множество A , определенное на множестве действительных чисел \mathbf{R} , т. е. для функции принадлежности которого $\mu_A(x)$ выполнено:

- 1) $\max_{x \in \mathbf{R}} \mu_A(x) = 1$, нечеткое число нормализовано;
- 2) $\mu_A(\lambda x_1 + (1 - \lambda)x_2) \geq \min(\mu_A(x_1); \mu_A(x_2))$, число выпукло.

2. Результаты исследования модели движения людей

Проведен обзор и анализ многих натуральных наблюдений за поведением реальных толп, в том числе изучение зависимости плотности, скорости толпы и пропускной способности различных сооружений от различных внешних параметров. Основным инструментальным средством для выполнения данной работы является программная система AnyLogic [10]. Автором данной работы была создана модель небольшого аэропорта.

AnyLogic является уникальным программным продуктом, поддерживающим три методологии имитационного моделирования, называемых системной динамикой, дискретно-событийным и агентным моделированием. Система включает набор примитивов и библиотечных объектов для эффективного моделирования производства и логистики, бизнес-процессов и персонала, финансов, потребительского рынка, а также окружающей инфраструктуры в их естественном взаимодействии.

В первой разработанной модели имеется возможность 2D-анимации (рис. 1). В следующей модели используется 3D-анимация (рис. 2). Добавлена камера, возникает соответствующее 3D-окно, и пассажиры также отображаются в виде 3D-объектов.

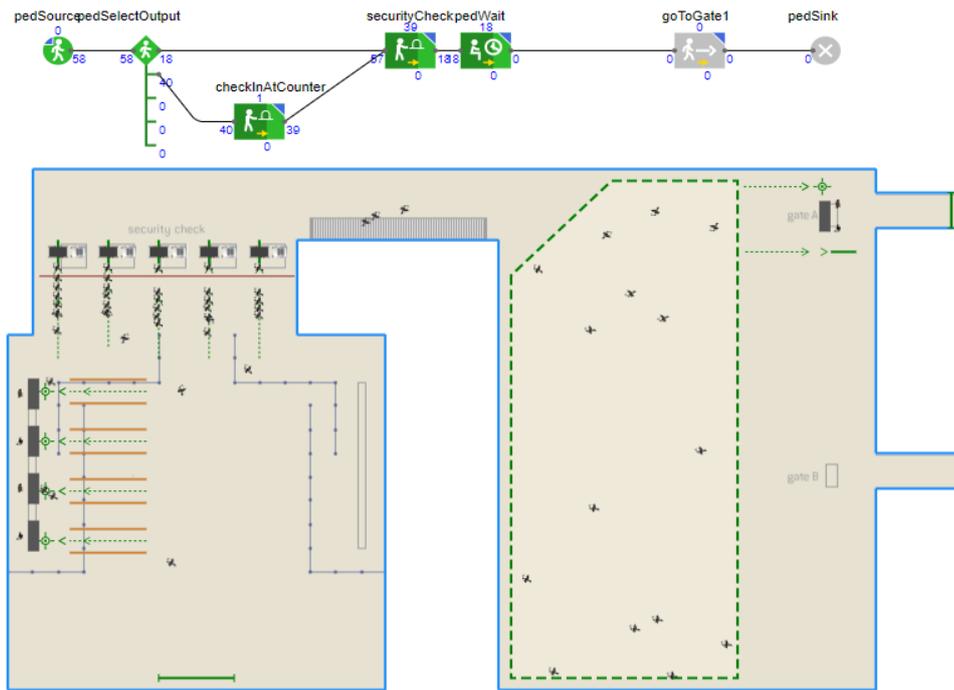


Рис. 1. 2D-модель аэропорта

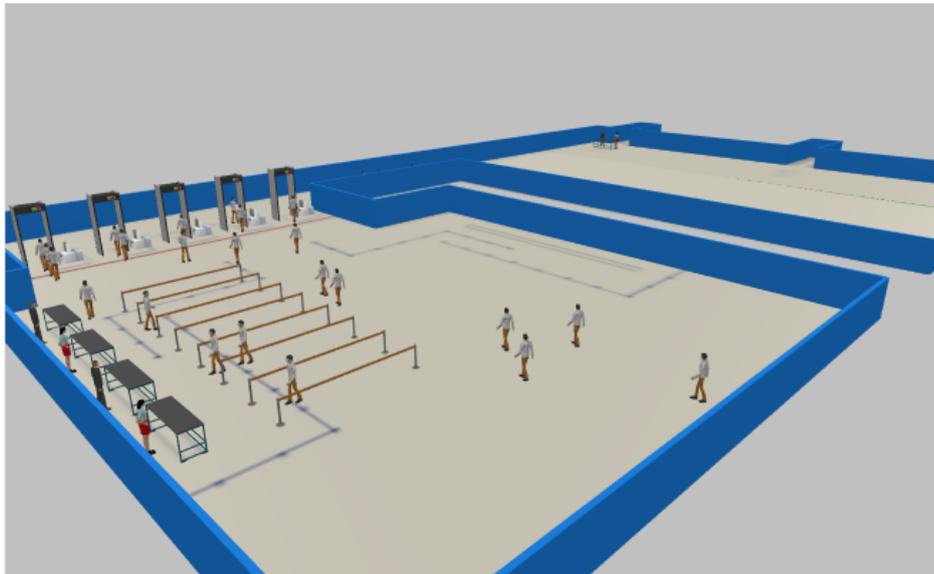


Рис. 2. 3D-модель аэропорта

Пассажиры, прибывающие в аэропорт, регистрируются на рейс. Затем все пассажиры должны будут пройти процедуру предполетного досмотра, после чего они смогут направиться в зону ожидания перед гейтами, дожидаясь начала посадки на свой рейс. При объявлении начала посадки пассажиры направляются к соответствующему гейту. У гейта служащие аэропорта проводят проверку посадочных талонов, после чего пассажиры проходят на посадку на самолет.

Изменения скоростей движения участников толпы зависят от многих причин, трудно подчиняющихся точному расчету. Поэтому при оценке скорости движения неизбежно приходится прибегать к средним значениям, которые можно считать надежными, если они установлены на основании статистических методов.

В значительной степени скорость движения зависит от плотности людской массы. В рассмотренных моделях движение людских потоков носит спокойный характер. Если предположить, что в таких скоплениях неожиданно возникнет аварийная ситуация, легко представить себе к каким серьезным последствиям она может привести. Изучение подобных ситуаций требует усложнения математических моделей.

Список литературы

1. *Малинецкий Г. Г., Степанцов М. Е.* Применение клеточных автоматов для моделирования движения группы людей // Журнал вычислительной математики и математической физики. 2004. № 11. С. 2094–2098.
2. *Bandini S., Rubagotti F., Vizzari G., Shimura K.* A Cellular Automata Model for Pedestrian and Group Dynamics: Motivations and First Experiments // Parallel Computing Technologies. Berlin Heidelberg : Springer, 2011. LNCS 6873. P. 125–139.
3. *Leng B., Wang J., Zhao W., Xiong Z.* An extended floor field model based on regular hexagonal cells for pedestrian simulation // Physica A. 2014. P. 119–133.
4. *Varas A., Cornejo M. D., Mainemer D. et al.* A Cellular automaton model for evacuation process with obstacles // Physica A.: Statistical Mechanics and its Applications. 2007. P. 631–642.
5. *Kirchner A., Nishinari K., Schadschneider A.* Friction effects and clogging in a cellular automaton model for pedestrian dynamics // Physical Review E. 2003. May. Vol. 67(5 Pt 2). P. 056122.
6. *Bandini S., Gorrini A., Vizzari G.* Towards an Integrated Approach to Crowd Analysis and Crowd Synthesis: a Case Study and First Results // Pattern Recognition Letters. 2014. P. 16–29.
7. *Кирик Е. С., Юргельян Т. Б., Круглов Д. В., Малышев А. В.* О непрерывно-дискретной стохастической модели движения людей // Материалы XIII Всерос. семинара Моделирование неравновесных систем. Красноярск : ИВМ СО РАН, 2010. С. 81–85.
8. *Заде Л. А.* Понятие лингвистической переменной и его применение к принятию приближенных решений. М. : Мир, 1976. 165 с.
9. *Пивкин В. Я., Бакулин Е. П., Кореньков Д. И.* Нечеткие множества в системах управления: учеб. пособие. Новосибирск : Изд-во НГУ, 1997. 52 с.
10. AnyLogic (XJ Technologies). URL: <http://www.gpss.ru/systems/anylogic.html> (accessed: 20.10.2018).

UDK 519.63, 519.688

APPLICATION OF CELLULAR AUTOMATA FOR MODELING THE GROUP OF PEOPLE MOVEMENT

M. A. Kantureeva

Ph.D. student, L. N. Gumilyov Eurasian National University,
010000, Astana, Kazakhstan, Satbayev str., 2, e-mail: monsiko@mail.ru

Abstract: In this paper, we are talking about modeling the group of people movement. A finite automaton (FA) model is used. The analysis of methods for modeling the crowd behavior was carried out, including the dependence of density, crowd speed and ability of throughput in various buildings for various external parameters. The main tool for developing this work is AnyLogic software system. The author has implemented a model of a small airport.

Keywords: cellular automata, simulation of the movement of people, field model of movement of people, FA model, FF model, airport model, AnyLogic

References

1. *Malinetsky G. G., Stepantsov M. E.* Primeneniye kletochnykh avtomatov dlya modelirovaniya dvizheniya gruppy lyudei [The use of cellular automata to simulate the movement of a group of people] // Zhurnal vychislitelnoi matematiki i matematicheskoi fiziki = Journal of computational mathematics and mathematical physics. 2004. № 11. P. 2094–2098. (in Russian)
2. *Leng B., Wang J., Zhao W., Xiong Z.* An extended floor field model based on regular hexagonal cells for pedestrian simulation // Physica A. 2014. P. 119–133.
3. *Bandini S., Rubagotti F., Vizzari G., Shimura K.* A Cellular Automata Model for Pedestrian and Group Dynamics: Motivations and First Experiments // Parallel Computing Technologies. Berlin Heidelberg : Springer, 2011. LNCS 6873. P. 125–139.
4. *Varas A., Cornejo M. D., Mainemer D. et al.* A Cellular automaton model for evacuation process with obstacles // Physica A.: Statistical Mechanics and its Applications. 2007. P. 631–642.
5. *Kirchner A., Nishinari K., Schadschneider A.* Friction effects and clogging in a cellular automaton model for pedestrian dynamics // Physical Review E. 2003. May. Vol. 67(5 Pt 2). P. 056122.
6. *Bandini S., Gorrini A., Vizzari G.* Towards an Integrated Approach to Crowd Analysis and Crowd Synthesis: a Case Study and First Results // Pattern Recognition Letters. 2014. P. 16–29.
7. *Kirik E. S., Yurgelyan T. B., Kruglov D. V., Malyshev A. V.* O nepreryvno-diskretnoi stokhasticheskoi modeli dvizheniya lyudei [On the continuous-discrete stochastic model of the movement of people] // Materialy XIII Vserossiiskogo seminar Modelirovaniye neravnovesnykh sistem = Proceedings of the XIII All-Russian Seminar Simulation of non-equilibrium systems. Krasnoyarsk : ICM SB RAS, 2010. P. 81–85.
8. *Zadeh L. A.* Ponyatiye lingvisticheskoi peremennoi i yego primeneniye k prinyatiyu priblizhennykh reshenii [The concept of a linguistic variable and its application to approximate reasoning]. M. : Mir, 1976. 165 p. (in Russian)
9. *Pivkin V. Ya., Bakulin E. P., Korenkov D. I.* Nechetkiye mnozhestva v sistemakh upravleniya [The concept of a linguistic variable and its application to making approximate decisions]: uchebnoye posobiye = the tutorial. Novosibirsk : NSU, 1997. 52 p. (in Russian)
10. AnyLogic (XJ Technologies). URL: <http://www.gpss.ru/systems/anylogic.html> (accessed: 20.10.2018).

Вопрос карьерных перспектив в области ИТ

М. М. Лаврентьев

д-р физ.-мат. наук, проф., Новосибирский государственный университет,
630090, Новосибирск, Россия, ул. Пирогова, 2,
Институт автоматизации и электротехники,
630090, Новосибирск, Россия, пр. Акад. Коптюга, 1, e-mail: mmlavrentiev@gmail.com

Л. В. Городня

канд. физ.-мат. наук, доц., Институт систем информатики им. А. П. Ершова СО РАН,
630090, Новосибирск, пр. Акад. Лаврентьева, 6,
Новосибирский государственный университет,
630090, Новосибирск, Россия, ул. Пирогова, 2, e-mail: lidvas@gmail.com

М. А. Держо

доц., Новосибирский государственный университет,
630090, Новосибирск, Россия, ул. Пирогова, 2, e-mail: m_derjo@mail.ru

Д. С. Мигинский

канд. физ.-мат. наук, доц., Институт систем информатики им. А. П. Ершова СО РАН,
630090, Новосибирск, пр. Акад. Лаврентьева, 6,
Новосибирский государственный университет,
630090, Новосибирск, Россия, ул. Пирогова, 2, e-mail: dmiginsky@gmail.com

Аннотация. Цель статьи — анализ стандартов на профессиональную квалификацию в области ИТ с точки зрения обучения сложным профессиям и формирования профессионального корпуса. Сформулировано предложение по уточнению стандарта на профессию «системный программист» для разработчиков компиляторов.

Ключевые слова: обучение системному программированию, создание компиляторов, парадигмы программирования, семантические системы, понятийная сложность деятельности, профессиональная квалификация

Введение

К настоящему времени разработано и утверждено большое число стандартов на профессиональную квалификацию, на которые должна опираться профессиональная подготовка в учебных заведениях, нацеленная на предстоящий переход к сертификации соответствия профессиональным стандартам в производственной деятельности. Дальнейшее развитие принятых стандартов, требует достаточно четкой методики измерения фактической сложности работ и необходимого для них уровня знаний. Важно обратить внимание на положения, сформулированные относительно такой ключевой профессии как «Системный программист», для которой трудно осуществить карьерный рост и отсутствует система послевузовской поддержки квалификационного развития.

1. Результаты анализа системы стандартов в области ИТ

Анализ и сравнение опубликованных стандартов разного уровня требований к профессиональной квалификации позволяет выделить заметные различия в возможностях карьерных перспектив для представителей разных профессий. Примеры приведены в табл. 1.

Отдельные характеристики стандартов в порядке регистрации

№ Стандарта	Профессия	Число станиц	Уровни квалификации
06.001	Программист	21	3–6
06.003	Архитектор программного обеспечения	62	4–6
06.004	Специалист по тестированию в области информационных технологий	26	4–6
06.011	Администратор баз данных	36	4–7
06.012	Менеджер продуктов в области информационных технологий	24	4–7
06.014	Менеджер по информационным технологиям	34	6–9
06.015	Специалист по информационным системам	248	4–7
06.016	Руководитель проектов в области информационных технологий	117	6–8
06.017	Руководитель разработки программного обеспечения	24	6–7
06.019	Технический писатель (специалист по технической документации в области информационных технологий)	47	4–8
06.022	Системный аналитик	44	4–7
06.025	Специалист по дизайну графических и пользовательских интерфейсов	25	4–7
06.026	Системный администратор информационно-коммуникационных систем	50	4–7
06.027	Специалист по администрированию сетевых устройств информационно-коммуникационных систем	32	4–7
06.028	Системный программист (разработка драйверов и компиляторов)	38 = 3+ 6 +5 +15	6
	Системный программист (разработка БД)		7
	Системный программист (разработка ОС)		7
06.029	Менеджер по продажам информационно-коммуникационных систем	38	5–8
06.035	Разработчик Web и мультимедийных приложений	35	3–7
06.040	Специалист по контролю качества информационно-коммуникационных систем	30	5–7
06.041	Специалист по интеграции прикладных решений	21	4–7
06.044	Консультант в области развития цифровой грамотности населения (цифровой куратор)	14	3–5

Более ясно разница видна при упорядочении стандартов по квалификационным уровням, что представлено в табл. 2.

Характеристики стандартов по уровням квалификационного роста

№	Стандарт	Профессия	Число станци	Уровни квалификации
1	06.014	Менеджер по информационным технологиям	34	6–9
2	06.016	Руководитель проектов в области информационных технологий	117	6–8
3	06.029	Менеджер по продажам информационно-коммуникационных систем	38	5–8
4	06.019	Технический писатель (специалист по технической документации в области информационных технологий)	47	4–8
5	06.017	Руководитель разработки программного обеспечения	24	6–7
6	06.040	Специалист по контролю качества информационно-коммуникационных систем	30	5–7
7	06.011	Администратор баз данных	36	4–7
	06.012	Менеджер продуктов в области информационных технологий	24	4–7
	06.015	Специалист по информационным системам	248	4–7
	06.022	Системный аналитик	44	4–7
	06.025	Специалист по дизайну графических и пользовательских интерфейсов	25	4–7
	06.026	Системный администратор информационно-коммуникационных систем	50	4–7
	06.027	Специалист по администрированию сетевых устройств информационно-коммуникационных систем	32	4–7
	06.041	Специалист по интеграции прикладных решений	21	4–7
8	06.028	Системный программист (разработка БД)	5 из 38	7
	06.028	Системный программист (разработка ОС)	15 из 38	7
9	06.035	Разработчик Web и мультимедийных приложений	35	3–7
10	06.001	Программист	21	3–6
11	06.003	Архитектор программного обеспечения	62	4–6
	06.004	Специалист по тестированию в области информационных технологий	26	4–6
12	06.028	Системный программист (разработка драйверов и компиляторов)	6 из 38	6
13	06.044	Консультант в области развития цифровой грамотности населения (цифровой куратор)	14	3–5

Примерное различие в карьерных лифтах

	1	2	3	4	5	6	7								8	9	10	11	12	13		
9	1																					9
8	1	2	3	4																		8
7	1	2	3	4	5	6	7	7	7	7	7	7	7	7	8	9						7
6	1	2	3	4	5	6	7	7	7	7	7	7	7	7		9	10	11			13	6
5			3	4		6	7	7	7	7	7	7	7	7		9	10	11			13	5
4				4			7	7	7	7	7	7	7	7		9	10	11			13	4
3																9	10				13	3
2																						2
1																						1
	1	2	3	4	5	6	7								8	9	10	11	12	13		

Таблица 4

Разнообразие требований к трудовым функциям уровня 6

№	Стандарт	Профессия	Число станций	Число требований	Уровни квалификации
1	06.019	Технический писатель (специалист по технической документации в области информационных технологий)	47	13–41	4–8
2	06.028	Системный программист (разработка драйверов и компиляторов)	8 стр. из 38	27–36	6
3	06.029	Менеджер по продажам информационно-коммуникационных систем	38	13–27	5–8
4	06.035	Разработчик Web и мультимедийных приложений	35	13–26	3–7
5	06.014	Менеджер по информационным технологиям	34	12–16	6–9
6	06.004	Специалист по тестированию в области информационных технологий	26	6–16	4–6
7	06.001	Программист	21	12–15	3–6
8	06.003	Архитектор программного обеспечения	62	3–13	4–6
9	06.016	Руководитель проектов в области информационных технологий	117	5–8	6–8
	06.004	Специалист по тестированию в области информационных технологий	26	6–16	4–6

Табл. 2 показывает прикладное и образовательное значение выбора профессиональной подготовки для формирования профессионального корпуса специалистов в области ИТ. Профессиональная подготовка системных программистов как ключевой профессии согласно

такому спектру стандартов не имеет карьерного лифта внутри профессии. Для гармонизации системы стандартов нужна методика представления сложности разных видов работ, что может быть представлено средствами деловой графики, успешно применяемыми для представления сложных данных, традиционно сводимых к числовым соотношениям примерно как в табл. 3 (столбик 12 соответствует профессии «Системный программист (разработка драйверов и компиляторов)», столбик 8 – «Системный программист (разработка БД и ОС)»).

2. Анализ содержания модели трудовых функций

Следует отметить зависимость понимания смысла от владения оттенками профессиональной терминологии и вариациями определения понятий. При анализе трудовых функций можно рассмотреть матрицу, представленную на табл. 5, и сравнивать содержательную сложность трудовых функций одного уровня.

Таблица 5

Понятийная матрица для анализа трудовых функций

Виды трудовых функций	Знания	Умения	Навыки	Понимание	Развитие
1. Основа	Образование	Действия	Опыт	Управление	Комплекты
2. Фон	Информация	Процедуры	Сценарии	Структуры	Иерархия
3. Границы	Квалификация	Предикаты	Типизация	Цели	Варианты
4. Системность	Внештатные ситуации	Программы	Внешние носители	Поиск решений	Обмен данными

Для примера рассмотрим 6-й уровень стандарта на профессию «Программист» и стандарт на профессию «Системный программист».

Таблица 6

Понятийная матрица для анализа трудовых функций 6-й уровень стандарта на профессию «Программист»

Знания	
1. Основа	<i>Образование</i> – ВУЗ
2. Фон	<i>Информация</i> – Возможности существующей программно-технической архитектуры
3. Границы	<i>Квалификация</i> – Методологии разработки программного обеспечения и технологии программирования
4. Системность	<i>Внештатные ситуации</i> – Возможности современных и перспективных средств разработки программных продуктов, технических средств
Умения	
1. Основа	<i>Действия</i> – Проводить анализ исполнения требований
2. Фон	<i>Процедуры</i> – <i>Выбирать средства</i>

Курсивом выделены пункты, заимствованные из трудовой функции «Разработка спецификаций на компоненты и их взаимодействие»

3. Границы	<i>Предикаты</i> – Проводить оценку и обоснование рекомендуемых решений
4. Системность	<i>Программы</i> – Оценка времени и трудоемкости реализации требований к программному обеспечению
Навыки	
1. Основа	<i>Опыт</i> – 3 года
2. Фон	<i>Сценарии</i> – Контроль выполнения заданий
Границы	<i>Типизация</i> – Анализ возможностей реализации требований к программному обеспечению
4. Системность	<i>Внешние носители</i> – Оценка и согласование сроков выполнения поставленных задач
Понимание	
1. Основа	<i>Управление</i> – Анализ требований к программному обеспечению
2. Фон	<i>Структуры</i> – Методологии и технологии проектирования и использования баз данных
3. Границы	<i>Цели</i> – Разработка требований и проектирование программного обеспечения
4. Системность	<i>Поиск решений</i> – Анализ возможностей реализации требований к программному обеспечению
Развитие	
1. Основа	<i>Комплексы</i> – Осуществлять коммуникации с заинтересованными сторонами
2. Фон	<i>Иерархия</i> – Согласование требований к программному обеспечению с заинтересованными сторонами
3. Границы	<i>Варианты</i> – Вырабатывать варианты реализации требований
4. Системность	<i>Обмен данными</i> – Отчет

Таблица 7

**Понятийная матрица для анализа трудовых функций стандарт на профессию
«Системный программист» (компиляторы и т. п.)**

Знания	
Основа	<i>Образование</i> – Бакалавриат
2. Фон	<i>Информация</i> – Архитектура целевой аппаратной платформы – Алгоритмика – Система команд процессора целевой машины

3. Границы	<i>Квалификация</i> – Синтаксис ЯП, особенности программирования на нем, библиотеки – Способы адресации памяти целевой платформы – Технологии разработки компиляторов – Принципы организации, состав и схема работы ОС 9 – Принципы управления ресурсами
4. Системность	<i>Внештатные ситуации</i> 10 – Стандарты взаимодействия систем 11 – Конструкции РИС 12 – Особенности параллельного программирования
Умения	
1. Основа	<i>Действия</i> 13 – Применять языки программирования
2. Фон	<i>Процедуры</i> 4 – Применять языки целевой платформы – Применять технологию разработки компиляторов
3. Границы	<i>Предикаты</i> 16 – Создавать блок-схемы компиляторов и т. п. 17 – Оценка сложности 18 – Создавать блок-схемы алгоритмов функционирования разрабатываемых продуктов
4. Системность	<i>Технологии</i> 19 – Кодирование компилятора 20 – Отладка компилятора
Навыки	
Основа	<i>Опыт</i> 21 – 1 год
2. Фон	<i>Сценарии</i> 22 – Получение ТД
3. Границы	<i>Квалификация</i> 23 – Представлять блок-схемы 24 – Методики тестирования – Разработка эксплуатационной документации – Оценивать сложность алгоритма функционирования – Осуществлять отладку программ для целевой ОС
4. Системность	<i>Внешние носители</i> 28 – Локальные правовые акты, законодательство 29 – ГосСтандарты – ЕСПД
Понимание	
1. Основа	<i>Управление</i> 30 – Блок-схемы
2. Фон	<i>Структуры</i> 31 – Методы и основные этапы трансляции 32 – Изучение ТД
3. Границы	<i>Цели</i> 33 – Разработка, отладка, модификация и поддержка системного ПО
4. Системность	<i>Поиск решений</i> 34 – Английский язык на уровне чтения документации
Развитие	
1. Основа	<i>Комплекты</i>

	35 – Отладка программ
2. Фон	<i>Иерархия</i> 36 – Сопровождение компиляторов
3. Границы	<i>Варианты</i> 37 – Отладка программ на целевой ОС 38 – Реинжиниринг разработанных компиляторов
4. Системность	<i>Обмен данными</i> 39 – Разработка документации

Можно сравнивать требования по квалификационной сложности, представляя в клетках таблиц число формулировок как в табл. 8.

Таблица 8

Сравнение квалификационной сложности трудовых функций специалистов уровня 6 «Программист» и «Системный программист» (компиляторы и т. п.)

6-й уровень стандарта на профессию						Стандарт на профессию «Системный программист» (компиляторы и т. п.)					
17	Знания	Умения	Навыки	Понимание	Развитие	38	Знания	Умения	Навыки	Понимание	Развитие
1	1	1	1	1	1	1	1	1	1	1	1
2	1	1	0	0	1	2	2	2	1	2	1
3	1	1	1	1	1	3	5	3	5	1	2
4	1	1	1	1	0	4	3	2	2	1	1

Можно обратить внимание, что разница в числе требований сосредоточена на требованиях «понимание», «развитие», «границы» и «системность», важнейших для решения задач современных ИТ.

3. Образовательное расширение требований к квалификации

В 2018–2019 учебных годах факультет информационных технологий (ФИТ) НГУ открыл новую образовательную программу подготовки бакалавров «Компьютерные науки и системотехника. Computer Science and System Design» в рамках направления 09.03.01 Информатика и вычислительная техника. Новая программа выстроена на основе актуальных международных принципов и практики образования в этой области. Программа сфокусирована на ранней мотивации студентов и опережающем развитии у них востребованных профессиональных навыков.

В соответствии с требованиями актуального федерального государственного образовательного стандарта (ФГОС3++), новая программа бакалавриата базируется на ряде профессиональных стандартов, утвержденных Министерством труда и социальной защиты РФ. Программа подготовки бакалавров нацелена на реализацию значимых принципов: создание условий для развития и профессионального роста; гибкость системы подготовки, учитывающая инновационный и научный векторы интересов студентов, их индивидуальные предпочтения и интересы. В рамках такого подхода студенты представлены как будущие профессионалы с самого первого дня; весь процесс обучения направлен на их профессиональную практику; для области Computer Science практическая подготовка включает, прежде всего, дизайн и реализацию обеспечения компьютерных систем, в том числе программного, аппаратного [2].

4. Предложение к реорганизации стандарта

Учитывая происходящее в мировой практике экстенсивное появление проблемно-ориентированных ЯП, необходимо для разработчиков компиляторов в определении стандарта выделить следующие уровни рабочих функций.

Системный программист. Уровень 6. Системный программист. Разработка простого компилятора с известного языка на массово используемую архитектуру, допускающая применение готовых инструментов по автоматизации процесса компиляции, подобных LEX или Clang-LLVM. Требования: 1, 3, 5, 7, 13, 18, 22, 24, 30–32, 34–36.

Старший системный программист. Уровень 7. Разработка многоуровневого компилятора, требующего создания специальных промежуточных языков для преодоления семантической дистанции между реализуемым языком и целевой архитектурой. Требования: –37.

Ведущий системный программист. Уровень 8. Разработка оптимизирующего компилятора, требующего глубокого знания целевой архитектуры и теоретической схематологии для достижения необходимого уровня результирующего кода. Требования: 4, 8, –34, 37–39.

Главный системный программист. Уровень 9. Разработка высокопроизводительного компилятора для производства программ, ориентированных на параллельные вычисления и распределенные системы, подчиненные требованиям особо высокой надежности и безопасности. Требования: 9, 12, 17, 26, 28, 34, 37–39.

Заключение

Описанные решения по анализу трудовых функций для определения профессиональных стандартов можно рассматривать как конкретизацию понятийной сложности по Колмогорову [3] в применении к разным видам квалификации в области ИТ. Предложенные формы при оценке сложности и трудоёмкости деятельности можно дополнить разделением требований к постановкам задач по сферам применения и качеству результатов. Разброс сложности квалификационных требований допускает выделение в стандарте на профессию «Системный программист» уровней с 7-го по 9-й, особенно учитывая требования ФГОС к компетенциям по пониманию изучаемого материала и формированию способностей к развитию специалиста. Следует отметить, что фактический список трудовых функций может быть существенно дополнен.

Список литературы

1. Портал Федеральных государственных образовательных стандартов высшего образования. 2019. URL: <http://fgosvo.ru/docs/101/69/2/6>.
2. Лаврентьев М. М., Шафаренко А. В., Шадрин М. Ю., Мигинский Д. С., Держо М. А. Преподавание математики для студентов направления Computer Science // Тезисы докладов Пятой междунар. конф., посвященной 95-летию со дня рождения члена-корр. РАН, акад. Европейской академии наук Л. Д. Кудрявцева. М. : РУДН, 26–29 ноября 2019.
3. Колмогоров А. Н. Три подхода к определению понятия «количество информации» // Проблемы передачи информации. Т. 1. Вып. 1. 1965. С. 3–11.

THE ISSUE OF CAREER PROSPECTS IN IT SECTOR**M. M. Lavrentyev**

Doctor of Physics and Mathematics, Prof., Novosibirsk State University,
630090, Novosibirsk, Russia, Pirogova Str., 2,
Institute of Automation and Electrometry, Siberian Branch of the Russian Academy of Sciences,
630090, Novosibirsk, Russia, Acad. Koptuyug pr., 1, e-mail: mmlavrentiev@gmail.com

L. V. Gorodnyaya

PhD, associate professor, A. P. Ershov Institute of Informatics Systems, Siberian Branch of the Russian
Academy of Sciences,
630090, Novosibirsk, Russia, Acad. Lavrentjev pr., 6,
Novosibirsk State University,
630090, Novosibirsk, Russia, Pirogova str., 1, e-mail: lidvas@gmail.com

M. A. Derzho

Associate professor, Novosibirsk State University,
630090, Novosibirsk, Russia, Pirogova str., 1, e-mail: m_derjo@mail.ru

D. S. Miginsky

PhD, associate professor, A. P. Ershov Institute of Informatics Systems, Siberian Branch of the Russian
Academy of Sciences,
630090, Novosibirsk, Russia, Acad. Lavrentjev pr., 6,
Novosibirsk State University,
630090, Novosibirsk, Russia, Pirogova str., 1, e-mail: dmiginsky@gmail.com

Abstract. The purpose of the article is to analyze the standards for professional qualifications in the field of IT from the point of view of training complex professions and the formation of a professional body. Formulated a proposal to clarify the standard for the profession “system programmer” for developers of compilers.

Keywords: Training in system programming, creation of compilers, programming paradigms, semantic systems, conceptual complexity of activity, professional qualification

References

1. Portal Federal'nyh gosudarstvennyh obrazovatel'nyh standartov vysshego obrazovaniya. 2019. URL: <http://fgosvo.ru/docs/101/69/2/6>. (in Russian)
2. *Lavrentyev M. M., Shafarenko A. V., Shadrin M. Y., Miginsky D. S., Derzho M. A.* Prepodavanie matematiki dlya studentov napravleniya Computer Science // Tezisy dokladov Pyatoy mezhdunar. konf., posvyashchennoi 95-letiyu so dnya rozhdeniya chlena-korr. RAN, akad. Evropejskoj akademii nauk L. D. Kudryavceva. M. : RUDN, 26–29 November, 2019. (in Russian)
3. *Kolmogorov A. N.* Tri podhoda k opredeleniyu ponyatiya “kolichestvo informacii” // Problemy peredachi informacii. Vol. 1. Iss. 1. 1965. P. 3–11.

Графическое моделирование на базе Eclipse в системах управления

М. В. Платонова

студент, Санкт-Петербургский государственный университет,
198504, Санкт-Петербург, Россия, Университетский пр., 28, e-mail:
platonova.maria@outlook.com

Аннотация. Тренд цифровизации только несколько лет назад появился в российском обществе и уже активно внедряется во всевозможные его сферы. Как показывают исследования, цифровизация тесно связана с визуализацией, что легко заметить на примере популярности графического представления бизнес-процессов.

В данном тексте приведено описание работы по моделированию и последующему исполнению процессов производства, таких как документооборот и работа со станками с ЧПУ, с использованием среды разработки Eclipse и стандарта BPMN.

Ключевые слова: цифровизация, визуализация, генерация кода, бизнес-процесс, документооборот, станки с ЧПУ, bpmn, eclipse

Введение

Информационные революции, происходившие на протяжении всей истории человечества, несомненно значительно изменяли жизнь общества, его уклады и устои. Конечно, такие перемены влияют и на экономическую сферу. Так, например, в современном мире превалирует цифровая экономика [1].

Уже несколько лет Россия уверенно внедряет использование возможностей цифровизации во многие сферы общества и показывает отличные результаты, которые заметны на мировой арене [2]. Однако необходимо заметить, что такие процессы подразумевают развитие собственной технологической базы, удовлетворяющей требованиям к качеству и надежности продуктов. Кроме того, в будущем это позволит российским компаниям, специализирующимся на создании ПО, уверенно выходить на мировой рынок, занимая лидирующие позиции.

На наш взгляд, применение цифровизации начинается с явного представления бизнес-процессов, так как именно такие процессы лежат в основе работы любой компании [3], и соответственно, необходимо описывать их в удобном для использования виде.

Несомненно, стоит рассмотреть графическое представление процессов, поскольку визуализация способствует снижению усилий, требуемых для понимания, дальнейшего обсуждения и корректировки необходимой информации [4]. Существует международный стандарт BPMN (Business Process Model and Notation), предназначенный для моделирования бизнес-процессов [5]. С его помощью любой процесс может быть представлен в виде диаграмм, где элементами являются какие-либо действия, соответствующие этому процессу. Кроме того, использование диаграмм значительно упрощает генерацию исполняемой программы, что необходимо для многих процессов.

Главная цель данной работы — создание средства для визуализации процессов и последующей генерации исполняемого кода. Так как для разных процессов исполняемые программы могут различаться, в данной работе ставится задача о генераторе для трех конкретных типов: документооборот, работа со станками с числовым программным управлением (далее ЧПУ) и управление поставками и рисками.

На кафедре Системного программирования СПбГУ (далее – СП) уже проводилась схожая работа по созданию средства для графического описания бизнес-процессов с

возможностью автоматически генерировать код [6]. Данное средство является частью metaCASE-технологии на базе QReal. Заметим, что применение metaCASE оправдано в случае создания не одного, а сразу нескольких продуктов.

Основываясь на предыдущем опыте, можно заметить, что использование мало распространенного, сложного или имеющего много ограничений инструментария может привести к проблемам с развитием и поддержкой продукта, в отличие от его аналогов. В данной работе используется популярная среда разработки Eclipse — бесплатная, кроссплатформенная система, постоянно развивающаяся и, кроме того, обладающая большим сообществом разработчиков, что облегчает решение возникающих проблем. Более того, важно учитывать различия open-source лицензий при использовании уже готовых продуктов. Например, GPL - General Public License [7] подразумевает сохранение статуса open-source для нового продукта, т. е. не разрешает создание на его основе коммерческих продуктов. Другая лицензия – Apache 2.0 [8] – не выставляет такого требования или какого-либо другого, ограничивающего применение. В связи с этим в данной работе применяются сторонние продукты только под этой лицензией.

В завершение нужно обратить внимание на важность удобства использования разработанного инструмента. На кафедре СП проводилось исследование человеко-компьютерного взаимодействия для средств визуального программирования [9]. Полученные результаты произвели на нас сильное впечатление, поскольку простые изменения в пользовательском интерфейсе помогли увеличить производительность в разы. По этой причине методы и рекомендации, приведенные в этом исследовании, учтены в данной работе.

1. Графический редактор

Основной задачей графического редактора в данной работе является построение диаграмм с использованием нотации BPMN, при этом подразумевается, что возможности редактора не ограничиваются лишь визуальным представлением процессов, с его помощью также можно обеспечить хранение логической модели, описываемой в виде графа.

Полностью реализовать графический редактор отнюдь не просто, а так как существуют различные варианты готовых решений, нам показалось разумным выбрать подходящую версию существующего редактора и доработать ее. Среди BPMN редакторов на базе Eclipse было выбрано приложение Activiti-Designer (далее Activiti) как популярное, развивающееся и имеющее подходящую лицензию [10].

Activiti состоит из холста и панели элементов, на которой представлено 42 элемента, разделенных на девять категорий: Start Event, End Event, Task, Container, Gateway, Boundary Event, Intermediate Event, Artifacts и Connection. Более того, редактор позволяет выравнивать элементы на диаграммах, а также сохранять полученные диаграммы как изображения. На рис. 1 приведен пример бизнес-процесса, описанного с использованием данного редактора.

Activiti обладает некоторыми недостатками, например, не всегда можно изменить имя или характеристики элемента, используя диаграмму. Кроме того, нельзя не упомянуть, что возникло множество ошибок, связанных с иерархией модулей исходного кода, решением этой проблемы стало добавление модулей в определенном порядке.

Activiti обладает не всеми желаемыми и необходимыми функциями, поэтому было принято решение о расширении возможностей редактора. Рассмотрим подробнее добавленные функции.

1.1. Быстрый доступ

Важной доработкой приложения стало добавление функции «быстрого доступа», что позволяет отображать на холсте элементы, не перемещая их из панели элементов.

Пользователю предоставляется набор комбинаций “CTRL+M CTRL+N”, где M — это номер категории, где расположен элемент, а N — это номер элемента внутри категории. Таким

образом, применяя какую-либо комбинацию клавиш из данного набора, соответствующий элемент добавляется на диаграмму в то место, где находится курсор мыши.

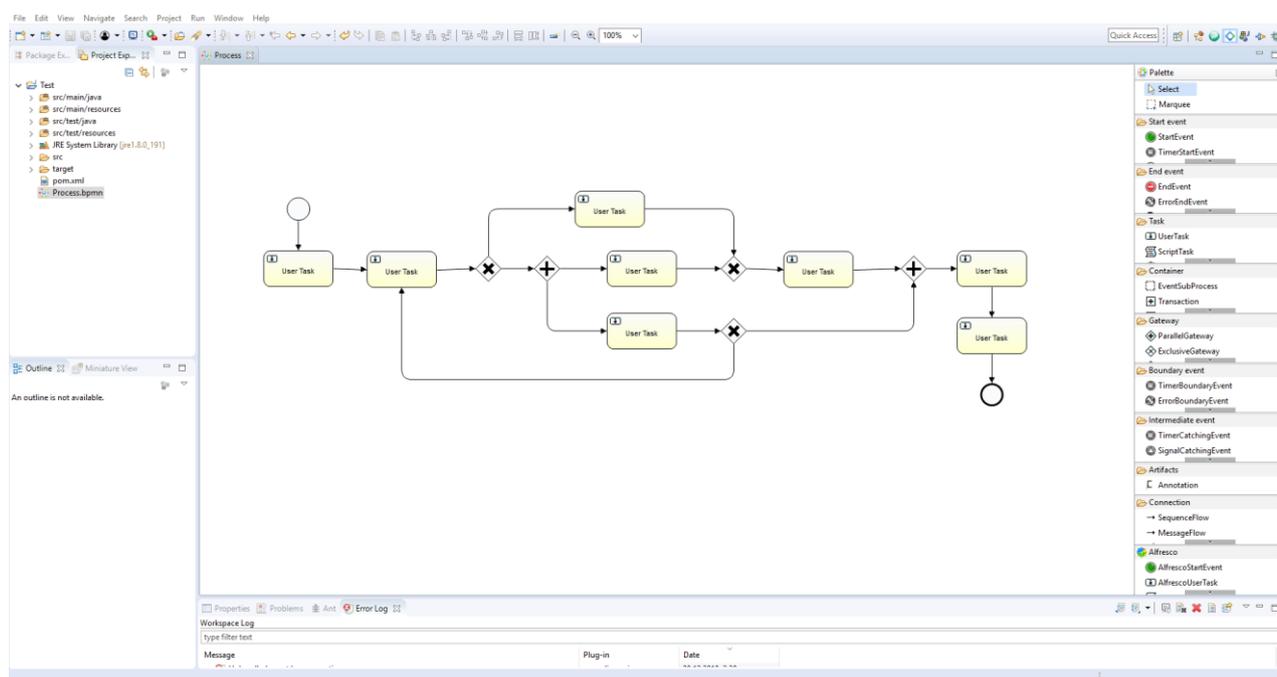


Рис. 1. Пример использования Activiti

1.2. Видимость элементов

Функция «видимость элементов», которая отсутствует в Activiti, также является значимым расширением возможностей редактора. Она позволяет пользователю скрывать элементы контейнера, что применимо в случае громоздких диаграмм и высокого уровня вложенности.

Пользователю предоставляется две функции: `minimize` и `maximize`, одна из которых добавляется в контекстное меню контейнера в зависимости от того, скрыты его элементы или нет. Таким образом, применяя функцию `minimize`, все элементы контейнера скрываются, а сам контейнер представляется в виде небольшого прямоугольника, сохраняя при этом все внешние связи, которые присутствовали в его полномасштабном варианте, а при применении функции `maximize` контейнер принимает свой изначальный вид.

2. Логическая модель

Визуальное, т. е. создаваемое пользователем представление бизнес-процесса, принято называть графической моделью, а внутреннее представление в виде, например, графа специального вида, необходимого для генерации отчетов, исполняемых программ и т. д., — логической моделью.

В данной работе одной из основных задач является выбор представления логической модели. Удобный и быстрый доступ к данным значительно упрощает последующие действия, связанные с генерацией исполняемой программы и для разбора и анализа рассматриваемой модели.

Основной причиной хранения информации в виде ориентированных графов (далее — граф) являлась простота преобразования диаграмм пользователей в графы, и, кроме того, на их ребрах и вершинах удобно хранить какую-либо дополнительную информацию.

Кроме того, из-за отсутствия каких-либо ограничений на диаграммы возрастает сложность работы с графами. Например, множество циклов и петель мешают определять

начальные и конечные вершины описанного процесса. Также это усложняет анализ переходов между элементами, поэтому мы приняли некоторые ограничения, направленные на повышение структурности моделей.

Учитывая особенности требуемого графа, была выбрана определенная модель его представления. Рассмотрим подробнее полученную структуру.

2.1. Базовый тип

Каждый элемент графа имеет уникальный идентификатор (далее – `id`) и дополнительную информацию об элементе (далее – `info`). `Id` задается при создании элемента и записывается в `info`. Кроме того, в `info` хранится имя, тип и множество других компонентов элемента. Невозможно не заметить важность этих данных, и именно поэтому `id` и `info` легли в основу базового класса — `BaseItem`.

2.2. Типы данных в графе

Граф состоит из элементов трех типов: `Container` — контейнер, `Node` — узел, `Edge` — ребро. Соответствующие им классы предоставляют всю необходимую информацию об элементах и их связях с другими элементами графа. Заметим, что хранение всей информации может быть избыточным и неэффективным, поэтому в качестве связей между элементами используется только `id`.

Важно пояснить, что такое контейнер и для чего он нужен. Контейнер предназначен для хранения любых элементов графа, в нем может лежать сколько угодно других контейнеров. Контейнер может также использоваться в качестве вершины.

Каждый элемент диаграммы содержит информацию о контейнере, в котором он находится (далее родительский контейнер), причем ребро может иметь два таких контейнера, поскольку его начало и конец могут находиться в разных контейнерах. Базовым контейнером является холст диаграммы, что означает наличие родительского контейнера для любого элемента графа.

2.3. Представление графа в памяти

За представление графа отвечают отдельные классы — `NodeUtil`, `EdgeUtil` и `ContainerUtil`. Эти классы являются статическими, что позволяет не создавать отдельный экземпляр при каждом применении, а использовать один и тот же для всех обращений.

Каждый класс отвечает за актуальный список соответствующих элементов. Модификация классов происходит при любых изменениях диаграммы, например, при добавлении, изменении или удалении элемента.

3. Примеры применения

3.1. Легкий документооборот

В любой компании важно отслеживать ее документы и проверять их статус, т. е. во время всего жизненного цикла конкретного документа необходимо знать, где и в каком состоянии он находится.

Такой процесс удобно представить в виде BPMN диаграммы (рис. 2). Диаграмма позволит без особых усилий отслеживать документ и его текущий статус, так как его движение происходит по построенному графу. Важно учитывать, что во многих компаниях существует система иерархии, и таким представлением движения документа можно контролировать, чтобы доступ к документу имели только сотрудники с требуемыми полномочиями.

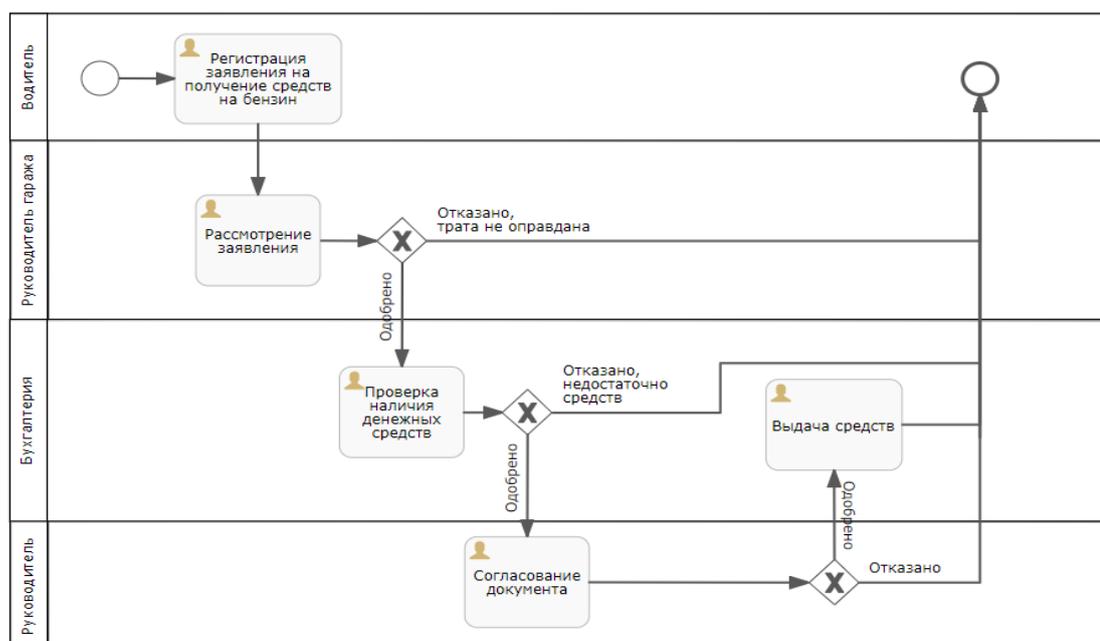


Рис. 2. Пример применения для легкого документооборота

3.2. Работа на станках с ЧПУ

Последние годы на фабриках становится все больше станков с ЧПУ [11], что значительно повышает автоматизацию производства и позволяет гарантировать качество результата.

Полная автоматизация таких процессов возможна с использованием в дополнение к станкам с ЧПУ складских роботов (далее – робот), например, робота-тележки, робота-манипулятора и других. Их основная задача — это передача необходимых материалов и инструментов от одного станка к другому.

Возникает вопрос: как программировать станки с ЧПУ и контролировать работу роботов? Важно не только задать всю последовательность действий, но и описать реакцию на сбойные ситуации, так как, например, из-за случайного мусора или неровностей поверхности робот может сбиться с пути или упасть, что способно вызвать нарушение корректной работы системы или даже поломку перевозимого груза.

Как работа станков с ЧПУ, так и работа роботов может быть описана в виде BPMN-диаграмм, и, как видно на примере рис. 3, таким способом удобно задавать в том числе и ветки отказов роботов, что позволит быстро реагировать на проблемы и восстанавливать работу производства.

3.3. Управление поставками и рисками

Анализ поставщиков и связанных с ними рисков является важной задачей в любой компании, так как сбои поставок могут существенно повлиять на ход производства, например, вызвать изменение сроков или увеличение бюджета. Кроме того, заметим, что поставщики представляют собой иерархическую структуру, поэтому необходимо контролировать всю систему, ведь проблема может возникнуть на любом уровне иерархии.

Одним из способов описать такую структуру является использование BPMN-диаграмм. Это позволит внедрить автоматический анализ данных, использовать накопленную статистику по каждому поставщику и их влияние на процесс целом, что значительно уменьшит пользовательские действия и влияние человеческого фактора.

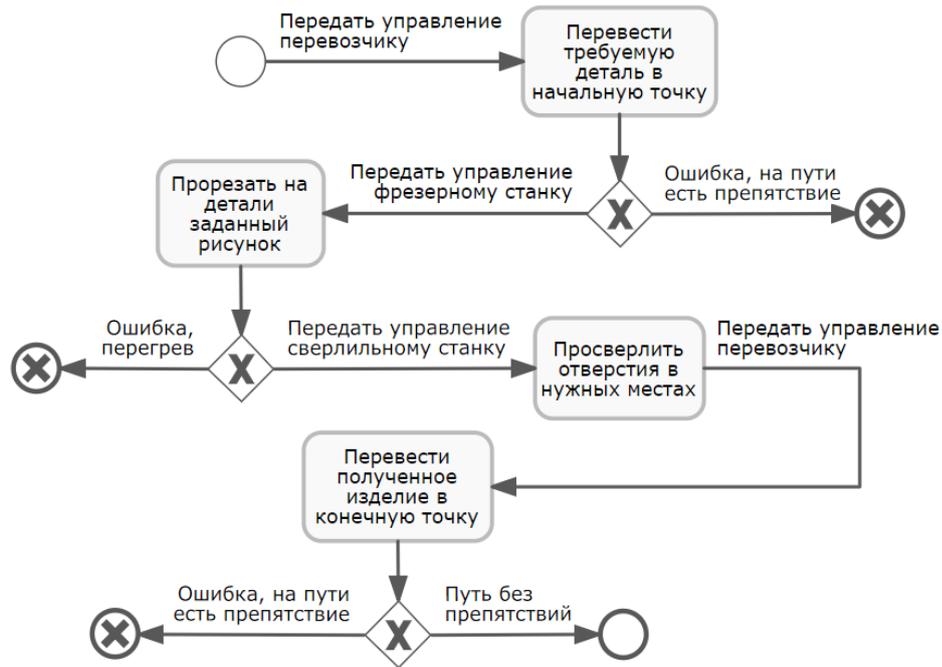


Рис. 3. Пример применения для работы на станках с ЧПУ

4. Кодогенератор

Исполняемые программы разных типов производственных процессов могут существенно различаться по структуре и конечному результату. В связи с этим генераторы кода реализуются только (пока!) для процессов, описанных в предыдущем разделе. Каждый такой процесс представляется в виде диаграммы, по которой строится логическая модель (далее – модель) с последующей генерацией исполняемого кода.

Основным этапом, предшествующим кодогенерации, является анализ и разбор модели. Этот промежуточный этап необходим для выявления начальных и конечных узлов, циклов, петель и т. д. Данные, полученные на этом этапе, являются входными для генератора кода, поэтому безусловно полезно иметь возможность их визуализировать, так как благодаря этому могут быть обнаружены различные неточности и ошибки, упущенные во время построения диаграммы.

Пользователю предоставляется функция `showModel`, которая вызывается нажатием кнопки “Show model”, расположенной в основном меню. Заметим, что эту функцию можно применить для любой диаграммы, даже если дальнейшая кодогенерация невозможна.

На рис. 4 приведен пример диаграммы, для которой на рис. 5 представлена визуализация ее модели.

Таким образом, применяя эту функцию для какой-либо диаграммы, открывается отдельное окно, где все ее элементы отображаются в конкретном виде. Важно напомнить, что «ребенок» узла — это узел, являющийся конечным для какого-либо ребра, выходящего из него, и что контейнер может также выступать в качестве узла, т. е. может как иметь «детей», так и быть «ребенком».

Для каждого контейнера, включая базовый, в окне отображаются все соответствующие ему элементы: узлы и их «дети». Более того, если элемент — контейнер, то он выделяется квадратными скобками ([]), а если элемент — шлюз (Gateway), то он выделяется треугольными скобками (< >). Кроме того, все контейнеры, расположенные в данном, дополнительно отображаются отдельно от остальных элементов. Связи между элементами представляются в определенном виде: элемент1 -> ребро -> элемент2.

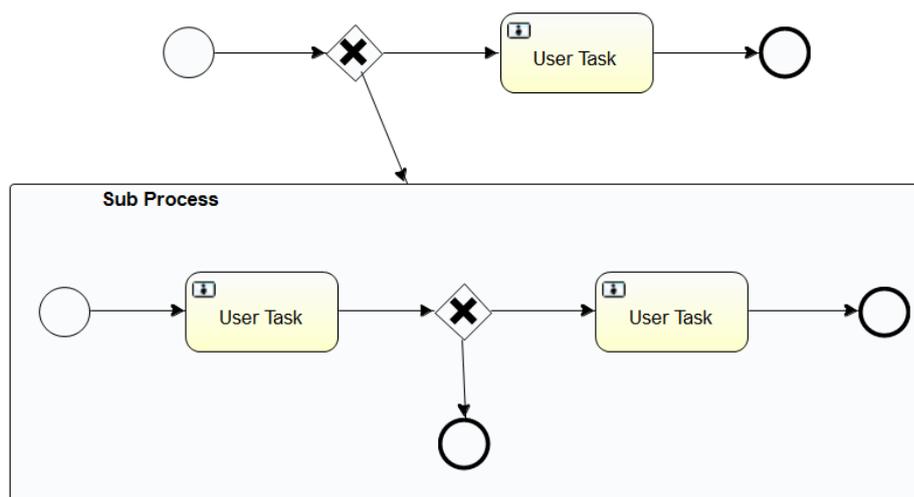


Рис. 4. Пример диаграммы

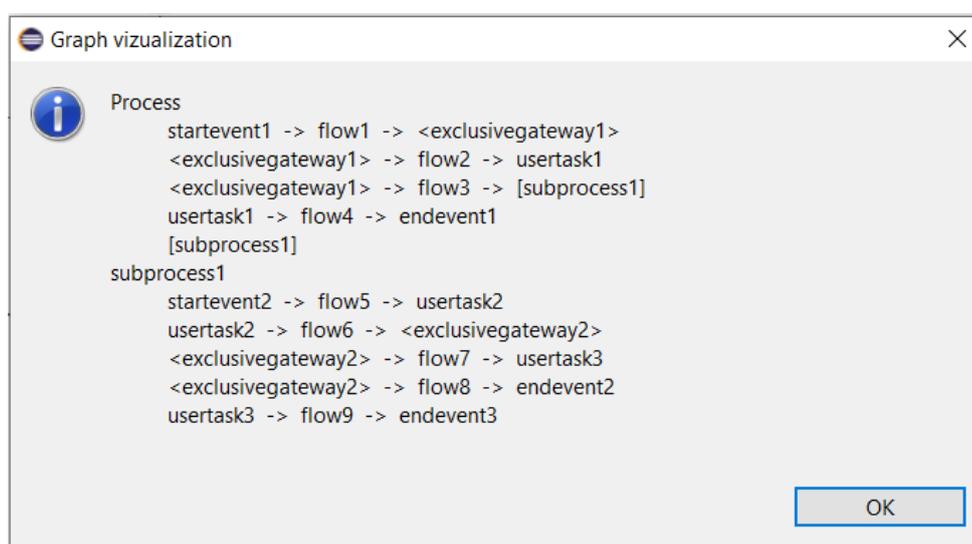


Рис. 5. Визуализация логической модели

Имея такое представление логической модели, задача генерации кода становится простой – линейный обход с генерацией нужного фрагмента кода по каждому действию.

Заключение

В данном тексте приведено описание работы по созданию средства для графического представления производственных процессов с возможностью генерации исполняемой программы. Основная часть работы, как и большая часть текста, посвящена доработке графического редактора Activiti и созданию логической модели, на основе которой генерируется исполняемая программа. Кроме того, важно учесть, что каждому этапу разработки предшествовал теоретический экскурс, поэтому значительную часть времени заняло изучение необходимого материала.

В заключение хочется добавить, что данная работа является актуальной и востребованной, поэтому уже в ближайшем будущем планируется применение разработанного продукта в реальных производственных задачах.

Список литературы

1. Хусьяинов Т. М. Информационная революция и трансформация занятости. М. : Наука. Мысль, 2017.
2. Chakravorti B., Chaturvedi R. S. Digital Planet 2017: How Competitiveness and Trust in Digital Economies Vary Across the World. Tufts University. The Fletcher School, 2017. URL: https://sites.tufts.edu/digitalplanet/files/2017/05/Digital_Planet_2017_FINAL.pdf.
3. Плотников В. А. Цифровизация производства: теоретическая сущность и перспективы развития в российской экономике. М. : Известия СПбГЭУ, 2018.
4. Афанасьев А. А. Технология визуализации данных как инструмент совершенствования процесса поддержки принятия решений. М. : Инженерный вестник Дона, 2014.
5. Business Process Model And Notation. Object Management Group (OMG). 2011. URL : <https://www.omg.org/spec/BPMN/2.0/> (дата обращения: 10.03.2019).
6. QReal:BP. QReal. 2013. URL: <http://qreal.ru/static.php?link=QRealBP> (дата обращения: 10.03.2019).
7. General Public License. Free Software Foundation. 2007. URL: <https://www.gnu.org/licenses/gpl-3.0.en.html> (дата обращения: 10.03.2019).
8. Apache 2.0. Apache Software Foundation. 2004. URL: <https://www.apache.org/licenses/LICENSE-2.0> (дата обращения: 10.03.2019).
9. Брыскин Т. А. Платформа для создания специализированных визуальных сред разработки программного обеспечения: автореф. дисс. ... канд. техн. наук. СПб, 2015. 16 с.
10. Activiti-Designer. URL: <https://github.com/Activiti/Activiti-Designer> (дата обращения: 10.03.2019).
11. Ловыгин А. А., Теверовский Л. В. Основы числового программного управления. Современный станок с ЧПУ и CAD/CAM система. М., 2015.

UDK 004.94

GRAPHICAL MODELING OF CONTROL SYSTEMS BASED ON ECLIPSE TECHNOLOGIES

M. V. Platonova

Student, St. Petersburg State University,
198504, Saint-Petersburg, Russia, Universitetsky pr., 28, e-mail: platonova.maria@outlook.com

Annotation. Only a few years ago Russia took the path of the digitalization and this trend is already taking active root into various areas of society. Digitalization is closely related to visualization, that is easily confirmed by the increasing popularity of graphical representation of business processes.

This text describes the work on modeling and subsequent execution of production processes, such as document circulation, working with CNC machines and analysing production risks, by using the Eclipse IDE and the BPMN standard.

Keywords: digitalization, visualization, code generation, business process, document circulation, CNC machine, bpmn, eclipse

References

1. Chusyainov T. M. Informacionnaya revolyuciya i transformaciya zanyatosti [Information revolution and employment transformation]. М. : Nauka. Mysl', 2017. (in Russian)

2. *Chakravorti B., Chaturvedi R. S.* Digital Planet 2017: How Competitiveness and Trust in Digital Economies Vary Across the World. Tufts University. The Fletcher School, 2017. URL: https://sites.tufts.edu/digitalplanet/files/2017/05/Digital_Planet_2017_FINAL.pdf.
3. *Plotnikov V. A.* Cifrovizaciya proizvodstva: teoreticheskaya sushchnost' i perspektivy razvitiya v rossijskoj ehkonomike [Digitalization of production: theoretical nature and development prospects in the Russian economy]. M. : Izvestiya UNECON, 2018. (in Russian)
4. *Afanas'ev A. A.* Tekhnologiya vizualizacii dannyh kak instrument sovershenstvovaniya processa podderzhki prinyatiya reshenij [Data visualization technology as a tool for improving decision-making process]. M. : Inzhenernyj vestnik Dona, 2014. (in Russian)
5. Business Process Model And Notation. Object Management Group (OMG). 2011. URL : <https://www.omg.org/spec/BPMN/2.0/>.
6. QReal:BP. QReal. 2013. URL: <http://qreal.ru/static.php?link=QRealBP>.
7. General Public License. Free Software Foundation. 2007. URL: <https://www.gnu.org/licenses/gpl-3.0.en.html>.
8. Apache 2.0. Apache Software Foundation. 2004. URL: <https://www.apache.org/licenses/LICENSE-2.0>.
9. *Bryskin T. A.* Platforma dlya sozdaniya specializirovannyh vizual'nyh sred razrabotki programmogo obespecheniya: avtoref. diss. ... kand. tekhn. nauk. SPb, 2015. 16 p.
10. Activiti-Designer. URL: <https://github.com/Activiti/Activiti-Designer>.
11. *Lovygin A. A., Teverovskij L. V.* Osnovy chislovogo programmogo upravleniya [Fundamentals of numerical control]. Sovremennyy stanok s CHPU i CAD/CAM Sistema. M., 2015. (in Russian)

Применение генетического алгоритма для генерации входных данных при тестировании программного кода

К. Е. Сердюков

аспирант, Новосибирский государственный технический университет,
630073, Новосибирск, Россия, пр-т К. Маркса, 20, e-mail: zores@live.ru

Т. В. Авдеенко

д-р техн. наук, проф., Новосибирский государственный технический университет,
630073, Новосибирск, Россия, пр-т К. Маркса, 20, e-mail: tavdeenko@mail.ru

Аннотация. В данной статье предлагается метод определения наборов данных для тестирования программного обеспечения. В статье предлагаются результаты исследования по оценке одного пути программного кода, работа еще не окончена, в дальнейшем будет расширена для подбора данных для тестирования множества путей. Разработка программного обеспечения является сложным процессом, в котором высока вероятность допущения различного рода ошибок. Несмотря на попытки автоматизировать процесс тестирования, полная автоматизация возможна не для всех этапов, например, для формирования наборов данных для последующего тестирования. Сложность автоматизации этого этапа связана с тем, что данные сложно подобрать из-за большого числа возможных комбинаций. Поэтому для решения данной задачи особую ценность представляют эвристические подходы, которые не перебирают все возможные варианты, например, генетические алгоритмы. Использование генетического алгоритма позволяет найти достаточно большое число комбинаций, каждая из которых будет проверять определенный путь. В алгоритм возможно вносить изменения, что обеспечит нахождение максимально разнообразных частей программного кода для более высокого уровня покрытия кода. Под покрытием понимается проверка наибольшего числа различных операций в коде. В данной статье исследуются возможности разработанного метода генерации тестовых данных на основе генетического алгоритма.

Ключевые слова: генетический алгоритм, тестирование, автоматическая генерация тестов

Введение

Программная инженерия является комплексным, систематическим подходом к разработке и сопровождению программного обеспечения. При разработке программ чаще всего выделяют следующие этапы – анализ, проектирование, программирование и тестирование. На этапе анализа определяются требования к программному обеспечению и производится документирование. На этапе проектирования детализируется внешний вид программы, определяется ее внутренний функционал, разрабатывается структура продукта и вводятся требования для последующего тестирования. Написание исходного кода программы на одном из языков программирования производится на этапе программирования.

Одним из наиболее важных шагов при разработке программных продуктов является тестирование. Важными целями тестирования являются соответствие разработанной программы заданным требованиям, соблюдение логики в процессах обработки данных и получение верных конечных результатов. Поэтому для тестирования очень важно сгенерировать входные данные, на основе которых программа будет проверяться на наличие ошибок и соответствие заданным требованиям. Для оценки качества входных данных используется показатель покрытия кода, т. е. какой процент всей программы предложенные

тестовые наборы могут «покрыть». Определяется отношением тестируемых операций ко всему количеству операций в коде.

Некоторые процессы тестирования программного кода совершенствуются достаточно медленно. Разработка большинства видов сценариев тестирования чаще всего происходит вручную, без использования каких-либо систем автоматизации. Из-за этого процесс тестирования становится невероятно сложным и затратным как по времени, так и по финансам, если подходить к нему со всей серьезностью. Для тестирования некоторых программ может уходить до 50 % всех временных затрат.

Одной из основных целей тестирования является создание такого тестового набора, который обеспечивал бы достаточный уровень качества конечного продукта за счет проверки большинства различных путей программного кода, т. е. обеспечивал бы его максимальное покрытие. Тем не менее сама задача поиска многих путей состоит из нескольких подзадач, решение которых необходимо для нахождения качественного набора данных. Одной из локальных задач, решаемых для поиска тестового набора, является определение одного, наиболее сложного пути кода.

По большей части валидация и верификация программных продуктов плохо поддается оптимизации. Особенно сложно автоматизировать генерацию тестовых данных, которая по большей части делается вручную.

Существуют исследования, в которые предлагается решения проблемы подбора данных или рассматриваются возможности автоматизации процесса тестирования при помощи других методов. В статье [1] для решения данной проблемы предлагается использовать логическое программирование в ограничениях (анг. Constraint Logic Programming) и символьное исполнение. В [2] используется правила обработки ограничений (анг. Constraint Handling Rules, CHR) для помощи в ручной верификации проблемных мест в программе.

Некоторые исследователи применяют эвристические методы для автоматизации процесса тестирования, например, диаграмму потоков данных (анг. Data-flow diagram). Исследования методов автоматизации с использованием данной диаграммы были проведены в статьях [3, 4, 5, 6]. В статье [3] предлагается дополнительно использовать генетические алгоритмы для определения новых входных наборов тестовых данных на основе ранее использованных.

В статьях [7, 8] рассматриваются гибридные автоматизированные системы генерации тестовых данных. В [7] применяется подход, объединяющий стратегии случайного поиска (анг. Random Strategy, RS), динамического символьного исполнения (анг. Dynamic Symbolic Execution, DSE) и стратегий на основе поиска (анг. Search-Based Strategy, SBS). В статье [8] предлагается теоретическое описание поисковой стратегии тестирования с применением генетического алгоритма. Рассматриваются подходы поиска локальных и глобальных экстремумов на реальных программах. Предлагается гибридный подход генерации тестовых данных – меметический алгоритм (анг. Memetic Algorithm).

Использование генетических алгоритмов в процессе тестирования позволяют обеспечить нахождение наиболее сложных частей программы, в которых риски из-за допущения ошибок наиболее велики. Исследования, предложенные в данной статье, основаны на работах [9, 10, 11]. Оценивание каждого пути происходит за счет использования функции приспособленности, в качестве параметров которой выступают веса каждой отдельной операции.

В данной статье предлагается использовать генетический алгоритм для создания наборов данных. Это позволит автоматизировать генерацию тестов, что существенно снизит нагрузку на весь этап тестирования. Предложенный метод позволит обеспечить формирование наборов данных для программ в режиме реального времени. Метод разрабатывается с учетом различных настроек, что позволит подобрать данные с различными требованиями и возможностью различным образом определять важные пути программного кода для концентрации на определенных частях кода. Это позволит сделать процесс тестирования более гибким и направленным. Статья состоит из двух частей. В части первой вводится

терминология и определяется способ расчета весов операций для последующего использования в функции приспособленности. В части второй предлагаются результаты работы алгоритма и анализ производительности при изменении настроек генетического алгоритма.

1. Применение генетического алгоритма для оценки кода

В качестве способа определения входных данных используется генетический алгоритм. Генетический алгоритм использует собственную терминологию, которая в рамках текущей задачи определяется следующим образом – в качестве хромосом выступают варианты подобранных значения для тестирования. Множество тестовых вариантов в одной итерации определяются как популяция.

Весы для последующей оценки и расчета функции приспособленности считаются следующим образом:

- 1) первой операции назначается вес, например, в 100 единиц;
- 2) каждой последующей операции также назначается вес – если нет никаких условий или циклов, вес приравнивается весу предыдущей операции;
- 3) условия разделяют вес в соответствии с правилом – если условие содержит только одну ветвь (только `if...`), то вес каждой операции снижается на 80 %. Если условие разделяется на несколько ветвей (`if...else...`), то вес делится на равнозначные части – для двух ветвей 50 % / 50 %, для трех 33 % / 33 % / 33 % и т. д.;
- 4) веса операций в цикле остаются, но также могут умножаться на определенный вес, если необходимо увеличить значимость циклов при тестировании;
- 5) все вложенные ограничения учитываются, например, для двух вложенных условий вес операций будет равен $80 \% * 80 \% = 64 \%$.

Назначенные веса можно использовать для того, чтобы разрабатывать тестовые варианты при помощи генетических алгоритмов, т. е. для оценки того, какой вес приходится на тот или иной путь при определенных значениях входных параметров.

Функция считается следующим образом:

–X – наборы данных;

–F(X) – значение функции приспособленности для каждого набора данных в зависимости от рассчитанных значений весов.

Задача заключается в максимизации целевой функции при суммировании всех весов каждого набора, т. е. $F(X) \rightarrow \max$.

В предложенном методе первая популяция формируется случайным образом. Для проверки были заданы определенные настройки генетического алгоритма – в каждой популяции содержится по 100 хромосом; общее число популяций также равняется 100. Это позволит сформировать достаточное число различных вариантов.

2. Результаты использования метода

В таблице представлены несколько результатов работы метода, где первая популяция задается случайным образом. В таблице показывается первая популяция, две средние популяции, 20 и 50, для контроля расчета функции и финальная. В таблице представлено только по пять «лучших» хромосом в популяции. Наилучшая хромосома в финальной популяции является решением.

Как можно заметить, в каждом из вариантов сформировалось как минимум два различных набора тестовых данных, при которых операции в рассматриваемом программном коде будут иметь наибольший вес. Кроме того, можно увидеть определенные закономерности в результатах – первое значение всегда максимально (случайные значения максимальным ограничивались значением 100 для увеличения сходимости), второе значение меньше, чем первое, но больше третьего.

Сравнение результатов запуска метода

Популяция	Вариант 1	Вариант 2	Вариант 3	Вариант 4
0	1: 78, 23, 35 2: 62, 36, 95 3: 52, 35, 27 4: 17, 77, 73 5: 75, 9, 96	1: 97, 3, 6 2: 82, 77, 64 3: 24, 47, 57 4: 90, 13, 82 5: 81, 69, 24	1: 92, 97, 28 2: 38, 66, 52 3: 63, 76, 64 4: 7, 24, 56 5: 57, 48, 8	1: 15, 67, 26 2: 32, 27, 83 3: 37, 52, 64 4: 70, 49, 64 5: 67, 29, 94
20	1: 95, 64, 54 2: 95, 64, 29 3: 95, 64, 54	1: 97, 80, 4 2: 97, 80, 53 3: 97, 80, 28	1: 99, 13, 10 2: 99, 13, 11 3: 99, 13, 11	1: 99, 71, 45 2: 99, 71, 15 3: 99, 71, 3
50	1: 95, 64, 54 2: 95, 64, 29 3: 95, 64, 54	1: 97, 80, 29 2: 97, 80, 4 3: 97, 80, 53	1: 99, 13, 10 2: 99, 13, 11 3: 99, 13, 11	1: 99, 71, 60 2: 99, 71, 3 3: 99, 71, 3
Финальная (100)	1: 95, 64, 54 2: 95, 64, 29	1: 97, 80, 4 2: 97, 80, 29	1: 99, 13, 10 2: 99, 13, 11	1: 99, 71, 60 2: 99, 71, 45

Для данного программного кода можно использовать тестовые наборы данных, полученных в последнем поколении. Приоритетность зависит от полученного ранга.

Таким образом, используя генетические алгоритмы можно найти такие начальные тестовые начальные значения, которые бы в полной мере проверяли все варианты программы в зависимости от назначенных весов. Это проверяется через максимизацию функций приспособленности, так как наиболее широко описывающий тестовый вариант проходит через те ветви, которые имеют наибольший вес.

Задачу по определению наиболее подходящих тестовых данных очень сложно решить при помощи стандартных алгоритмов, так как фактическое количество элементов массива возможных значений невероятно огромно.

Использование генетических алгоритмов позволяет сравнивать множество различных вариантов данных для тестирования программы. Широкие возможности к усовершенствованию позволяет увеличить количество начальных тестовых вариантов, количество поколений и добавить новые свойства, благодаря которым можно существенно увеличить возможности нахождения более подходящих вариантов. Если отслеживать пройденные узлы и снижать веса тех, которые наиболее часто встречаются в различных вариантах, можно обеспечить поиск новых путей, которые на данный момент могут не иметь высокого веса, но могут быть так же важны, как и наиболее часто встречающиеся.

Для дополнительного анализа проверим, как именно меняется скорость работы алгоритма в зависимости от настроек ГА.

На графиках ниже можно увидеть, каким образом меняется длительность работы программы в зависимости от размера популяции, т. е. количества хромосом, и от общего

количества популяций. При исследовании изменения размера популяций количество популяций приравнялось к 100. И наоборот, при исследовании зависимости от количества популяций, число хромосом равнялось 100.

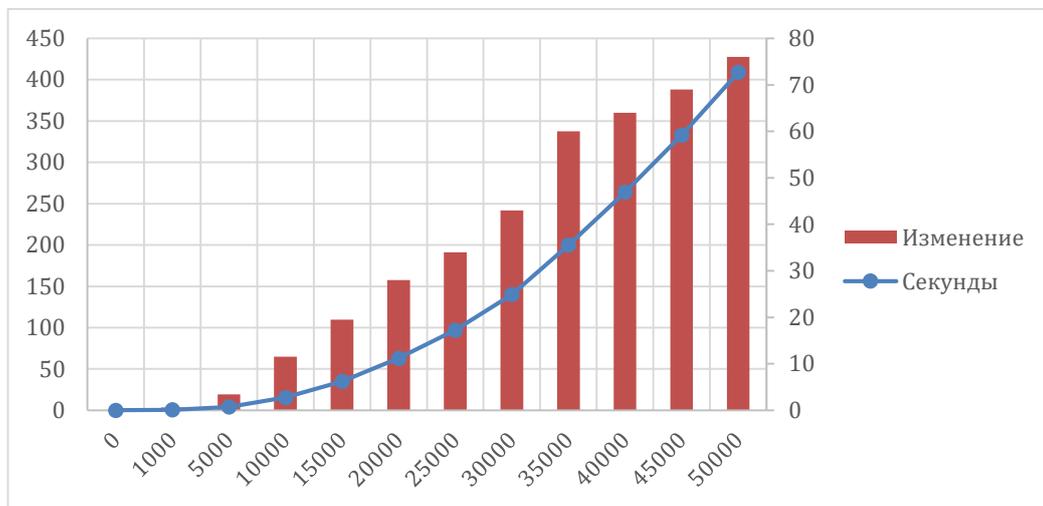


Рис. 1. Зависимость длительности выполнения от размера популяций

На рис. 1 показывается зависимость от количества хромосом в популяции. Основываясь на нем, можно сделать вывод, что при увеличении количества хромосом в популяции длительность работы существенно увеличивается.

На рис. 2 отражена зависимость от количества популяций.

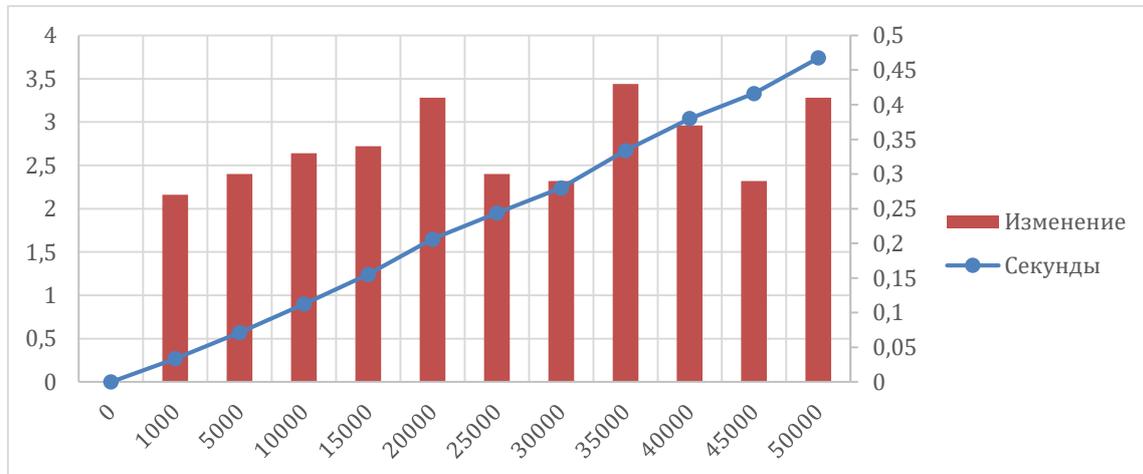


Рис. 2. Зависимость длительности выполнения от количества популяций

Становится очевидно, что при изменении количества популяций увеличивается длительность работы алгоритма, но уже в линейной прогрессии. При этом заметны колебания прироста в обе стороны, сохраняющиеся примерно на одном уровне.

Несмотря на то, что и в одном, и в другом случае общее количество хромосом оставалось одинаковым, изменения в длительности работы существенно различаются. Увеличение размера популяции существенно увеличивает длительность работы — число хромосом в популяции с 1 000 до 50 000 увеличилось в 50 раз, но длительность возросла в 750. При изменении количества популяций в 50 раз привело к увеличению времени лишь в 14 раз.

Связано это с тем, что наиболее сложные с точки зрения загрузки операции происходят при расчете функции приспособленности и при поиске оптимальных для скрещивания

хромосом. Именно из-за того, что поиск лучших хромосом зависит от числа хромосом в популяции, то текущий алгоритм поиска, т. е. сортировки от лучших к худшим, существенно загружает мощности компьютерных систем и увеличивает скорость работы экспоненциально размеру популяции.

Количество хромосом в одной популяции позволяет обеспечить разнообразие вариантов, т. е. с большей вероятностью найти более подходящие варианты. Увеличение количества популяций приводит к более точному результату, но только при большом количестве хромосом. Если хромосом будет достаточно мало, то алгоритм быстро придет к одному повторяющемуся значению.

Исходя из-за всего вышесказанного, наибольшее влияние на результат оказывает число хромосом в одной популяции, но при этом при увеличении этого количества существенно увеличивается нагрузка. Но чтобы обеспечить более качественный конечный результат, при увеличении числа хромосом следует увеличивать общее число популяций, что также влияет на загрузку.

Заключение

Эволюционные методы работают таким образом, чтобы находить наиболее хорошие решения в задачах, которые невозможно или слишком затратно решать стандартными методами оптимизации. Они не всегда работают быстро или качественно, но в задачах с нестандартными подходами показывают превосходство.

Метод подбора входных данных на основе генетического алгоритма позволит автоматизировать метод подбора входных данных, при этом существенно увеличив скорость поиска данных. За исключением настроек ограничений, алгоритм полностью автоматизирован, поэтому не требует дополнительных тестировщиков или разработчиков. Полученный на выходе набор данных может напрямую использоваться для в процессе тестирования и, при необходимости, быть собран заново без дополнительных затрат.

В дальнейшем планируется провести исследование влияния различных метрик на конечный результат и величину покрытия кода, для обеспечения таких наборов данных, которые бы позволили протестировать код максимально эффективно и с максимальным числом операций.

Список литературы

1. Christophe Meudec ATGen: Automatic Test Data Generation using Constraint Logic Programming and Symbolic Execution // Software Testing Verification and Reliability, 2001. Vol. 11, Iss. 2. P. 81–96. URL: <https://doi.org/10.1002/stvr.225>.
2. Ralf Gerlich Automatic Test Data Generation and Model Checking with CHR // 11th Workshop on Constraint Handling Rules, 2014. URL: <https://arxiv.org/pdf/1406.2122.pdf>.
3. Moheb R. Girgis Automatic Test Data Generation for Data Flow Testing Using a Genetic Algorithm // Journal of Universal Computer Science. 2005. Vol. 11, № 6. P. 898–915.
4. Weyuker E. J. The complexity of data flow criteria for test data selection // Inf. Process. Lett. 1984. Vol. 19, № 2. P. 103–109.
5. Khamis A., Abdelaziz R. Automatic test data generation using data flow information // Dogus University Journal. 2011. Vol. 2. P. 140–153.
6. Singla S., Kumar D., Rai H. M., Singla P. A hybrid pso approach to automate test data generation for data flow coverage with dominance concepts // Journal of Advanced Science and Technology. 2011. Vol. 37. P. 15–26.
7. Zicong L., Zhenyu C., Chunrong F. Qingkai Shi Hybrid Test Data Generation // State Key Laboratory for Novel Software Technology. ICSE Companion 2014 Companion Proceedings of the 36th International Conference on Software Engineering. 2014. P. 630–631.

8. *Harman M., McMinn P.* A Theoretical and Empirical Study of Search-Based Testing: Local, Global, and Hybrid Search / *IEEE Transactions on Software Engineering*. 2010. Vol. 36(2). P. 226–247.
9. *Myers G. J., Sandler C., Badgett T.* The Art of Software Testing. New Jersey, 2012. 256 c.
10. *Crispin L., Gregory J.* Agile Testing: A Practical Guide for Testers and Agile Teams. Boston, USA : Pearson Education, 2010. 576 c.
11. *Praveen R. S., Tai-hoon K.* Application of Genetic Algorithm in Software Testing // *International Journal of Software Engineering and Its Applications*. 2009. Vol. 3, № 4. P. 87–96.

UDK 519.711.2

AUTOMATIC GENERATION OF TEST DATA BASED ON GENETIC ALGORITHM**K. Serdyukov**

Graduate student, Novosibirsk State Technical University,
630073, Novosibirsk, Russia, K. Marks ave., 20, e-mail: zores@live.ru

T. Avdeenko

Doctor of Engineering, Prof., Novosibirsk State Technical University,
630073, Novosibirsk, Russia, K. Marks ave., 20 e-mail: tavdeenko@mail.ru

Annotation. This article proposes a method for defining datasets for software testing. The article presents the results of a study on the evaluation of a single path of program code, the work is in progress, and will be further expanded to select data for testing a variety of paths. Software development is a complex process where possibility of making various kinds of errors is high. Despite attempts to automate the testing process, this is not possible for all phases, for example, to define data sets for subsequent testing. The complexity of the automation of this stage is due to the fact that the data is difficult to pick up due to the large number of possible combinations. Therefore, to solve this problem, heuristic approaches that do not go through all possible options, for example, genetic algorithms, are of particular value. Using a genetic algorithm allows to find a sufficiently large number of combinations, each of which will check a particular path. It is possible to make changes to the algorithm, which will ensure finding the most diverse parts of the program code for a higher level of code coverage. Coverage refers to checking the largest number of different operations in a code. This article explores the possibilities of the developed method for generating test data based on a genetic algorithm.

Keywords: Genetic algorithm, testing, automatic test generation

References

1. . Christophe Meudec ATGen: Automatic Test Data Generation using Constraint Logic Programming and Symbolic Execution // *Software Testing Verification and Reliability*, 2001. Vol. 11, Iss. 2. P. 81–96. URL: <https://doi.org/10.1002/stvr.225>.
2. Ralf Gerlich Automatic Test Data Generation and Model Checking with CHR // 11th Workshop on Constraint Handling Rules, 2014. URL: <https://arxiv.org/pdf/1406.2122.pdf>.
3. *Moheb R. Girgis* Automatic Test Data Generation for Data Flow Testing Using a Genetic Algorithm // *Journal of Universal Computer Science*. 2005. Vol. 11, № 6. P. 898–915.
4. *Weyuker E. J.* The complexity of data flow criteria for test data selection // *Inf. Process. Lett.* 1984. Vol. 19, № 2. P. 103–109.
5. *Khamis A., Abdelaziz R.* Automatic test data generation using data flow information // *Dogus University Journal*. 2011. Vol. 2. P. 140–153.

6. *Singla S., Kumar D., Rai H. M., Singla P.* A hybrid pso approach to automate test data generation for data flow coverage with dominance concepts // *Journal of Advanced Science and Technology*. 2011. Vol. 37. P. 15–26.
7. *Zicong L., Zhenyu C., Chunrong F.* Qingkai Shi Hybrid Test Data Generation // *State Key Laboratory for Novel Software Technology. ICSE Companion 2014 Companion Proceedings of the 36th International Conference on Software Engineering*. 2014. P. 630–631.
8. *Harman M., McMinn P.* A Theoretical and Empirical Study of Search-Based Testing: Local, Global, and Hybrid Search / *IEEE Transactions on Software Engineering*. 2010. Vol. 36(2). P. 226–247.
9. *Myers G. J., Sandler C., Badgett T.* *The Art of Software Testing*. New Jersey, 2012. 256 c.
10. *Crispin L., Gregory J.* *Agile Testing: A Practical Guide for Testers and Agile Teams*. Boston, USA : Pearson Education, 2010. 576 c.
11. *Praveen R. S., Tai-hoon K.* Application of Genetic Algorithm in Software Testing // *International Journal of Software Engineering and Its Applications*. 2009. Vol. 3, № 4. P. 87–96.

Инструментарий для исследования лексических характеристик текста

Е. А. Сидорова

канд. физ.-мат. наук, ст. науч. сотр.,
Институт систем информатики им. А. П. Ершова СО РАН,
630060, Новосибирск, Россия, пр. Акад. Лаврентьева, 6, e-mail: lsidorova@iis.nsk.su

Аннотация. Рассматривается среда для многоцелевого исследования лексических характеристик текста. Среда включает интерфейсы для разработки словарей и построения системы признаков, а также средства автоматической генерации лексического наполнения словаря по текстам. При извлечении терминов осуществляется морфологический анализ текста и построение словосочетаний на основе правил согласования грамматических характеристик слов. Термины снабжаются частотными характеристиками на основе обработки обучающего корпуса текста. Для исследования контекстов употребления терминов предоставляются средства построения конкордансов, что позволяет пользователю пронаблюдать грамматические, семантические, стилистические и проблемно-ориентированные особенности терминов и осуществить их разметку.

Ключевые слова: предметный словарь, термин, конкорданс

Введение

Текст как источник и средство передачи информации нуждается во всестороннем исследовании, необходимом как для оценки «качества» изложенного, так и при автоматической обработке и поддержке языковых поисковых сервисов. Изучение языковых явлений и моделирование процессов понимания текста на разных языковых уровнях является фокусом современных исследований в компьютерной лингвистике.

Для решения данных задач, как правило, используют разнообразные знания в формализованном виде, такие как тезаурусы (WordNet, RusNet), толково-комбинаторные словари, аннотированные корпуса текстов (например, Национальный корпус русского языка www.ruscorpora.ru) и т. п. Тезаурус, как инструмент описания предметной лексики, позволяет характеризовать термин и его связи с точки зрения особенностей употребления в данной предметной области [1]. Другим вариантом исследования языковых явлений является использование корпусов текстов. Корпус является источником и инструментом многоаспектных лексикографических работ [2], позволяя на основе разметки автоматизировать создание и начальное наполнение словарей.

Анализ литературы [3–5] показывает, что при извлечении терминологии из большого массива текстов используются подходы, объединяющие лингвистические и статистические методы. Результатом такого рода исследований может стать аннотированный корпус текстов, который позволит в дальнейшем осуществлять частотный анализ лексики, создавать конкордансы по различным основаниям и строить электронные словари. Использование специализированных методов позволит автоматизировать работу экспертов по исследованию формальных структур и дискурса в целом и построению лингвистических моделей на основе размеченного корпуса текста.

Данная работа посвящена описанию инструментов, обеспечивающих исследование лексических характеристик текста на основе корпусов текстов. Совокупность предложенных инструментов позволяет сформировать среду для создания проблемно-ориентированных словарей и предоставить конечному пользователю различные возможности исследования языковых явлений.

1. Требования к среде поддержки исследования текста

Разработка моделей и создание качественных ресурсов требует кропотливого ручного труда, поддерживаемого программным инструментарием. Программная среда должна предоставлять специалистам различные рабочие инструменты для формирования необходимых баз знаний и проведения корпусных исследований.

В рамках данной работы были сформулированы следующие функциональные требования к системе:

- 1) обеспечивать автоматическое наполнение словарей на базе корпусов текстов;
- 2) предоставлять возможность настраивать и приписывать различные характеристики терминам словаря;
- 3) осуществлять лексический анализ текста – сегментацию и извлечение из текста заданных в словаре терминов;
- 4) обеспечивать накопление данных о статистико-комбинаторных свойствах обнаруженных в тексте языковых явлений;
- 5) осуществлять построение конкордансов терминов и их визуализация.

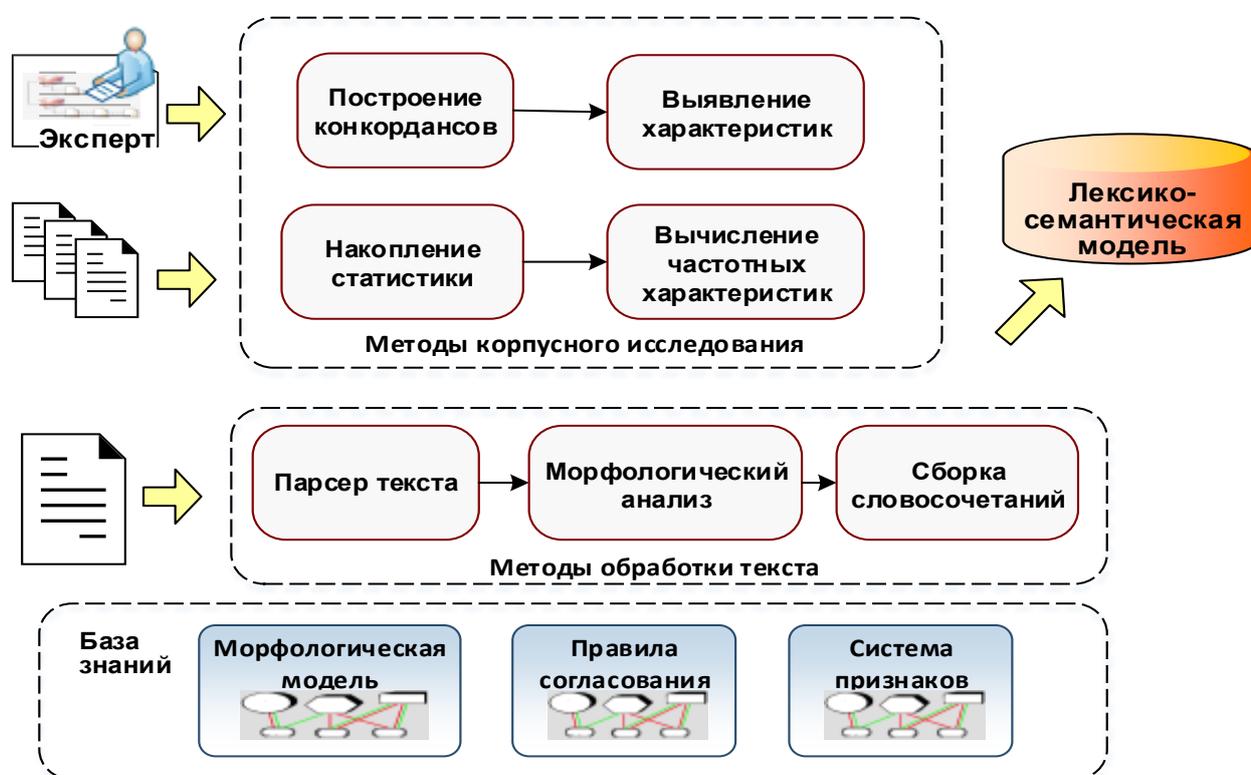


Рис. 1. Среда исследования лексических характеристик текста

Созданная система включает следующие базовые инструменты для проведения исследований (рис. 1): интерфейс разработки словаря и построения системы признаков, средства автоматической генерации лексического наполнения словаря по корпусу текстов и вычисления количественных характеристик найденных терминов, инструменты построения конкордансов для исследования контекстов лексических единиц.

2. Модель представления знаний

Рассматриваемая лексикографическая модель знаний включает три основных компонента. Словарь задает лексическую модель рассматриваемого подязыка, определяемого проблемной областью. Грамматика обеспечивает поиск и извлечение

лексических единиц из текстов. Система прагматически-ориентированных характеристик, задаваемая пользователем, поддерживает фиксацию наблюдений и ориентирована на дальнейшую поддержку автоматизированных методов обработки текста.

Базой исследований является представительный проблемно-ориентированный корпус текстов. Основным инструментами, обеспечивающими поддержку исследований, является поиск примеров употребления терминов словаря, построения множества контекстов (конкордансов), вычисление количественных характеристик встречаемости, совместной встречаемости, распределения и др.

2.1. Лексическая модель

В рамках нашего подхода в словарной статье хранится необходимая информация как для извлечения термина из текста, так и для поддержки последующих этапов анализа.

Проблемно-ориентированный словарь – это объем лексики, организованной по семантическому (тематическому / жанровому / и др.) принципу с отражением определенного набора базовых формальных отношений. Формально словарь описывается системой вида:

$$V = \{W, P, M, G, S, F_w, F_p\}, \text{ где}$$

W – множество лексем; каждой лексеме сопоставлена информации обо всей совокупности ее форм,

P – множество многословных терминов, описываемых парой $\langle N\text{-грамма, тип структуры} \rangle$, где N -грамма задает последовательность лексем, а тип структуры определяет вершину и правила согласования элементов N -граммы,

M – морфологическая модель языка, включающая описание морфологических классов и признаков;

G – множество правил согласования для извлечения многословных терминов,

S – проблемно-ориентированная система признаков для разметки терминов,

$F_w = W \rightarrow 2^{M \times S}$, $F_p = P \rightarrow 2^{G \times S}$ – функции, сопоставляющие терминам наборы признаков.

Особенностью морфологического представления в рамках системы, является возможность ее изменения в зависимости от задачи, решаемой с помощью словаря. Пользователь может сформировать собственный набор признаков, классов и обеспечить их интеграцию посредством правил сопоставления с базовым морфологическим представлением. Морфологический класс определяется частью речи, набором лексических признаков (например, одушевленность или род у существительных) и типом парадигмы. Потребность в изменении классов возникает достаточно редко, например, когда используются дополнительные специализированные словари терминов (словари имен, географических названий) или есть потребность включить в словарь слова другого языка.

Другой важной особенностью системы является поддержка многословных терминов – словосочетаний, сформированных по правилам, реализующих поверхностно-синтаксический анализ. Большинство многословных терминов включают от двух до четырех слов и формируются с помощью правил вида $P+C$ (*аналоговый датчик*) – согласование существительного и прилагательного, $C+Срд$ (*автор учебника*), $P+P+C$ (*новая информационная технология*), $C+Прд+Срд$ (*обработка естественного языка*) и т. п. Встречаются также термины с более сложной структуры, например, с зависимыми предложными группами $C+Предл+C$ (*резервуар с жидкостью*), $C+Предл+C+C$ (*поиск в пространстве состояний*) и т. д.

2.2. Система признаков

В зависимости от решаемой проблемы, термины словаря могут снабжаться различными типами признаков: статистическими (для решения задач классификации), жанровыми (для жанрового анализа текста), семантическими (для семантического анализа), формальными (для выявления маркеров определенных структур) и др.

Статистические признаки накапливают информацию о частоте появления термина в обрабатываемых текстах. Для решения задач классификации текста необходимо, чтобы пользователь задал систему связанных между собой тем, а также наличие обучающего корпуса текстов, т. е. корпуса размеченного заданными признаками. В словаре для каждого термина хранится: встречаемость в обучающей выборке (абсолютная частота) и количество текстов, в которых встречался термин (текстовая частота), список тем, в которых встретился термин, частота и текстовая частота по каждой теме из списка. Часть параметров (относительная частота, $tf*idf$, вес) вычисляются динамически.

Система признаков, используемая для разметки терминов словаря, формируется пользователем и зависит от решаемой задачи, т. е. является проблемно-ориентированной. Для кодирования различной информации о слове (семантической, жанровой, стилистической и др.) предусмотрены следующие возможности.

1. Класс. Термин может быть отнесен к определенному признаку-классу. Иерархия классов позволяет отнести термин к определенному уровню иерархии, более общему или конкретному с наследованием свойств общего класса.

2. Атрибут. Для представления лексического значения термина используются атрибуты. Совокупность семантических значений атрибутов, приписанных слову, в определенной мере моделирует компонентную семантическую структуру слова. Основные компоненты семантической структуры термина могут рассматриваться как тезаурусные дескрипторы.

3. Альтернативные наборы признаков позволяют выразить неоднозначность термина.

3. Модули работы с корпусом текста

Созданная среда включает словарные компоненты и обработчики, которые обеспечивают, с одной стороны, автоматизацию создания, наполнения, редактирования словарей, с другой стороны – применение созданных словарей для лексического анализа текста и последующую работу со словарной информацией найденных в тексте терминов. Важной частью поддержки пользователя является набор поисковых средств, таких как сортировка и фильтрация терминов по различным наборам параметров, визуализация покрытия текста терминами словаря, построение конкордансов терминов с расширяемым окном контекста и др.

3.1. Обучение словаря на основе корпуса текстов

Процесс извлечения терминологии включает такие этапы, как: а) графематический анализ, обеспечивающий токенизацию и выделение нетекстовых элементов и иноязычных вкраплений, б) лексико-морфологический анализ (лемматизация, определение лексико-грамматических признаков, представление парадигмы, нормализация), в) выделение терминоподобных словосочетаний (идентификация на основе предопределенных грамматических моделей и нормализация), г) обновление статистики найденных терминов.

Для создания словарей используются следующие модули.

Морфологический анализ осуществляется на базе модуля «Диалинг» (www.aot.ru), который содержит универсальный словарь русского языка и обеспечивает поиск слова в словаре, определение его грамматических признаков и нормальной формы. Также поддерживается функция предсказания [6], которая по незнакомому слову формирует гипотезы (как правило, около трех вариантов) о его части речи, нормальной форме и других признаках.

Модуль сборки словокомплексов извлекает из текста словосочетания по фиксированному набору грамматических правил. Основной задачей модуля является

выявление наиболее важных терминообразующих синтаксических групп, большинство из которых представляют собой именные группы либо строятся на их основе.

Результатом обработки корпуса текста с помощью данных модулей являются лексическое наполнение словаря с одновременным сбором статистики встречаемости. Если корпус размечен признаками, то соответствующие термины снабжаются данными признаками и относительно признаков также ведется статистика.

Таким образом, среда обеспечивает автоматизацию начального наполнения словарей, на основе которого можно проводить дальнейшие исследования.

3.2. Конкорданс

Конкорданс – традиционный способ изучения корпуса текста. Он дает полный индекс терминов в ближайших и расширенных контекстах, что дает возможность исследователю проверить свою гипотезу относительно функций той или иной лексической единицы. Конкорданс создается для поиска и извлечения образцов целевых единиц, в частности, встречаемости различной модальной, каузальной лексики, речевых предикатов и предикатов пропозиционального отношения. Таким образом, конкорданс осуществляет обратную связь словаря, словарных терминов с корпусом и обеспечивает своего рода лингвистическую разметку на морфологическом и поверхностно-синтаксическом уровне.

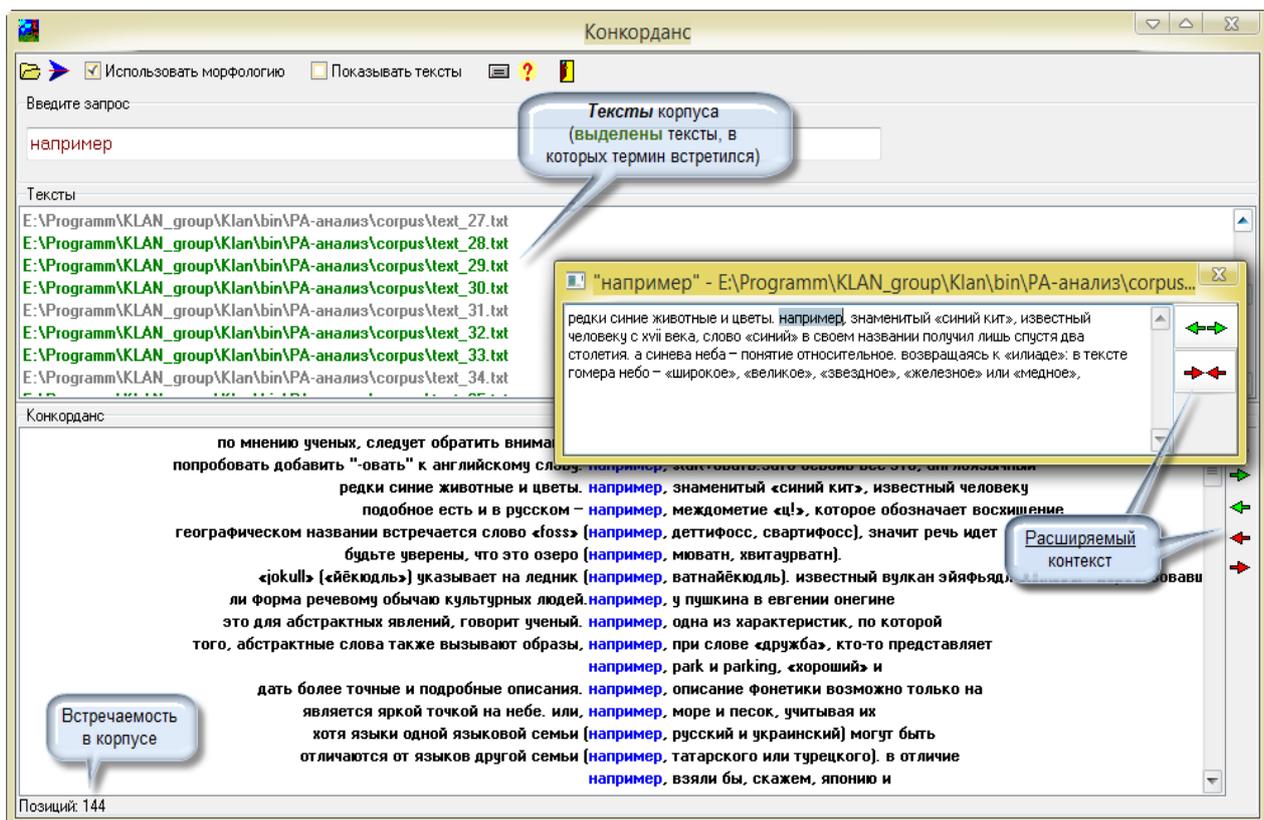


Рис. 2. Модуль конкорданса

Реализованный в системе модуль конкорданса (рис. 2.) работает с текстовыми файлами, каждый из которых может содержать несколько размеченных сообщений. При просмотре контекста вхождения термина пользователь может самостоятельно определять длину просматриваемого фрагмента текста (поддерживается пословное расширение контекста, просмотр абзаца или всего сообщения).

Заключение

Данная работа посвящена описанию подходов и методов разработки лексикографических ресурсов и проведения корпусных исследований для обеспечения полноты и достоверности разрабатываемых моделей. В фокусе внимания создаваемого инструментария являются лексические единицы, выступающие в качестве маркеров и индикаторов объектов более высокого уровня (семантического, прагматического, структурно-жанрового, логико-аргументативного и др.).

Отличительными особенностями рассматриваемой среды является возможность ее многоцелевого использования при решении различных задач анализа текста, интеграция различных возможностей для проведения языковых исследований, обеспечение настройки словарей на проблемную область пользователя, поддержка мультязычности.

Дальнейшее развитие системы заключается в развитии средств корпусного исследования, таких как построение конкордансов совместной встречаемости терминов, использование в поисковых запросах условий на наличие / отсутствие наборов признаков и т. п.

Исследование выполнено при поддержке РФФИ в рамках проектов №17-07-01600 и № 18-00-01376 (18-00-00889).

Список литературы

1. Лукашевич Н. В. Тезаурусы в задачах информационного поиска. М. : МГУ, 2011. 495 с.
2. Sinclair J. Corpus, Concordance, Collocation / Ed. by R. Carter. Oxford: Oxford University Press, 1991. 200 p.
3. Захаров В. П., Хохлова М. В. Автоматическое выявление терминологических словосочетаний // Структурная и прикладная лингвистика. Вып. 10. СПб.: Изд-во СПб. гос. ун-та, 2014. С. 182–200.
4. Bolshakova E., Loukachevitch N., Nokel M. Topic Models Can Improve Domain Term Extraction // International conference on Information Retrieval ECIR-2013. 2013. LNCS-7814. P. 684–687.
5. Митрофанова О. А., Захаров В. П. Автоматизированный анализ терминологии в русскоязычном корпусе текстов // Компьютерная лингвистика и интеллектуальные технологии : труды междунар. конф. «Диалог–2009». М. : Наука, 2009. С. 321–328.
6. Сокирко А. В. Морфологические модули на сайте www.aot.ru // Компьютерная лингвистика и интеллектуальные технологии : труды междунар. конф. «Диалог–2004» / под ред. И. М. Кобозевой, А. С. Нариньяни, В. П. Селегея. М. : Наука, 2004. С. 559–564.

UDK 004.912, 81`32

TOOLS FOR THE STUDY OF LEXICAL CHARACTERISTICS OF TEXT

E. A. Sidorova

Ph.D., senior researcher A. P. Ershov Institute of Informatics Systems,
Siberian Branch of the Russian Academy of Sciences,
630090, Novosibirsk, Russia, Acad. Lavrentjev pr., 6, e-mail: lsidorova@iis.nsk.su

Annotation. The environment for multipurpose research of lexical characteristics of the text is considered. It includes interfaces for developing vocabularies and building a system of features, as well as tools for automatic generation of lexical content of the vocabulary using texts. When extracting terms, the morphological analysis of the text and the construction of a phrase based on the rules of matching the grammatical characteristics of words are carried out. Terms are supplied with

frequency characteristics on the basis of processing of the training corpus of the text. To study the contexts of the use of terms, concordance construction tools are provided. This tools allow the user to observe the grammatical, semantic, stylistic and problem-oriented features of terms and perform their markup.

Keywords: domain vocabulary, terminology, concordance

References

1. Loukachevitch N. V. Tezaurusy` v zadachakh informatcionnogo poiska [Thesauri in information retrieval tasks]. M. : MSU Publ., 2011. 512 p. (in Russian)
2. *Sinclair J.* Corpus, Concordance, Collocation / Ed. by R. Carter. Oxford: Oxford University Press, 1991. 200 p.
3. *Zakharov V. P., Khokhlova M. V.* Automatic Extracting of Terminological Phrases. Strukturnaia i prikladnaia lingvistika = Structural and applied linguistics. № 10. St.-Petersburg : St.-Petersburg University, 2014. P. 182–200. (in Russian)
4. *Bolshakova E., Loukachevitch N., Nokel M.* Topic Models Can Improve Domain Term Extraction // International conference on Information Retrieval ECIR-2013. 2013. LNCS-7814. P. 684–687.
5. *Mitrofanova O. A., Zaharov V. P.* Avtomatizirovanny`i` analiz terminologii v ruskoiazny`chnom korpuse tekstov [Automatic extracting terminological phrases] // Kompyuternaya lingvistika i intellektualnye tekhnologii. Mezhdunarodnoj konferencii Dialog [Computational Linguistics and Intellectual Technologies: Proceedings of the International Conference “Dialog-2009”]. M. : Nauka Publ., 2009. P. 321–328. (in Russian)
6. *Sokirko A. V.* Morfologicheskie moduli na sai`te www.aot.ru [Morphological modules on the site www.aot.ru] // Kompyuternaya lingvistika i intellektualnye tekhnologii. Mezhdunarodnoj konferencii Dialog [Computational Linguistics and Intellectual Technologies: Proceedings of the International Conference “Dialog-2004”]. M. : Nauka Publ., 2004. P. 559–564. (in Russian)

Разработка сертифицированного компилятора предикатных программ

В. И. Шелехов

канд. техн. наук, зав. лабораторией системного программирования,
Институт систем информатики им. А. П. Ершова СО РАН,
630090, Новосибирск, Россия, пр. Акад. Лаврентьева, 6, e-mail: vshel@iis.nsk.su

Аннотация. Компилятор со средствами дедуктивной верификации подлежит тщательной сертификации для подтверждения высокого уровня доверия к результатам дедуктивной верификации. Описывается построение модели внутреннего представления программы в рамках третьего релиза компилятора предикатных программ. Верификация модели и разработка программы компилятора на базе модели обеспечат высокую надежность компилятора.

Ключевые слова: дедуктивная верификация, сертификация программ, модельно-ориентированная технология

Введение

Метод дедуктивной верификации обеспечивает абсолютную гарантию корректности программы относительно ее спецификации. Ввиду большой сложности и трудоемкости этот метод применяется лишь для критических фрагментов программы в приложениях, где требуется сверхвысокая надежность и безопасность.

Дедуктивная верификация намного проще и быстрее для предикатных программ, чем для аналогичных императивных программ. Преимущество по времени верификации более чем в пять раз. Для программы быстрой сортировки с двумя опорными элементами зафиксировано преимущество в 10 раз [1].

В соответствии со стандартами DO178C и ГОСТ Р ИСО/МЭК 15408-3 [2] сертификация программ, для которых применялась дедуктивная верификация, сопровождается сертификацией корректности сопутствующих инструментов и методов. Сертифицируется подсистема верификации вместе с содержащей ее системой программирования, отдельно front-end и back-end генерации кода. Далее в этой цепочке – операционная система, загрузчик программ и, наконец, система команд компьютера. Сертифицируется также сам метод дедуктивной верификации.

Возможность ситуации, когда в процессе компиляции в программу вносится ошибка, принципиально снижает ценность проведенной дедуктивной верификации. Поэтому сертификации компилятора отводится особое внимание. В идеале компилятор должен пройти процедуру дедуктивной верификации. Имеется лишь один такой компилятор. Это оптимизирующий компилятор CompCert [3] с языка Си, доступный с 2015 г. Компилятор состоит из 20 просмотров транслируемой программы и использует 11 промежуточных представлений программы; для каждого представления разработана формальная семантика. Затраты на дедуктивную верификацию программы компилятора CompCert в системе автоматического доказательства Coq [4] – шесть человеко-лет.

Язык предикатного программирования P [5] принципиально сложнее языка Си. Разработка и дедуктивная верификация компилятора для языка P потребовала бы на порядок больше затрат, что за границей текущих возможностей. Есть другая альтернатива – *модельно-ориентированная* технология. Ее применение при разработке компилятора с языка P позволит сертифицировать компилятор высоким уровнем доверия в соответствии со стандартом ГОСТ Р ИСО/МЭК 15408-3 [2].

Модельно-ориентированная технология в системной и программной инженерии базируется на построении и верификации *модели* создаваемого объекта (или программы). В традиционной технологии разработка большой программной системы сопровождается наличием множественных нестыковок и разломов во внутренних интерфейсах системы. В модельно-ориентированной технологии внутренние интерфейсы выстраиваются строго в соответствии с моделью (после ее верификации), что принципиально повышает надежность программной системы. За последние 20 лет сменилось три поколения модельно-ориентированной технологии: от model-based к model-driven, в которой программа генерируется по модели автоматически, к интегрированной модели, единой для создаваемого объекта и встроенного в него программного обеспечения. В модели выделяются различные слои, например, связанные с надежностью, безопасностью и отказоустойчивостью. Реализуется адекватная интеграция различных слоев.

Примером успешного применения модельно-ориентированной технологии является разработка и верификация модели политики безопасности управления доступом в операционных системах [6], в частности, в Astra Linux Special Edition [7]. На языке EventB разработана формальная модель политик безопасности, доказана ее корректность и согласованность. Подтверждена корректность отображения этой модели на ядро ОС Linux размером 2 млн строк кода. Проведена дедуктивная верификация центральной компоненты Linux Security Module. Данная работа обеспечила высокий уровень доверия к безопасности Astra Linux Special Edition, подтвержденный международной организацией Linux Foundation (URL: <http://www.connect-wit.ru/gk-astra-linux-rasshiryayet-sotrudnichestvo-s-the-linux-foundation.html>).

1. Язык предикативного программирования

Предикатная программа относится к классу программ-функций [8] и является предикатом в форме вычислимого оператора $H(x: y)$ с аргументами x и результатами y . Программа имеет две основы: логику и вычислимость, причем логика является первичной, главной. Минимальный полный *базис предикатных программ* определен в виде языка **P₀**. Предикатная программа определяется следующей конструкцией:

<имя предиката>(<аргументы>: <результаты>) { <оператор> }

Пусть x , y и z обозначают разные непересекающиеся наборы переменных. Набор x может быть пустым, наборы y и z не пусты. В составе набора переменных x может использоваться логическая переменная e со значениями **true** и **false**. Пусть B и C – имена предикатов, A и D – имена переменных предикатного типа. Операторами являются: *оператор суперпозиции* $B(x: z); C(z: y)$, *параллельный оператор* $B(x: y) \parallel C(x: z)$, *условный оператор* **if** (e) $B(x: y)$ **else** $C(x: y)$, *вызов предиката* $B(x: y)$ и *оператор каррирования*. В таблице представлен полный базис вычисляемых предикатов и соответствующих им операторов.

Вычисляемые предикаты и их программная форма

$H(x: y) \cong \exists z. B(x: z) \& C(z: y)$	$H(x: y) \{ B(x: z); C(z: y) \}$
$H(x: y, z) \cong B(x: y) \& C(x: z)$	$H(x: y, z) \{ B(x: y) \parallel C(x: z) \}$
$H(x: y) \cong (e \Rightarrow B(x: y)) \& (\neg e \Rightarrow C(x: y))$	$H(x: y) \{ \text{if } (e) B(x: y) \text{ else } C(x: y) \}$
$H(x: y) \cong B(x \sim : y)$	$H(x: y) \{ B(x \sim : y) \}$
$H(A, x: y) \cong A(x: y)$	$H(A, x: y) \{ A(x: y) \}$
$H(x: D) \cong \forall y, z. D(y: z) \equiv B(x, y: z)$	$H(x: D) \{ D(y: z) \{ B(x, y: z) \} \}$
$H(A, x: D) \cong \forall y, z. D(y: z) \equiv A(x, y: z)$	$H(A, x: D) \{ D(y: z) \{ A(x, y: z) \} \}$

Набор $x \sim$ составлен из набора переменных x с возможным добавлением имен предикатных программ. Результатом исполнения оператора каррирования $D(y: z) \{ B(x, y: z) \}$ является новый предикат D , получаемый фиксацией значения набора x .

Для языка P_0 построена формальная операционная семантика $R(H)(x, y)$ и доказано тождество $R(H) = H$ [11]. На базе языка P_0 последовательным расширением [9] и сохранением тождества $R(H) = H$ построен язык предикатного программирования P [5]:

$P_0 \rightarrow P_1 \rightarrow P_2 \rightarrow P_3 \rightarrow P_4 = P$.

Каждый очередной язык в этой цепочке строится на базе предыдущего в виде системы обозначений. Одной из задач построения формальной модели внутреннего представления программы является систематическое построение формальной семантики конечного языка P прохождением через все цепочки обозначений.

Система типов вносится в язык P_4 . Предыдущие языки бестиповые. В языке P имеется развитая система типов: подтипы, структуры, множества, алгебраические типы, предикатные типы, массивы. Тип массива является предикатным типом, его значения (массивы) являются тотальными и однозначными предикатами. Типы могут быть параметризованы переменными.

Эффективность предикатных программ достигается применением следующих *оптимизирующих трансформаций* [10], переводящих программу на язык императивного расширения P_{imp} :

- замена хвостовой рекурсии циклом;
- открытая подстановка программы на место ее вызова;
- склеивание переменных: замена всех вхождений одной переменной на другую переменную;
- кодирование объектов алгебраических типов (списков и деревьев) с помощью массивов и указателей.

В языке P нет циклов и указателей; вместо них используются рекурсивные программы и алгебраические типы данных.

2. Модель внутреннего представления программы

Внутреннее представление (ВП) программы, получаемое в результате работы front-end'а компилятора и являясь функционально эквивалентной копией исходной программы, ориентировано на удобство работы с программой в middle- и back-end'ах. Опыт разработки компиляторов показывает, что структура ВП программы обычно громоздка и неоправданно сложна, содержит много дублирующей информации. Лет тридцать назад было большое число работ по технологии трансляции. В настоящее время разработка компилятора считается рутинной задачей. Однако классическая техника разработки компиляторов не решает проблему построения адекватной структуры ВП.

В настоящей работе описывается метод построения модели ВП, абстрагированной от ее реализации на языке C++. Разрабатываемая модель ВП является частью проекта [12].

2.1. Метаязык

Для описания модели ВП используется язык абстрактных структур, содержащий следующие примитивы:

- произвольные неупорядоченные наборы (множества) объектов;
- последовательности объектов;
- вершины (записи) определенного именованного вида с набором полей (атрибутов);
- абстрактные указатели на объекты;
- понятия, определяющие множество структур некоторого вида.

Например: Prog(Заголовок программы, Область локализации, Тело программы) определяет вершину с именем Prog и тремя полями. Значение каждого поля определяется соответствующим понятием.

2.2. Архитектура внутреннего представления программы

Программа во ВП определяется в виде дерева структур. Точнее, это граф с учетом дуг, определяемых абстрактными указателями. Для всех языковых конструкций используется единое унифицированное представление в виде вершин (записей).

Всякий объект, идентифицируемый некоторым именем в тексте программы, представлен во ВП указателем на описание этого объекта. Описание *именованного объекта* представлено во ВП вершиной определенного вида. Имеются разные виды именованных объектов: простые переменные, типы, предикатные программы, формулы и другие.

Совокупность имен и соответствующих им описаний одного уровня объединены в структуру вида *область локализации*, определяющую отображение имен на описания. Область локализации прикрепляется к некоторой конструкции программы. Например, область локализации, содержащая параметры типа в описании типа, прикрепляется к описанию типа.

Иерархия языковых конструкций определяет иерархию областей локализации, на базе которой проводится *идентификация* – определение соответствующего описания для каждого вхождения имени в программе в процессе семантического анализа. Дополнительно, вне иерархии областей локализации, существуют *автономные области локализации* для полей структур и интерфейсных элементов классов, причем для наследуемых классов реализуется независимая иерархия. Приведенная модель именования объектов является адекватной для проведения идентификации, в том числе и для полиморфных объектов.

Структура операторной части предикатной программы определена в виде дерева сегментов. *Сегмент* – фрагмент программы с одним входом и несколькими выходами. Каждый *выход* связан с некоторым входом другого сегмента, либо с выходом программы. Отметим, что все виды операторов выбора (**switch**, **case**) [5] преобразуются в условные операторы. Вызов функции заменяется во ВП вызовом предиката и корректным образом выносится из выражения, содержащего вызов функции. В итоге во ВП программа возвращается к языку **P₂**.

Во ВП устраняются все умолчания, существующие в языке **P** [5]. Например, вхождение переменной в позиции значения представляется вершиной Value(Переменная). Устраняется полиморфизм операций. Вместо операции «+» используются вершины вида AddNat, AddInt, AddSet, AddLists и др.

Заключение

Чтобы обеспечить высокий уровень доверия к результатам дедуктивной верификации предикатных программ, необходимы соответствующие гарантии корректности разрабатываемой программы компилятора с языка **P**. С этой целью разрабатывается модель ВП программы, на базе которой планируется выстраивать все интерфейсы компилятора.

Цель первого этапа – обеспечить адекватность ВП языку **P** и структурную согласованность частей ВП. Описание структуры ВП в разд. 2.2 и в проекте [12] – это лишь форма (синтаксис) программы, которую необходимо наполнить формализованной семантикой на базе содержательного описания языка **P** [5]. Следующий шаг – построение формальной операционной семантики ВП программы. Полная модель ВП должна пройти процесс ее верификации.

Еще одной задачей является построение модели системы типов языка **P** и ее верификация.

Наиболее сложной задачей следующего уровня является построение модели оптимизирующих трансформаций предикатных программ. Необходимо формально доказать корректность применения всех видов оптимизирующих трансформаций.

Список литературы

1. Шелехов В. И., Чушкин М. С. Верификация программы быстрой сортировки с двумя опорными элементами // Научный сервис в сети Интернет. М. : ИПМ им. М. В.Келдыша, 2018. 26 с. URL: <http://persons.iis.nsk.su/files/persons/pages/dqsort.pdf> (дата обращения: 12.03.2019).
2. ГОСТ Р ИСО/МЭК 15408-3-2013. Информационная технология. Методы и средства обеспечения безопасности. Критерии оценки безопасности информационных технологий. Часть 3. Компоненты доверия к безопасности. М. : Стандартинформ, 2014. 150 с.
3. Kästner D., Wünsche U., Barrho J. et al. CompCert: Practical experience on integrating and qualifying a formally verified optimizing compiler // ERTS 2018: Embedded Real Time Software and Systems. 2018. P. 1–9.
4. Coq Proof Assistant. URL: <https://coq.inria.fr/> (дата обращения: 12.03.2019).
5. Карнаухов Н. С., Перишин Д. Ю., Шелехов В. И. Язык предикатного программирования Р. Новосибирск, 2018. 42 с. URL: <http://persons.iis.nsk.su/files/persons/pages/plang14.pdf> (дата обращения: 12.03.2019).
6. Девянин П. Н., Ефремов Д. В., Кулямин В. В. и др. Моделирование и верификация политик безопасности управления доступом в операционных системах. М. : Горячая линия – Телеком. 2019. 261 с. URL: http://www.ispras.ru/publications/security_policy_modeling_and_verification.pdf (дата обращения: 21.03.2019).
7. Операционные системы Astra Linux. URL: <http://www.astralinux.ru> (дата обращения: 12.03.2019).
8. Шелехов В. И. Классификация программ, ориентированная на технологию программирования // Программная инженерия. 2016. Т. 7, № 12. С. 531–538. URL: <http://persons.iis.nsk.su/files/persons/pages/prog.pdf> (дата обращения: 12.03.2019).
9. Шелехов В. И. Доказательное построение, верификация и синтез предикатных программ // Знания-Онтологии-Теории (ЗОНТ-2017). Новосибирск, 2017. Т. 2. С. 156–165. URL: <http://persons.iis.nsk.su/files/persons/pages/lbase.pdf> (дата обращения: 12.03.2019).
10. Каблуков И. В., Шелехов В. И. Реализация оптимизирующих трансформаций в системе предикатного программирования // Системная информатика. № 11. Новосибирск, 2017. С. 21–48. URL: <http://persons.iis.nsk.su/files/persons/pages/opttransform4.pdf> (дата обращения: 12.03.2019).
11. Шелехов В. И. Семантика языка предикатного программирования // ЗОНТ-15. Новосибирск, 2015. 13 с. URL: <http://persons.iis.nsk.su/files/persons/pages/semZont1.pdf> (дата обращения: 12.03.2019).
12. Шелехов В. И. Проект системы предикатного программирования. Новосибирск, 2018. 24 с. URL: <http://persons.iis.nsk.su/files/persons/pages/project1.pdf> (дата обращения: 12.03.2019).

UDK 004.05

DEVELOPMENT OF CERTIFIED COMPILER OF PREDICATE PROGRAMS

V. I. Shelekhov

Head of System Programming Laboratory,
 A. P. Ershov Institute of Informatics Systems
 Siberian Branch of the Russian Academy of Sciences
 630090, Novosibirsk, Russia, Acad. Lavrentiev, 6, e-mail: vshel@iis.nsk.su

Annotation. A compiler with deductive verification facilities should be thoroughly certified to obtain high level of assurance. The development of the model of inner program representation is

described. The verification of the model and compiler development on the base of the model should guarantee a high level of compiler reliability.

Keywords: deductive verification, program certification, model-oriented design

References

1. *Shelekhov V. I., Chushkin M. S.* Verifikacija programmy bystroj sortirovki s dvumja opornymi jelementami // Nauchnyj servis v seti Internet. M. : IPM im. M. V. Keldysha, 2018. 26 p. (in Russian)
2. GOST R ISO/MJeK 15408-3-2013. Informacionnaja tehnologija. Metody i sredstva obespechenija bezopasnosti. Kriterii ocenki bezopasnosti informacionnyh tehnologij. Chast' 3. Komponenty doverija k bezopasnosti. M. : Standartinform, 2014. 150 p. (in Russian)
3. *Kästner D., Wünsche U., Barrho J. et al.* CompCert: Practical experience on integrating and qualifying a formally verified optimizing compiler // ERTS 2018: Embedded Real Time Software and Systems. 2018. P. 1–9.
4. Coq Proof Assistant. URL: <https://coq.inria.fr/>.
5. *Karnauhov N. S., Pershin D. Ju., Shelekhov V. I.* Jazyk predikatnogo programirovanija P. Novosibirsk, 2018. 42 p. (in Russian)
6. *Devjanin P. N., Efremov D. V., Kuljamine V. V. et al.* Modelirovanie i verifikacija politik bezopasnosti upravljenija dostupom v operacionnyh sistemah. M. : Gorjachaja linija – Telekom, 2019. 261 p. (in Russian)
7. Operacionnye sistemy Astra Linux. URL: <http://www.astralinux.ru>.
8. *Shelekhov V. I.* Klassifikacija programm, orientirovannaja na tehnologiju programirovanija // Programmaja inzhenerija. 2016. T. 7, № 12. P. 531–538. (in Russian)
9. *Shelekhov V. I.* Dokazatel'noe postroenie, verifikacija i sintez predikatnyh programm // Znaniya-Ontologii-Teorii (ZONT-2017). T. 2. Novosibirsk, 2017. P. 156–165. (in Russian)
10. *Kablukov I. V., Shelekhov V. I.* Realizacija optimizirujushchih transformacij v sisteme predikatnogo programirovanija // Sistemnaja informatika. № 11. Novosibirsk, 2017. P. 21–48. (in Russian)
11. *Shelekhov V. I.* Semantika jazyka predikatnogo programirovanija // ZONT-15. Novosibirsk, 2015. 13 p. (in Russian)
12. *Shelekhov V. I.* Proekt sistemy predikatnogo programirovanija. Novosibirsk, 2018. 24 p. (in Russian)

Научное издание

ПСИ'19

12-я МЕЖДУНАРОДНАЯ ЕРШОВСКАЯ КОНФЕРЕНЦИЯ
ПО ИНФОРМАТИКЕ

НАУКОЕМКОЕ ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ

Труды семинара

Материалы конференции публикуются в авторской редакции

Корректор *Я. О. Козлова*
Верстка *Я. О. Козловой*
Дизайн обложки *Т. Бульонковой*
Подготовка обложки к печати *Е. В. Неклюдовой*
Ответственный за выпуск *Т. В. Батура*

Подписано в печать 14.06.2019 г.
Формат 60×84 1/8. Уч.-изд. л. 19. Усл. печ. л. 17,7.
Тираж 80. Заказ № 148
Издательско-полиграфический центр НГУ.
630090, Новосибирск, ул. Пирогова, 2

Организаторы:



Институт систем информатики им. А. П. Ершова

N* Новосибирский
государственный
университет
***НАСТОЯЩАЯ НАУКА**

Спонсоры:

N* Novosibirsk
State
University
***THE REAL SCIENCE**



Microsoft Research

ARQA
TECHNOLOGIES

SibERS



STI · INTERNATIONAL



STI · INNSBRUCK

ISBN 978-5-4437-0909-3



9 785443 709093