ИСП Институт Системного Программирования Российской Академии наук

ISSN 2079-8156 (Print)
ISSN 2220-6426 (Online)

Труды Института Системного Программирования РАН

Proceedings of the Institute for System Programming of the RAS

Том 28, выпуск 3

Volume 28, issue 3

Москва 2016

ISSN 2079-8156 (Print), ISSN 2220-6426 (Online)

Труды Института системного программирования РАН

Proceedings of the Institute for System Programming of the RAS

Труды ИСП РАН – это издание с двойной анонимной системой рецензирования, публикующее научные статьи, относящиеся ко всем областях системного программирования, технологий программирования и вычислительной техники. Целью издания является формирование научно-информационной среды в этих областях путем публикации высококачественных статей в открытом доступе.

Издание предназначено для исследователей, студентов и аспирантов, а также практиков. Оно охватывает широкий спектр тем, включая, в частности, следующие:

- операционные системы;
- компиляторные технологии;
- базы данных и информационные системы;
- параллельные и распределенные системы;
- автоматизированная разработка программ;
- верификация, валидация и тестирование;
- статический и динамический анализ;
- защита и обеспечение безопасности ПО;
- компьютерные алгоритмы;
- искусственный интеллект.

Журнал издается по одному тому в год, шесть выпусков в каждом томе.

Поддерживается открытый доступ к содержанию издания, обеспечивая доступность результатов исследований для общественности и поддерживая глобальный обмен знаниями.

Труды ИСП РАН реферируются и/или индексируются в:

Proceedings of ISP RAS are a double-blind peer-reviewed journal publishing scientific articles in the areas of system programming, software engineering, and computer science. The journal's goal is to develop a respected network of knowledge in the mentioned above areas by publishing high quality articles on open access.

The journal is intended for researchers, students, and practitioners. It covers a wide variety of topics including (but not limited to):

- Operating Systems.
- Compiler Technology.
- Databases and Information Systems.
- Parallel and Distributed Systems.
- Software Engineering.
- Software Modeling and Design Tools.
- Verification, Validation, and Testing.
- Static and Dynamic Analysis.
- Software Safety and Security.
- Computer Algorithms.
- Artificial Intelligence.

The journal is published one volume per year, six issues in each volume.

Open access to the journal content allows to provide public access to the research results and to support global exchange of knowledge. **Proceedings of ISP RAS** is abstracted and/or indexed in:

УДК004.45

Адрес: 109004, г. Москва, ул. А. Солженицына, дом 25.
Телефон: +7(495) 912-44-25
E-mail: info-isp@ispras.ru
Сайт: http://www.ispras.ru/proceedings/

Address: 25, Alexander Solzhenitsyn st., Moscow, 109004, Russia.
Tel: +7(495) 912-44-25
E-mail: info-isp@ispras.ru
Web: http://www.ispras.ru/en/proceedings/

Содержание

Table of Contents

The Method of Converting an Expert Opinion to Z-number

E.A. Glukhoded <glukhodedkate@gmail.com>
S.I. Smetanin <sismetanin@gmail.com>
National Research University Higher School of Economics
20, Myasnitskaya st., Moscow, 101000 Russia

Abstract. The concept of Z-numbers introduced by Zade in 2011 is discussed topically nowadays due to it aptitude to deal with nonlinearities and uncertainties whose are common in real life. It was a large step of representing fuzzy logic, however that numbers created much larger problems of how to calculate them or aggregate multiple numbers of that type. Z-numbers have a significant potential in the describing of the uncertainty of the human knowledge because both the expert assessment and the Z-number consists of restraint and reliability of the measured value. In this paper, a method of converting an expert opinion to Z-number is proposed according to set of specific questions. In addition, the approach to Z-numbers aggregation is introduced. Finally, submitted methods are demonstrated on a real example. The topicality of the research is developing a new algorithm and software (based on that development) which could help people make decision in a messy uncertainty with many parameters and factors that are also defined imprecisely. In this work, we present the research that is aimed to find the most efficient methods to operate them (aggregate, add, divide). Firstly, it is important to identify all existing methods of aggregating Z-numbers. Secondly, it is needed to invent new methods of how work with them. The most interesting techniques should be detailed and summarized. There is also a program that is developed to model the real-word task of decision-making.

Keywords: Z-number, fuzzy measure, aggregation, expert opinion, fuzzy number.

DOI: 10.15514/ISPRAS-2016-28(3)-1

For citation: Glukhoded E.A., Smetanin S.I. The Method of Converting an Expert Opinion to Z-number. *Trudy ISP RAN/Proc. ISP RAS*, vol. 1, issue 2, 2016. pp. 7-20. DOI: 10.15514/ISPRAS-2016-28(3)-1

1. Introduction

Science and engineering tends to deal with different kinds of measures and evaluations, but in fact not all assessment of information can be represented as a clear number. It's common practice for human beings to describe the information in a linguistic terms which are more convenient in everyday life but unsuitable for a

standard mathematical representation. In this case information seems to be approximate because usually people assigns a different degree of the certainty depending on circumstances and the context of the data.

In order to resolve problem of the uncertainty degree representation, Zadeh proposed the concept of Z-numbers in 2011 [1]. According to this concept, Z-number describes an uncertain variable V as an ordered pair of fuzzy numbers (A, B), where the first number is a fuzzy set of the domain X of the variable V and the second one is a fuzzy set that specifies the level of a reliability of the first number as a unit interval.

Fuzzy logic methods are discussed topically last few decades due to its aptitude to deal with nonlinearities and uncertainties whose are common in real life. Despite the widespread application of many methods of fuzzy logic, it seems to be critical to talk about decision appliance without relation to the confidence and the reliability of analysed information especially in the field of fuzzy decision-making. For example, the decision, which was accepted based on low- reliability data, tends to be useless or even harmful on a practice usage. In this case, Z-numbers have a significant potential in describing uncertainty of the human knowledge because both the expert assessment and the Z-number consist

of restraint and reliability of the measured value. In this paper, the method of converting an expert opinion to Z-number is proposed and the new aggregation approach is introduced. At the end, suggested methods are demonstrated.

The paper is organized as follows. In section 2 required preliminaries are presented. In section 3 problem statement is described in details. In section 4 a method of converting expert assessment to Z-numbers is proposed. In addition, the approach to Z-numbers aggregation is introduced. In section 5 proposed methods is demonstrated on the real-life example. In the last section the key results of the article is mentioned and further ways of research is suggested.

2. Preliminaries

2.1. A linguistic variable

A linguistic variable is a variable whose values are linguistic expression such as sentences, phrases or words in an artificial or natural language. Processing data provided in linguistic variables requires the computing in terms of nonlinear approaches and leads to results, which are also not precise as the original data.

In general, the usage of linguistic variables is motivated by the feature that they provide more generalized information in contrast with numeric variables. For example, Speed is a linguistic variable which can be set to ‘very slow’, ‘slow’, ‘middle’, ‘quite high’, ‘high’, ‘very high’, etc. In natural language this linguistic variable may be represented as follows: 'The speed of the car is slow'. In this case, the characteristic of object under observations given in generalized form i.e. without

any specific numeric values, so expert has no need in specific measuring equipment for object estimation.

2.2. Fuzzy sets

Let X be a space of points (objects), with a generic element of X denoted by x. Thus, $X = \{x\}$. A fuzzy set [2] (class) A in X is characterized by a membership (characteristic) function $\mu(x)$ which associates with each point in X a real number in the interval [0, 1], with the value of $\mu(x)$ at x representing the 'grade of membership' of x in A. Thus, the nearer the value of $\mu(x)$ to unity, the higher the grade of membership of x in A. When A is set in the ordinary sense of term, its membership function can take on only two values 0 and 1, with $\mu(x)$ reduces to the familiar characteristic function of set A. [17, 18]

In the decision-making tasks [3] each expert gives his own opinion and then it is needed to represent given information in a form that can be processed by a machine. We can use fuzzy numbers for representing the information. Fuzzy numbers can be defined as follows.

2.3. A fuzzy number

A fuzzy number [14, 15] is a convex and normalized fuzzy set with membership function, which is defined in R and piecewise continuous. In other words, a fuzzy number represents an interval of crisp numbers with fuzzy boundary.

Classical example of fuzzy number is triangular fuzzy number. It is represented by a set of two boundary points a1 a3 and a peak point a2, i.e. [a1, a2, a3], as shown in Fig. 1.

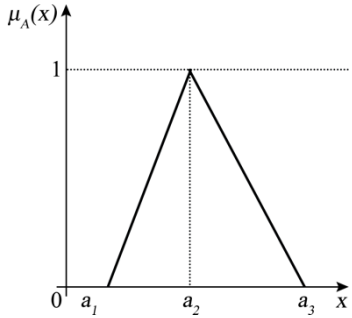


Fig. 1. A triangular fuzzy number

A concept of Z-numbers [8] was proposed in 2010, which was associated with a factor of information reliability for decision-making tasks, a description of the various aspects in the world, an expression of ideas or assessments.

2.4. A Z-number

A Z-number [2] is an ordered pair of fuzzy numbers denoted as Z = (A, R). The first component A, a restriction on the values, is a real-valued uncertain variable X. The second component R is a measure of reliability for the first component.

2.5. A Z+-number

Z+ number [2] is a combination of fuzzy number, A, and a random number, R, written as an ordered paid Z+=(A, R). A plays the same role as in a Z-number, R is a probability distribution of random number.

3. *Problem statement*

Communications between people is often reduced to the expression of their opinions, reviews or evaluations. Some examples of everyday expert assessments are follows:

1. «What is the weather forecast for tomorrow? I really don't know, but I am quite sure that it will be warm». In this example, during the conversation an expert provides an assumption of the prospective weather in linguistic terms and mention a degree of confidence in it. Therefore, X = Weather forecast for tomorrow, and Z=<warm, quite sure>.
2. «It takes me about 2 weeks to finish course work». Therefore, X = Time to finish course work, and Z=<about 2 weeks, usually>.

Generally, the formalization of the statements in a natural language is a complex and unobvious task [9, 10]. For example, the degree of confidence or reliability of the expert estimation can be provided in two ways, namely in explicit or implicit form [11]. The explicit form is represented in example (i) in linguistic term ‘quite sure’ and the implicit form is contained in the context in example (ii).

Now, it is needed to formulate the problem in a general way. Consider, set of objects (Ω) that needs to be assessed by experts or people who have specific knowledge in the field that relates to Ω. Also, there is set of criteria (ⱨ), that should be taken into account. In this paper it is not supposed to describe the methods for assessments aggregation. That is why it does not need to involve set of experts in problem statement formulation.

Each expert expresses his own opinion by filling the form with question that are written in a developed form. Questions are formulated using conventional language, such as “how can you assess the level of safety of the given system?” or “Are you sure that the level is high?” or, probably, the most complicated: “How can you assess the distribution of that parameter? Is it Gaussian?” Strict form will be illustrated in the chapter V.

When expert filled his questionnaire with answers it is needed to represent given information in a form that can be processed by machine. There could be several levels of abstraction for representing the information (table 1).

Table 1. Levels of abstraction [5]

Level	Appearance (in Time New Roman or Times)	
	Numbers	
3	Z-numbers or Z+-numbers	
2	Fuzzy numbers	Random numbers
1	Intervals	
0	Crisp numbers (Integer or Float)	

In this paper the highest level supposed to be considered – Z-numbers. Actually, there will be even Z+-numbers in the chapter IV and V.

The main goal of all paper is to describe the recipe of how human-readable information could be represented as Z- numbers. It should be also mentioned that expert's opinions per each criterion should be somehow accumulated (or aggregated) in a common Z-number that will describe whole relation of a criterion to the considered object.

4. The proposed method of converting an expert opinion to z-number

First of all, it is needed to describe strict form that expert needs to fill in order to provide knowledge. Form should contain N sections of $\mathfrak{m}$. Each section should contains several questions that should describe relation of the criterion $C \in \mathfrak{m}$ to the object $O \in \Omega$. Common questions are follows:

1. How does O meet C? Specify the level.
2. Do you have an experience with O? How wide was it?
3. Have you taken into account C when using O previously? (If have any experience)
4. Do you follow the latest information about O? When was the latest update?
5. (Only if experts said 'yes' on second question) Which distribution, you think, respects to people perception of C when talking about O?

First and fourth questions are related to the main part of Z- number, other questions are somehow related to the measure of reliability for the first component. Let us begin with the main part of Z-number. First question is direct. It is supposed that expert assesses the level of affection C on O in the following terms: very high (8, 9, 10), high (7, 8, 9), medium (5, 6, 7), low (3, 4, 5), very low (1, 2, 3), does not meet at all (0, 1, 2). However, it would be incorrect to assign fuzzy numbers to each option before he answered fourth question. This question would show how precise expert could be answering previous one. For example, if he says that he does not follow information for a long time, it should be a clear sign that bounds of each

triangle (or trapezoidal) fuzzy-numbers should be widened due to some incompetency in the given field. According to that, there could be different fuzzy numbers for medium: (4, 5, 6) or (2, 5, 8).

The formation of second part should be following. Possible answers for question two are: wide experience, have experience, some experience, little experience, no experience. All these statements could be represented as fuzzy number: (0.8, 0.9, 1.0), (0.65, 0.75, 0.85), (0.4, 0.5, 0.6), (0.1, 0.3, 0.4), (0, 0.1, 0.2). Answers for the question three could be: yes, a lot (0.8, 0.9, 1.0); yes, sometimes (0.5, 0.6, 0.7); yes, a little (0.2, 0.3, 0.4); no (0.0, 0.1, 0.2). Fifth question is auxiliary and will not be converted into fuzzy number. It relates to Z+- numbers. Possible answers: Gaussian, Inverse Gaussian, Binomial, Gamma. It is not prohibited for an expert to skip this question. However, if expert answers the question the specific 'confidentiality' fuzzy number has a value of (0.6, 0.7, 0.8) – 'high', otherwise – (0.0, 0.1, 0.2) – 'low'.

Then several rules should be formulated to construct possibility measure of Z-number: IF experience='wide' AND take-in-account='a lot' AND confidentiality='high' THEN measure='very high'.

Table 2 is completed according to the given format.

Table 2. Rules for probability measure

#	CONDITION			RESULT
	Exper.	**Take-in**	**Conf.**	**Measure**
1	Wide	A lot	High	Very high
2	Wide	A lot	Low	High
3	Wide	Sometimes	High	Very high
4	Wide	Sometimes	Low	High
5	Wide	A little	High	High
6	Wide	A little	Low	Medium
7	Wide	No	High	High
8	Wide	No	Low	Medium
9	Have	A lot	High	Very high
10	Have	A lot	Low	High
11	Have	Sometimes	High	High
12	Have	Sometimes	Low	Medium
13	Have	A little	High	High
14	Have	A little	Low	Medium
15	Have	No	High	Medium
16	Have	No	Low	Low
17	Some	A lot	High	High
18	Some	A lot	Low	Medium
19	Some	Sometimes	High	Medium
20	Some	Sometimes	Low	Medium
21	Some	A little	High	Medium
22	Some	A little	Low	Low

23	Some	No	High	Medium
24	Some	No	Low	Low
25	Little	A lot	High	Medium
26	Little	A lot	Low	Low
27	Little	Sometimes	High	Medium
28	Little	Sometimes	Low	Low
29	Little	A little	High	Low
30	Little	A little	Low	Low
31	Little	No	High	Low
32	Little	No	Low	Very low
33	No	-	-	Very low

Then, after the result is given, it should be converted to fuzzy number:

- Very high – (0.8, 0.9, 1.0)
- High – (0.6, 0.7, 0.8)
- Medium – (0.4, 0.5, 0.6)
- Low – (0.2, 0.3, 0.4)
- Very low – (0.0, 0.1, 0.2)
- The B-part is given.

The resulting Z-number should be constructed from both parts Z (A, B).

Then, it is time to aggregate different Z-numbers into one Z-number which can describe whole relation of O to the subject according to C based on expert's opinion. There are, a least, three methods of aggregation:

1. Converting Z-number into simple fuzzy number and aggregate them using simple methods [4, 12, 13].
2. Aggregating A and B-part separately [14, 18].
3. Converting Z-number into Z+-number, aggregating Z+-numbers and then convert given Z+-number into Z-number [6, 7].

First approach is the simplest one, but lose some information from an expert. Third approach is one where loss of information is minimized, but it is complicated and it would be difficult to provide all calculations in this paper (requires some non-linear optimization algorithms at some stages). That is why second approach is chosen.

At first stage, it is needed to aggregate A-parts [17, 18]:

$$\widetilde{w} = (a_r, b_r, c_r)$$

Here $a_r = \min(a_i)\,, b_r = \frac{1}{n}\sum_{i=1}^{n} b_i,\ \ c_r = max(c_i)$, N- total number of Z-numbers. (1)

Aggregation of B-parts are more complicated. First of all, it is needed to multiply one number on another.

$$\mu\ (z) = \begin{cases} \frac{-a_1b_2+a_2b_1-2a_1a_2)+\sqrt{(a_1b_2-a_2b_1)^2+4(b_1-c_1)(b_2-a_2)z}}{2(b_1-a_1)(b_2-a_2)}, a_1a_2 \le z \le b_1b_2 \\ \frac{-c_1b_2+c_2b_1-2c_1c_2)+\sqrt{(c_1b_2-c_2b_1)^2+4(b_1-c_1)(b_2-c_2)z}}{2(b_1-c_1)(b_2-c_2)}, b_1b_2 \le z \le c_1c_2 \\ 0, \quad otherwise \end{cases} \quad (2)$$

Second stage of aggregation assumes that square root calculations should be applied to the given fuzzy number. Somehow, these transformations could be compared with a calculation of geometric mean, when talking about crisp numbers.

$$\mu_{\sqrt{x}}(x) = \begin{cases} \frac{x^2-a}{b-a}, \ \sqrt{a} \le x \le \sqrt{b} \\ \frac{c-x^2}{c-b}, \ x\sqrt{b} \le x \le \sqrt{c} \\ 0, otherwise \end{cases} \quad (3)$$

The resulting fuzzy number is aggregated B-part.

5. *A numerical example*

Consider following example. It is needed to decide whether it is important to have a new developer in a company or not. To decide more accurately, management of software company introduces a research. Several 'experts' are chosen from different departments. Now, focus on a specialist of marketing. His assessment should be transformed into Z-number for further calculations. It is not necessary for that time to come to a complete conclusion, just focus on how expert's evaluation lead to obtaining Z-number.

5.1 Filling a form

There is a strict form with several questions and two criteria.

Questions for the 1st criterion – level of business in the department:

1. How do you think, does the department of software development are filled with work? (very much, much, probably, not so much, no, not at all)
2. How often do you communicate with developers during business tasks? Select from: very often, often, quite often, rarely, not communicate.
3. Did you notice anytime, that deadline was broken due to lack of programmers? Do you pay attention on it? Select from: Notice very often, sometimes notice, I've noticed once, Not at all.
4. Did you follow the news of our developers' team? Do you know most of them? Select from: Yes, communicating each day; Yes, communicating several times a week; Yes, but communicating rarely; No, I'm not.
5. Which distribution, you think, respects to people perception of lack of human resources when talking about new coming developer? (Only if you know)

Questions for the 2nd criterion – company's resources:

1. How do you think, does the company have enough resource to hire new employee(s) in a software department? Select from: More than enough; enough; quite enough; probably, not enough; not enough at all.
2. Have you ever been interested in our company's revenue, stock prices etc.? Select from: Yes, I follow all news; yes, sometimes; yes, but rarely; probably, once; no.
3. Have you ever thought about how newcomers can change our budget or resources distribution? How you thought about it when you came? Select from: Yes, thought a lot; yes, sometimes; yes, when I came; No.
4. Do you follow latest news about our state, about our resources distribution on different projects? Select from: Yes, always; yes, sometimes; yes, but rarely; no, I'm not.
5. Which statistics distribution, you think, respects to people perception of our resources when talking about new employees? (Only if you know)

The marketing expert gives answers:

Criterion 1:

1. Not so much
2. Rarely
3. Sometimes notice
4. No, I'm not
5. –

Criterion 2:

1. Quite enough
2. Yes, I follow all news
3. Yes, sometimes
4. Yes, sometimes
5. –

5.2 Constructing Z-numbers

According to calculations in section 4 and table 2 it is possible to calculate two Z-numbers from his answers.

Z-number from the 1st criterion: {(1, 4, 7), (0.2, 0.3, 0.4)}

Z-number from the 2nd criterion: {(5, 6, 7), (0.6, 0.7, 0.8)}

Second part of Z-numbers is given by applying corresponding rules from Table II.

5.3 Aggregating Z-numbers

A-part of Z-numbers is aggregated simply by applying formula (1) from section 4. The resulting A-part: (1, 5, 7).

B-part of Z-numbers is calculated using formula (2) and (3) from section 4. For simplicity, all calculations would not be provided, only bounds for each of fuzzy number at every step. After multiplication following fuzzy number is obtained:

$$B_m = (0.12, 0.21, 0.32)$$

Then, after applying square root transformation:

$$B_r \approx (0.35, 0.46, 0.57)$$

$$B_r - resulting\ B - part\ of\ Z - number$$

It should be noticed, that bounds are not lines in this case, they are quadratic functions.

That is why, the resulting Z-number looks as follows:

$$\boldsymbol{Z_r} = \{(\mathbf{1}, \mathbf{5}, \mathbf{7}), (\mathbf{0.35}, \mathbf{0.46}, \mathbf{0.57})\}$$

This Z-number expresses overall relation to the problem of marketing expert. It could be translated to the normal language such as: “he doubts that software department needs new employee and probably, they do not need, but he is not sure enough”

6. Conclusion

Consider following example. It is needed to decide whether it is important to have a

As a result of this research, a method of converting an expert opinion to Z-number was proposed. The key problems of Z-number extraction from natural language statements were discussed and an example illustrating the supposed algorithm were provided. In addition, the new method of Z-numbers aggregation was proposed and demonstrated on the real example.

The further research will be aimed on improving methods of aggregation in order to obtain a more accurate and reasonable result at the output. The high-quality processing of experts assessments allows the use of this approach in the real world in order to solve complex problems not only in the business sector, but also in the everyday life. The next step is developing an approach to perform arithmetic operations with observed Z-numbers. Only after successful finishing of these steps, a complete system for Z-numbers processing could be built.

References

[1]. Adrian K. Rantilla, David V. Budescu. Aggregation of Expert Opinions. In Systems Sciences, HICSS-32. Proceedings of the 32nd Annual Hawaii International Conference, 1999.

[2]. Aliev, R. A., Alizadeh A. V. and Huseynov O. H. The arithmetic of discrete Z-numbers, Inform. Sciences, 290(1), 2015, pp.134-155.

[3]. Aliev, R.A., Alizadeh, A.V, Huseynov, O. H, Aliyev, R.R. The arithmetic of Z-numbers. Theory and Applications. World Scientific, 2015.

[4]. Aliev, R.A., Zeinalova, L. M. Decision-making under Z-information. In Human-centric decision-making models for social sciences, Springer-Verlag,, 2013, pp. 233-252.

[5]. Bai, Y., Wang, D. Fundamentals of Fuzzy Logic Control – Fuzzy Sets, Fuzzy Rules and Defuzzifications. In Advanced Fuzzy Logic Technologies in Industrial Applications, Springer, 2006 pp. 17-36.
[6]. Detyniecki, M. Fundamentals on Aggregation Operators. Berkeley initiative in Soft Computing, Computer Science Division, University of California, Berkeley, United Sates of America, 2001.
[7]. Eiichiro Takahagi. Usage: Choauet integral. Fuzzy Integral Calculation Site (online publication). Available at: http://www.isc.senshu-u.ac.jp/~thc0456/Efuzzyweb/mant1/mant1.html, accessed 03.04.2016.
[8]. Farina, M., and Amato, P. A fuzzy definition of "optimality" for many criteria optimization problems. *IEEE T. Syst. Man Cy. A: Systems and Humans*, 34(3), 2004, pp. 315-326.
[9]. Gilboa I., Schmeidler D. Additive Representations of Non-Additive Measures and the Choquet Integral. Kellogg School of Management, Northwestern University, Center for Mathematical Studies in Economics and Management Science, Discussion Papers (online publication). Available at: https://www.kellogg.northwestern.edu/research/math/papers/985.pdf, accessed 17.05.2016.
[10]. Gilboa, I. Theory of Decision under Uncertainty. Cambridge University Press, Cambridge, 2009.
[11]. Kang B., Wei D., Li Y., Deng Y. Decision Making Using Z-numbers under Uncertain Environment. Journal of Information & Computational Science, №8(7), 2012, pp. 2807-2814.
[12]. Kang, B., Wei, D., Li, Y., Deng, Y. A method of converting Z-number to classical fuzzy number. Journal of Information & Computational Science, №9(3), 2012 pp. 703-709.
[13]. Lala M. Zeinalova, Choquet aggregation based decision making under Z-information. ICTACT Journal on Soft Computing, 4(4), 2014, pp. 819-824.
[14]. Studies in Fuzziness and Soft Computing, vol. 97. Tomasa Calvo, Gaspar Mayor, Radko Mesiar (eds). Aggregation Operators. New Trends and Aplications. Physica-Verlag Heidelberg, 2002, 352 p.
[15]. Zadeh, L.A. A Note on Z-numbers. Information Sciences, №181, pp. 2923-2932.
[16]. Zadeh, L.A. Fuzzy sets. Information and Control, vol. 8(3), pp. 338–353, 1965.
[17]. Gulnara Yakh'yaeva. [Fundamentals of the theory of fuzzy sets. Lecture 4. Indicator for fuzziness of fuzzy sets. Fuzzy measures and integrals] (online publication). Available at: http://www.intuit.ru/studies/courses/87/87/lecture/20505?page=3, accessed 15.05.2016 (in Russian).
[18]. Sakulin S.A., Alfimtsev A.N. On the issue of the practical application of fuzzy measures and Choquet integral. Herald of the Bauman Moscow State Technical University. Series Instrument Engineering, spec. issue 4: Computer Systems and Technologies, 2012, pp. 55-63 (in Russian).

IE.A.Glukhoded, S.I.Smetanin. The Method of Converting an Expert Opinion to Z-number. *Trudy ISP RAN /Proc. ISP RAS*, 2016, vol. 28, issue 3, pp. 7-20.

Метод представления мнений экспертов в виде Z-чисел

Глуходед Е.А <glukhodedkate@gmail.com>
Сметанин С.И. <sismetanin@gmail.com>
Научно-исследовательский университет Высшая школа экономики
101000, Россия, г. Москва, ул. Мясницкая, д. 20

Аннотация. Нечеткие числа используются в задачах моделирования для учета лингвистической неопределенности. Большинство информации, обрабатываемой в различных сферах деятельности, основано на оценках, которые не всегда могут быть выражены точным числом. Как правило, используются привычные для человека слова или выражения естественного языка. Надежность и достоверность данных, которые мы получаем для решения тех или иных задач играет важную роль. Мы часто работаем с неполной информацией, основанной на опыте и оценках различных экспертов. Поэтому возможность формализации данных такого типа и выполнения с ними различных вычислений помогает более точно решать задачи по планированию, принятию решений, оценке рисков и других аспектов практической деятельности. В 2010 году профессор Лотфи Заде предложил концепцию *Z*-чисел, которая связана с фактором надежности используемой информации (при принятии решений, при описании различных аспектов окружающего мира, при выражении идей или суждений людьми). *Z*-число описывает значение некоторой неопределённой переменной *X* и представляет собой упорядоченную пару из двух нечетких чисел *Z= (A, B).* Первое из которых *(A)* выражает ограничение на возможные (вероятные) значения рассматриваемой в конкретном приложении переменной *X*. Второе число *(B)* есть мера (оценка) уверенности в том, что *A* именно такова, как она представлена. Числа *A* и *B* часто описываются фразами естественного языка, например, *Z* = (Java, максимально уверен). В данном случае переменная *X* = «язык программирования для решения определенной задачи», следовательно, утверждение «*X* является Java» оценивается как надежное (*B* = “максимально уверен”). Данная концепция имеет большой потенциал стать важным инструментом в решении различного рода задач, связанных с неполнотой и неточностью описания используемой информации. Вторым основным аспектом данной работы является агрегация информации, а именно *Z*-чисел. Агрегация (англ. aggregation) – основной этап для решения задач по принятию решений. Как правило, чтобы прийти к определенному выводу, необходимо проанализировать несколько источников и объединить полученную информацию. В настоящее время большинство задач по принятию решений имеют множество факторов, определенных нечетко и поэтому решаются интуитивно. Агрегация данных, представленных в нечетком виде, а именно Z-числами, может оказать поддержку в решении задач такого рода. Данная работа представляет собой исследование, связанное с разработкой и изучением эффективных методов обработки *Z*-чисел и выполнения операций над ними.

Ключевые слова: Z-число, нечеткая мера, агрегация, мнения экспертов, нечеткое число.

DOI: 10.15514/ISPRAS-2016-28(3)-1

Для цитирования: Глуходед Е.А., Сметанин С.И. Метод представления мнений экспертов в виде Z-чисел. Труды ИСП РАН, том 28, вып. 3, 2016 г., стр. 7-20 (на английском). DOI: 10.15514/ISPRAS-2016-28(3)-1

Список литературы

[1]. Adrian K. Rantilla, David V. Budescu. Aggregation of Expert Opinions. n Systems Sciences, HICSS-32. Proceedings of the 32nd Annual Hawaii International Conference, 1999.

[2]. Aliev, R. A., Alizadeh A. V. and Huseynov O. H. The arithmetic of discrete Z-numbers, Inform. Sciences, 290(1), 2015, pp.134-155.

[3]. Aliev, R.A., Alizadeh, A.V, Huseynov, O. H, Aliyev, R.R. The arithmetic of Z-numbers. Theory and Applications. World Scientific, 2015.

[4]. Aliev, R.A., Zeinalova, L. M. Decision-making under Z-information. In Human-centric decision-making models for social sciences, Springer-Verlag, 2013, pp. 233-252.

[5]. Bai, Y., Wang, D. Fundamentals of Fuzzy Logic Control – Fuzzy Sets, Fuzzy Rules and Defuzzifications. In Advanced Fuzzy Logic Technologies in Industrial Applications, Springer, 2006, pp. 17-36.

[6]. Detyniecki, M. Fundamentals on Aggregation Operators. Berkeley initiative in Soft Computing, Computer Science Division, University of California, Berkeley, United Sates of America, 2001.

[7]. Eiichiro Takahagi. Usage: Choauet integral. Fuzzy Integral Calculation Site (online). Доступно по ссылке: http://www.isc.senshu-u.ac.jp/~thc0456/Efuzzyweb/mant1/mant1.html, 03 апреля 2016.

[8]. Farina, M., and Amato, P. A fuzzy definition of "optimality" for many criteria optimization problems. *IEEE T. Syst. Man Cy. A: Systems and Humans*, 34(3), 2004, pp. 315-326.

[9]. Gilboa I., Schmeidler D. Additive Representations of Non-Additive Measures and the Choquet Integral. Kellogg School of Management, Northwestern University, Center for Mathematical Studies in Economics and Management Science, Discussion Papers (online), №985, 1992. Доступно по ссылке: https://www.kellogg.northwestern.edu/research/math/papers/985.pdf, 17 мая 2016.

[10]. Gilboa, I. Theory of Decision under Uncertainty. Cambridge University Press, Cambridge, 2009.

[11]. Kang B., Wei D., Li Y., Deng Y. Decision Making Using Z-numbers under Uncertain Environment. Journal of Information & Computational Science, № 8(7), , 2012, pp. 2807-2814.

[12]. Kang, B., Wei, D., Li, Y., Deng, Y. A method of converting Z-number to classical fuzzy number. Journal of Information & Computational Science, № 9(3), 2012, pp. 703-709.

[13]. Lala M. Zeinalova, Choquet aggregation based decision making under Z-information. ICTACT Journal on Soft Computing, 4(4), pp. 819-824, 2014.

[14]. Studies in Fuzziness and Soft Computing, vol. 97. Tomasa Calvo, Gaspar Mayor, Radko Mesiar (eds). Aggregation Operators. New Trends and Aplications. Physica-Verlag Heidelberg, 2002, 352 p.

[15]. Zadeh, L.A. A Note on Z-numbers. Information Sciences, №181, pp. 2923-2932.

[16]. Zadeh, L.A. Fuzzy sets. Information and Control, vol. 8(3), pp. 338–353, 1965.

[17]. Гульнара Яхъяева. Основы теории нечетких множеств. Лекция 4. Показатель размытости нечетких множеств. Нечеткие меры и интегралы (online). Доступно по ссылке: http://www.intuit.ru/studies/courses/87/87/lecture/20505?page=3, 15 мая 2016.

[18]. Сакулин С.А., Алфимцев, А.Н. К вопросу о практическом применении нечетких мер и интеграла Шоке. Вестник МГТУ им. Н. Э. Баумана. Сер. Приборостроение, спец. вып. 4: Компьютерные системы и технологии, стр. 55-63, 2012.

Deep Web Users Deanonimization System

S.M. Avdoshin <savdoshin@hse.ru>
A.V. Lazarenko <avlazarenko @edu.hse.ru>
School of Software Engineering,
National Research University Higher School of Economics,
20, Myasnitskaya st., Moscow, 101000 Russia

Abstract. Privacy enhancing technologies (PETs) are ubiquitous nowadays. They are beneficial for a wide range of users: for businesses, journalists, bloggers, etc. However, PETs are not always used for legal activity. There a lot of anonymous networks and technologies which grants anonymous access to digital resources. The most popular anonymous networks nowadays is Tor. Tor is a valuable tool for hackers, drug and gun dealers. The present paper is focused on Tor users' deanonimization using out-of-the box technologies and a basic machine learning algorithm. The aim of the work is to show that it is possible to deanonimize a small fraction of users without having a lot of resources and state-of-the-art machine learning techniques. The first stage of the research was the investigation of contemporary anonymous networks. The second stage was the investigation of deanonimization techniques: traffic analysis, timing attacks, attacks with autonomous systems. For our system, we used website fingerprinting attack, because it requires the smallest number of resources needed for successful implementation of the attack. Finally, there was an experiment held with 5 persons in one room with one corrupted entry Tor relay. We achieved a quite good accuracy (70%) for classifying the webpage, which the user visits, using the set of resources provided by global cybersecurity company. The deanonimization is a very important task from the point of view of national security.

Keywords: Tor; deanonimization; website fingerprinting; traffic analysis; anonymous network; deep web.

DOI: 10.15514/ISPRAS-2016-28(3)-2

For citation: Avdoshin S.M., Lazarenko A.V. Deep Web Users Deanonimization System. *Trudy ISP RAN/Proc. ISP RAS*, vol. 28, issue 3, 2016, pp. 21-34. DOI: 10.15514/ISPRAS-2016-28(3)-2

1. Introduction

Internet privacy is considered as an integral part of freedom of speech. A lot of people are concerned about their anonymity in public and therefore, there is a growing need for privacy enhancing technologies.

The Deep Web is a layer of the Internet, which can not be accessed by traditional search engines, so the content in this layer is not indexed. The typical website in the deep web is static, with potentially no links to outer resources. For that reason, it is very hard to measure the real size of the deep web.

In the modern world, there are a lot of networks and technologies, which grant access to deep web resources, for example, Tor, I2P, Freenet, etc. Each of these instruments hides users' traffic from adversaries, thus making the deanonimization a hard thing to do. A detailed overview of such technologies can be accessed in paper [1].

Nowadays, the largest and most widely used system is Tor [2]. Our research focuses on Tor users' deanonimization, because of its popularity and prevalence.

2. Tor background

Tor is the largest active anonymous network in the world. There are more than two million users per month, and the number of relays is close to 7000 [3]. Tor is a distributed overlay network consisting of volunteer servers. Every user in the world can provide Tor with computational resources needed for traffic retranslation over the network.

Despite being a great privacy enhancing technology for law-abiding citizens, Tor is an essential tool in criminal society. Terrorists, drug and arm dealers in line with other offenders use Tor for their criminal activities. Thus, the solution of the deanonimization problem is very important for government special services [4]. For example, Russian Ministry of Internal Affairs (MIA) has recently announced a bidding for Tor deanonimization system [5].

The next key component of Tor is Hidden Services (HS). Tor HS provides users with anonymous servers to host their websites or any other applications. HS are accessed via special pseudo-domains «.onion», where Deep Web is located. From the user`s point of view, accessing a particular hidden service is as easy as visiting a normal website.

In order to establish a connection with Tor network, the user must have pre-installed software (Tor client). The easiest way is to install TorBrowser, which is a customized version of Mozilla Firefox with built-in Tor software. To initiate the connection, a Tor client obtains a list of Tor nodes from a directory server. Then, the client builds a circuit of encrypted connections through relays in the network. The circuit is extended hop by hop, and each relay on the path knows only which relay gives data and which relay it is giving data to. There is no particular relay in the circuit (see Fig. 1), which knows the complete users path through the network.

A Layered encryption is used along the path. The most interesting relays for a potential attacker are entry and exit relays. Every piece of information in the network is transferred in Tor cells that have equal size. An Entry relay (also called the guard) knows the IP address of the user, and Exit relay knows the destination

resource. Traffic interception in the middle would not give any advantage to the attacker because everything is encrypted and secure.

3. Deanonimization techniques

There is a wide range of deanonimization methods (attacks). Some of them are passive: an adversary only observes traffic, without any trials to modify it somehow. Contrariwise, some of them are active: an attacker modifies traffic causing delays, insert patterns, etc. Earlier, we proposed classification of attacks, where the main principle is the amount of resources needed by an attacker to perform the deanonimization (see table 1).

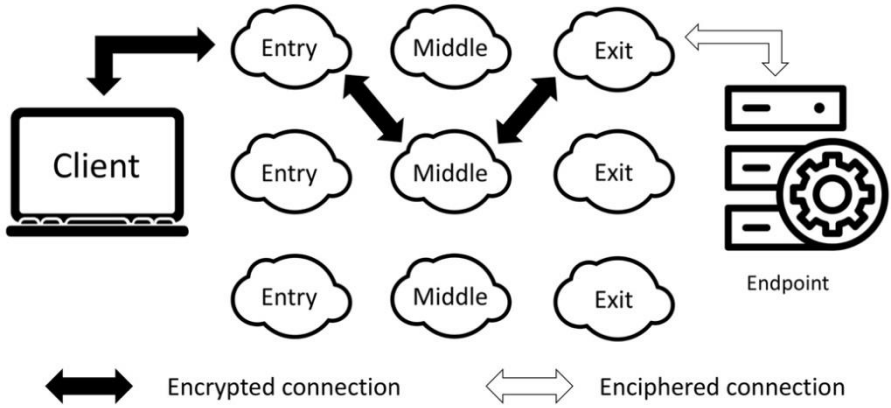


Fig. 1. Tor circuit example

Table 1. Attacks classification

#	Resources	Attacks
1	Corrupted entry guard	• Website fingerprinting attack
2	Corrupted entry and exit nodes	• Traffic analysis • Timing attack • Circuit fingerprinting attack • Tagging attack
3	Corrupted exit node	• Sniffing of intercepted traffic
4	Corrupted entry and exit nodes, external server	• Browser based timing attack with JavaScript injection • Browser based traffic analysis attack with JavaScript injection
5	Autonomous system	• BGP hijacking • BGP interception • RAPTOR attack
6	Big number of various corrupted nodes	• Packet spinning attack • CellFlood DoS attack • Other DoS and DDoS attacks

More information about attacks mentioned in Table 1 can be found in paper [6]. We are focused on the resource-effective attack (WF), which only requires an attacker to control an entry relay of the user. The relay, which is fully controlled by an attacker is called a *corrupted* relay.

4. Website fingerprinting attack

4.1 Website Fingerprinting Attack Overview

A website fingerprinting attack (WF) is an attack designed for a local passive eavesdropper to determine the client's endpoint using features from packet sequences. Generally speaking, WF breaks privacy, which is achieved by the proxy, VPN or Tor. This is an application of various machine learning techniques in the field of privacy.

The first appearance of the WF was discussed in paper [7]. This attack has been widely discussed in the researchers` community because it has proven its effectiveness against various privacy enhancing technologies, such as Tor, SSL and VPN.

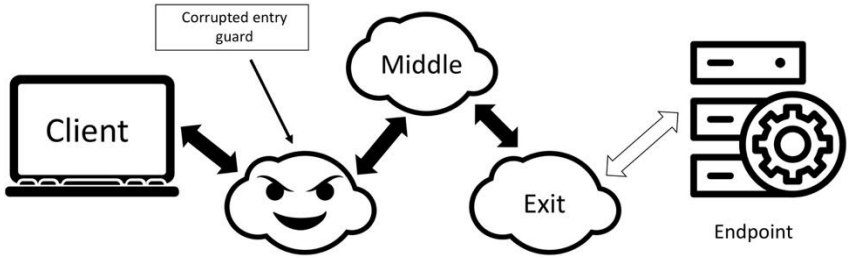


Fig. 2. Configuration of Tor circuit suitable for the WF attack

To perform a WF, an eavesdropper has to simulate users` behavior in the network, using the same conditions as the victim. In case of Tor, an attacker must have a corrupted entry relay (see Fig.2) that will be used for collecting data. The Attacker visits each site from the list and stores all packet sequences related to the request. Afterwards, he uses the traffic for training a classifier in a supervised way. The machine learning problem could be stated as a binary classification problem or multilabel classification problem. In the first case, classifier is trained to answer the question: «If the user visits a site from our list?». The second option is about guessing a particular website that the user visits.

4.2 The Oracle Problem

Since WF works with packet sequences, determining sequences related to the webpage is quite a difficult task. This issue is known as the Oracle problem. Researchers make two major assumptions, which simplify WF a lot: 1) an attacker has such an oracle at his disposal, 2) the victim loads pages one-by-one in a single

tab. The Oracle helps to find precise subsequence of packets from overall captured traffic. Any excess packet sequence sent to classifier can significantly reduce its' accuracy. That is why, splitting the whole sequence is crucially important. Another reason is the user`s web-browsing behavior. The majority of people uses multi-tab browsing instead of loading a page in a single tab, working with it and loading another one. This behavior makes WF difficult in real life.

An Oracle problem for packet sequences has not been solved yet, but Wang proposed a solution for Tor, which can work with a single tab [8]. He considered three-step process of determining correct split in case of single tab browsing between two pages. Wang used Tor cells instead of packets. The first step is making a time based split. The Attacker splits sequences if the time gap between two adjacent cells is greater than some constant, then the sequence is splitted there into two subsequences. If the time gap is too small, classification-based splitting is typically used. Wang used machine learning techniques that decide where to split and whether to split or not. After splitting, the result is ready for further classification. This method achieves quite good accuracy. However, the proposed solution doesn't work with multi-tab browsing and raw packet sequences, narrowing the range of real implementations. Study [9] proposed a time-based way to split traffic traces when the user utilizes 2 open tabs. They classify the first page with 75.9% and second with 40.5% of accuracy.

4.3 Real World Scenario

Overall, the applicability of WF in the real world scenario is still questionable. Users may visit hundreds of thousands of webpages every day. So, can the attacker successfully apply WF in reality? Panchenko et al. [10] checked the attack with a really huge dataset, and their approach outperformed the previous state of the art attack proposed by Wang. To conclude, WF attacks are still a serious threat to anonymous communication systems.

The aim of the current work is to show that an attacker can build a deanonimization system, applying learning libraries for most popular programming languages, which will be able to deanonimize a group of users trying to access the deep web content.

5. Deanonimization system scheme

For the sake of simplicity, we will use as much preconfigured software as possible. In order to deal with deanonimization problem, our system must have two modules. The first module is used for mining Tor data, which will be used for collecting traffic traces. The second is aimed at applying machine learning techniques.

5.1 Data Mining

The data mining module is using various software, which can be easily installed on Mac OS or any Linux distributive. Since the packet traces can be collected on the relay side, or on the client side (the difference is only in the source/destination pair),

we can use data mining module on local machine or on the remote server. We will use local machine for data mining (see Fig. 3). Simple data transformation can be applied for packet traces, to look exactly like those collected on the relay.

The following software must be installed on the machine:

- Tor – free software for enabling anonymous communication,
- Torsocks – free software that allows using any kind of application via the Tor network,
- Wget – a program, which retrieves content from the web server and supports downloading via http, https, ftp,
- Tshark – a free and open packet analyzer; it is used for network troubleshooting, analysis, etc.,
- Mozilla Firefox or Tor Browser – an open web-browser (in case of Mozilla Firefox, it is needed to configure it for using Tor manually).

Nevertheless, any program can be replaced by the specific library. The simplest solution is to use the proposed software. We must have full control over Tor circuits construal to use our own relay. For this purpose, we will use *Stem* Python library, which is freely accessible on the web. *Stem* is a Python controller library for Tor.

We use *Stem* to create Tor circuits through our corrupted entry guard. Without this action, the accuracy of the classifier might become worse, because of different Tor versions on the relays and other reasons. Another option is to modify Tor configuration for using specified entry guards. It is very important to use the same entry guard, which will be used in production.

Tshark is used as the main packet capturing tool. We also use Tshark for extracting TLS records from data. Tshark can be substituted with any library, which supports capturing of TCP packets.

After that, the attacker has to automate the data gathering process. There are two ways to do it, namely, using wget via torsocks, or Mozilla Firefox. In case of wget, an attacker just launches page downloading from the command line, but the use of Mozilla Firefox requires more work. The automation of Mozilla can be done in two ways. The first option is to launch it from the command line and wait while the page is uploading; another one is to use Selenium Webdriver to automate the process.

5.2 Feature Extraction

We can extract features of the traffic at three different levels (see Fig.4) – they are Tor cells, TLS and TCP. At the application level, Tor retranslates data in the fixed size packets called cells. All cells have equal length of 512 bytes and travel throughout the network in TLS records. It is noteworthy that several cells can be packed in a single TLS record. The last level is transport level: TLS record is then fragmented into several TCP packets. TCP packets size is limited by the MTU. Furthermore, several TLS records can be packed into a single TCP packet. However, it is questionable, which level is the most informative from the website

fingerprinting attack perspective. The majority of researchers assume that the most informative level is the cells level.

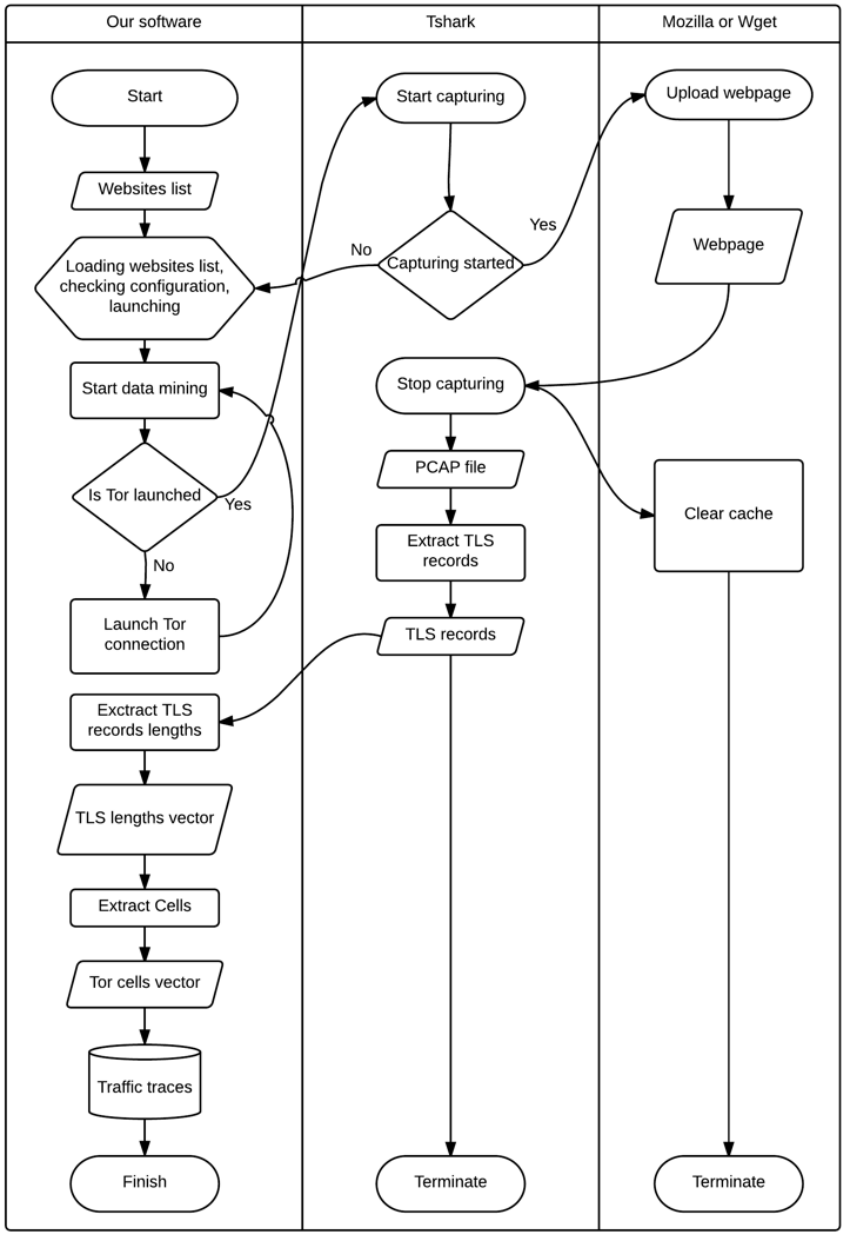


Fig. 3. Data mining process

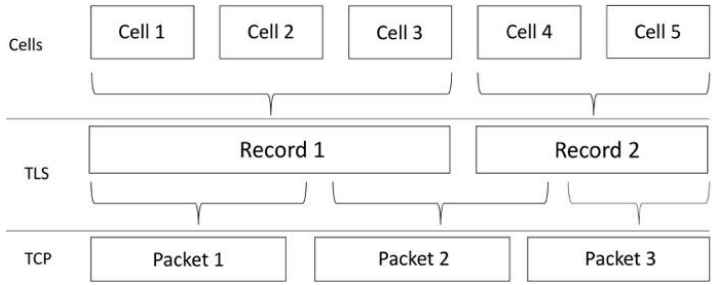


Fig. 4. Information extraction levels

Firstly, the cell traces extraction should be performed in the following way: an attacker must extract TLS records from TCP packets – it could be achieved with the tshark software.

Here the file_name should be substituted by the .pcap file with TCP packets, whereas ouput_file is the desired ouput file with textual representation of TLS records. Hence, a simple regular expression can than be used for length extraction. Once the number is an extended extraction, an attacker should then multiply it by -1 if it is outgoing.

The resulting array of TLS records lengths should then be transformed into Tor cells. An attacker should divide each number by 512 and append to the cells vector as many -1's or 1's as the number of integers found in the result of division. For example, if the length of TLS record is equal to 2048, the resulted cells vector would be [1,1,1,1].

After completion of cell traces extraction, we will have the representation of data in the form of [-1,1,1,1,-1,…]. Such arrays are then used as features, subsequently, the actual webpages are used as labels. However, such arrays have different lengths. Hence, as we are trying to simplify the process, we will append zeros to the end of input vectors because the majority of machine learning algorithms requires the input vectors to have equal lengths. By means of such operation, we will equalize the length of cell vectors.

5.3 Machine Learning Module

For machine learning purposes, we will use *sklearn* Python library, which is the most popular Python library for machine learning. The trained model will be used for classification of new traffic samples.

This module works in a straightforward way. An attacker must train the model using collected cells and then use it as a ready model.

6. Experimental setup

We have implemented such a scheme using Java programming language and Python (Fig. 5). The aim of our experiment is to show that we can deanonimize a small

fraction of users in the real world even if we don't use cutting-edge deanonimization techniques.

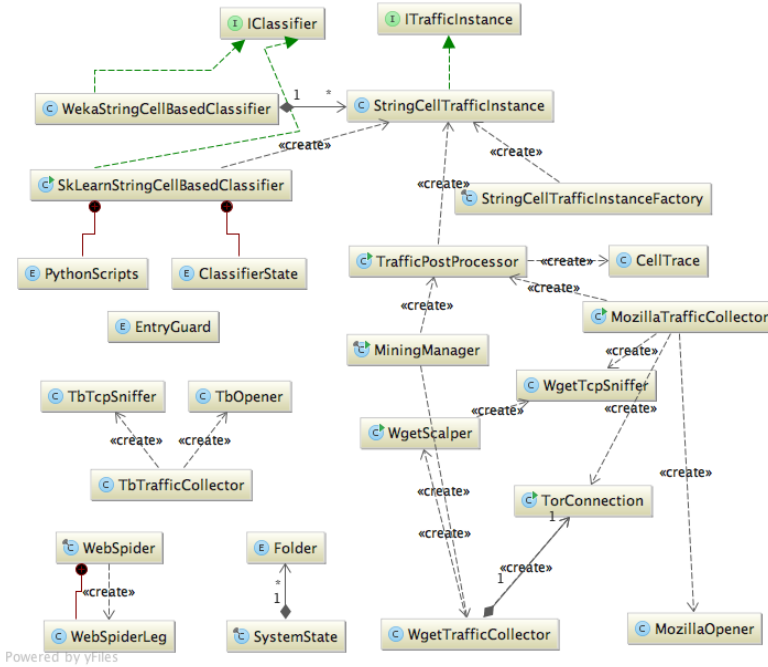


Fig. 5. UML class diagram of traffic collection module

6.1 Experimental Environment

Consider the following situation: the group of terrorists is trying to gain access to illegal content from a small room in the dormitory. The list of resources was provided by the Group-IB cybersecurity company. In our experiment there were three users playing the role of terrorists. Each of them visited the resources from the list according to the following rules: only single tab browsing is used, and the time spent to read the webpage is, at least, 5 seconds. According to the research [10], situation described looks pretty realistic. Such rules allow us to simplify the process of splitting packet sequences and extracting traces.

6.2 Data Gathering

Before trying to deanonimize users, we made a preparation step and collected 80 traffic instances from our list of resources. Such a low number of traffic instances is sufficient, because bigger datasets are not affecting accuracy of classifier on the

same number of websites. We have studied 7 resources related to drugs, weapons and extremism issues.

Our users repeated the process of reading and uploading a webpage 5 times for each webpage from the list. After that, we downloaded collected packet sequences and made the data preprocessing step. We used time-based splitting as was proposed by Wang [11]. After this step, our data became ready for classification.

6.3 Machine Learning Model

Support vector machines (SVM) are supervised learning models with associated learning algorithms that analyze data used for classification and regression analysis. An SVM model represents examples as points in space mapped so that the examples of separate categories are divided by a clear gap that is as wide as possible. New examples are then mapped into that same space and predicted as belonging to a category based on which side of the gap they fall to.

We used the NuSVC machine learning algorithm with default hyperparameters from the *sklearn* library. NuSVC is Nu-Support vector classification based on the support vector machines. This algorithm uses a parameter to control the number of support vectors, where the parameter is an upper bound of the fraction of training errors and lower bound of the fraction of support vectors.

7. Evaluation

7.1 Evalution metrics

- True positives (tp) - equal with hit.
- False positives (fp) - type I error, equal with correct rejection.
- False negatives (fn) - type II error.
- Precision - the ratio tp / (tp + fp); there is an intuitive ability of the classifier to avoid labelling a negative sample with the positive label.
- Recall - the ratio tp / (tp + fn); there is an intuitive ability of the classifier to find all the positive samples (the best is 1, the worst is 0).
- F1-score - a weighted average of the precision and recall (its best value is 1, the worst is 0) = 2 * (precision * recall) / (precision + recall).
- Score – the subset accuracy returned in a multilabel classification. If the entire set of predicted labels for a sample strictly matches the true set of labels, then the subset accuracy is 1.0, otherwise it is 0.0.

7.2 Experimental results

We have performed the classifier evaluation using a built-in *sklearn* function. For ethical reasons documented in Tor ethical research [12], we've anonymized the websites used in the experiment.

Our simple model has achieved results presented in table 2.

Table 2. Classifier evaluation

Website	Precision	Recall	F1-score
Site_1	1.00	1.00	1.00
Site_2	0.80	0.80	0.80
Site_3	0.80	0.80	0.80
Site_4	0.50	0.40	0.44
Site_5	1.00	1.00	1.00
Site_6	0.38	0.60	0.46
Site_7	0.67	0.40	0.50
Avg/total	0.73	0.71	0.72

Overall, the total score of the classifier = 0.714

These results are not outstanding in comparison with the state-of-the-art techniques, but they show that we can deanonimize users with the help of a relatively simple program and achieve sufficient accuracy.

8. Conclusion

It was shown that the attacker without cutting-edge machine learning techniques can apply website fingerprinting. If the attacker has enough experience and technical competence, he will be able to build such a system and use it for the purpose of deanonimization. Moreover, the proposed solution will work better if the attacker sniffs Wi-Fi or other local network, because it is very easy for him to find Tor related traffic and collect traces. In this case, the deanonimization is targeted and easily implemented.

9. Future work

In our future work, we are going to solve the Oracle problem using the recurrent neural networks and test them in the field of website fingerprinting attacks. Next, we are going to build a cloud application using state-of-the art techniques and results based on Recurrent Neural Networks research.

The main purpose of solving the Oracle problem is to have a pretty accurate splitting algorithm, which will allow to use WF attacks even with the multi-tab browsing.

References

[1]. S.M. Avdoshin, A.V. Lazarenko. [Technology of anonymous networks]. Informacionnye tehnologii [Information Technologies], vol. 22, №4, pp. 284-291, 2016 (in Russian).

[2]. R. Dingledine, N. Mathewson, P. Syverson. “Tor: The Second-Generation Onion Router”. In Proceedings of the 13th USENIX Security Symposium, August 2004 (online publication). Available at: http://www.onion-router.net/Publications/tor-design.pdf, accessed 12.07.2016.

[3]. Relays and bridges in the network (online publication). Tor METRICS [Official website]. Available at: https://metrics.torproject.org/networksize.html, accessed 12.07.2016.

[4]. The NSA's Been Trying to Hack into Tor's Anonymous Internet For Years (online publication). Gizmodo [Official website]. Available at: http://gizmodo.com/the-nsas-been-trying-to-hack-into-tors-anonymous-inte-1441153819, accessed 12.07.2016.
[5]. Zakupka No0373100088714000008 (online publication). Gosudarstvennie zakupky [State Procurements] [Official website]. Available at: http://zakupki.gov.ru/epz/order/notice/zkk44/view/common-info.html?regNumber=0373100088714000008, accessed 12.07.2016 (in Russian).
[6]. S.M. Avdoshin, A.V. Lazarenko, [Tor Users Deanonimization Methods]. Informacionnye tehnologii [Information Technologies], vol. 22, №5, pp. 362-372, 2016 (in Russian).
[7]. X. Cai, X.C. Zhang, B. Joshy, R. Johnson. Touching from a Distance: Website Fingerprinting Attacks and Defenses (online publication). Available at: http://www3.cs.stonybrook.edu/~xcai/fp.pdf, accessed 12.07.2016.
[8]. T. Wang, Website Fingerprinting: Attacks and Defenses, PhD Thesis (online publication), 2015. Available at: https://uwspace.uwaterloo.ca/bitstream/handle/10012/10123/Wang_Tao.pdf?sequence=3, accessed 12.07.2016.
[9]. X.Gu, M.Yang, J.Luo. A Novel Website Fingerprinting Attack Against Multi-Tab Browsing Behavior (online publication). In Computer Supported Cooperative Work in Design (CSWD), 2015. Available at: http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=7230964&url=http%3A%2F%2Fieeexplore.ieee.org%2Fxpls%2Fabs_all.jsp%3Farnumber%3D7230964, accessed: 12.07.2016.
[10]. A. Panchenko, F. Lanze, A. Zinnden, M. Henze, J. Pannekamp, K. Wehrle, T. Engel. Website Fingerprinting at Internet Scale (online publication). Available at: https://www.comsys.rwth-aachen.de/fileadmin/papers/2016/2016-panchenko-ndss-fingerprinting.pdf, accessed 12.07.2016.
[11]. J.Nielsen. How Long Do Users Stay on Web Pages (online publication), Available at: https://www.nngroup.com/articles/how-long-do-users-stay-on-web-pages/, accessed 12.07.2016.
[12]. Ethical Tor Research: Guidelines (online publication). Available at: https://blog.torproject.org/blog/ethical-tor-research-guidelines, accessed 12.07.2016.

Система деанонимизации пользователей теневого интернета

С.М. Авдошин <savdoshin@hse.ru>
А.В. Лазаренко <avlazarenko @edu.hse.ru>
Департамент программной инженерии,
Национальный исследовательский университет "Высшая школа экономики",
101000, Россия, г. Москва, ул. Мясницкая, д. 20.

Аннотация. Технологии обеспечения пользовательской приватности являются неотъемлемой частью жизни современных людей. Они востребованны широким пользовательским сегментом. Однако такие инструменты зачастую используются для мошеннической и нелегальной деятельности. В современном мире есть много сетей и технологий, которые предоставляют анонимный доступ к ресурсам сети. Наиболее

распространенной и широко используемой анонимной сетью является Tor. При этом именно Tor является основным инструментом многочисленных хакеров, торговцев наркотиками и оружием. Настоящая статья фокусируется на деанонимизации пользователей Tor с применением доступных в интернете технологий и базового алгоритма машинного обучения. Цель работы – показать, что деанонимизация небольшого количества пользователей возможна без использования большого количества вычислительных ресурсов. В начале работы представлен обзор различных анонимных сетей. Затем - различные методы деанонимизации: анализ траффика, тайминг атаки, атаки на уровне автономных систем. Построена классификация атак по ресурсам, необходимым атакующим для успешного применения. Для реализации была выбрана website fingerprinting атака. Эта атака требует наименьшего количества ресурсов для ее использования и внедрения в сеть Tor с целью успешной деанонимизации пользователей. Описан эксперимент использования website fingerprinting атаки. Список отслеживаемых в эксперименте ресурсов был получен от компании, специализирующейся в области информационной безопасности. Эксперимент проводился в одной комнате при участи 5 человек и одного входного узла. Была достигнута точность классификации просматриваемых страниц равная 70% процентам. Задача деанонимизации крайне важна для национальной безопасности, что подчеркивает актуальность проведенного исследования.

Ключевые слова: Tor; деанонимизация; website fingerprinting; анализ трафика; анонимная сеть; теневой интернет.

DOI: 10.15514/ISPRAS-2016-28(3)-2

Для цитирования: Авдошин С.М., Лазаренко А.В.. Система деанонимизации пользователей теневого интернета. Труды ИСП РАН, том 28, вып. 3, 2016 г. стр. 21-34 (на английском). DOI: 10.15514/ISPRAS-2016-28(3)-2

Список литературы

[1]. Авдошин С.М., Лазаренко А.В. Технология анонимных сетей // Информационные технологии. 2016. Т. 22, № 4, стр. 284-291.

[2]. R. Dingledine, N. Mathewson, P. Syverson. Tor: The Second-Generation Onion Router, in Proceedings of the 13th USENIX Security Symposium, August 2004. URL: http://www.onion-router.net/Publications/tor-design.pdf, 12.07.2016.

[3]. Relays and bridges in the network (online). Tor METRICS [Official website], Доступно по ссылке: https://metrics.torproject.org/networksize.html, 12.07.2016.

[4]. The NSA's Been Trying to Hack into Tor's Anonymous Internet For Years (online), Gizmodo [Official website], Доступно по ссылке: http://gizmodo.com/the-nsas-been-trying-to-hack-into-tors-anonymous-inte-1441153819, 12.07.2016.

[5]. Закупка № 0373100088714000008 (online). Государственные закупки [Официальный сайт]. Доступно по ссылке: http://zakupki.gov.ru/epz/order/notice/zkk44/view/common-info.html?regNumber=0373100088714000008, 12.07.2016.

[6]. Авдошин С.М., Лазаренко А.В. Методы деанонимизации пользователей TOR // Информационные технологии. 2016. Т. 22, № 5, стр. 362-372.

[7]. X. Cai, X.C. Zhang, B. Joshy, R. Johnson. Touching from a Distance: Website Fingerprinting Attacks and Defenses (online). Доступно по ссылке: http://www3.cs.stonybrook.edu/~xcai/fp.pdf, 12.07.2016.

[8]. T. Wang, “Website Fingerprinting: Attacks and Defenses”, PhD Thesis, 2015 (online). Доступно по ссылке: https://uwspace.uwaterloo.ca/bitstream/handle/10012/10123/Wang_Tao.pdf?sequence=3 12.07.2016.

[9]. X.Gu, M.Yang, J.Luo. A Novel Website Fingerprinting Attack Against Multi-Tab Browsing Behavior, in Computer Supported Cooperative Work in Design (CSWD), 2015 (online). Доступно по ссылке: http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=7230964&url=http%3A%2F%2Fieeexplore.ieee.org%2Fxpls%2Fabs_all.jsp%3Farnumber%3D7230964, accessed: 12.07.2016.

[10]. A. Panchenko, F. Lanze, A. Zinnden, M. Henze, J. Pannekamp, K. Wehrle, T. Engel. , Website Fingerprinting at Internet Scale (online). Доступно по ссылке: https://www.comsys.rwth-aachen.de/fileadmin/papers/2016/2016-panchenko-ndss-fingerprinting.pdf, accessed: 12.07.2016.

[11]. J.Nielsen. How Long Do Users Stay on Web Pages (online). Доступно по ссылке: https://www.nngroup.com/articles/how-long-do-users-stay-on-web-pages/, accessed: 12.07.2016.

[12]. Ethical Tor Research: Guidelines (online). Доступно по ссылке: https://blog.torproject.org/blog/ethical-tor-research-guidelines, accessed: 12.07.2016.

Model of security for object-oriented and object-attributed applications

[1] *Pavel P. Oleynik, PhD <xsl@list.ru>*
[2] *Sergey M. Salibekyan, PhD <ssalibekyan@hse.ru>*
[1] *Platov Southern Russian State Polytechnic University (NPI), 1 Lenin sq., Shakhty, 346500, Russian Federation*
[2] *National Research University "Higher School of Economics" (NRU HSE), Institute of Electronics and Mathematics, 20 Myasnitskaya str., Moscow, 101000, Russian Federation*

Abstract. The article describes two approaches for control access rights based on role approach (RBAC) and the use of tables (lists) access rights (ACL). At first, an overview of modern approaches to information security and control user access rights of applications with different architectures is provided. After that, two author's methods of data protection is described. The first approach was developed for the protection of object-oriented applications, the second approach was developed for object-attribute applications used to operating network (graph) databases and knowledge bases. The focus of attention is the first author's approach based on the description of access rights for classes, attributes of classes and objects that has a certain criterion. The approach is implemented by the use of a class hierarchy, composition and structure describing in detail in the article. The article gives examples of specific information systems developed by the first author: information system for scientific conferences that was repeatedly used at the conference "Object systems" (objectsystems.ru) and information system of the beauty salon. Further focus is on the second approach required development of new technique to the information security of network (graph) information structures. The approach developed by second author fully duplicates the functionality of the first approach. In particular, it provides permissions copy when copying of the network data structure, just as in the object-oriented paradigm is a transfer of the properties of parent to child class; the article gives a detailed description of such mechanism. For access control, the method involves the use of a special virtual device. Information about access rights is linked to the node network (graph) if restrict access is needed.

Keywords: Security of information systems; Object-oriented applications; Object System Metamodel; Model of Permissions; object-attribute approach.

DOI: 10.15514/ISPRAS-2016-28(3)-3

For citation: Oleynik P.P., Salibekyan S.M. Model of security for object-oriented and object-attributed applications. Trudy ISP RAN/Proc. ISP RAS, vol. 28, issue 3, pp. 35-50. DOI: 10.15514/ISPRAS-2016-28(3)-3

1. Introduction

At present, the greatest number of new applications is being developed by an object-oriented approach. This paradigm, based on the inheritance technology, allows one to reuse the previously developed elements implemented as classes. The result is the reduced development time and the costs of the whole information system. This is the key advantage when large software products are created. Such systems are typically multi-user systems. At the same time, each category of user needs is only a part of the available information, i.e. there is a problem of access control for multi-user applications. The paper presents a model of access control for object-oriented applications, which was developed by the authors and repeatedly used when developing large applications, and a model of access control in object-attribute computation system.

The paper is organized as follows. Section 1 provides a detailed survey of the papers devoted to similar topics. Section 2 describes the model of access control used by the author. Section 3 shows real examples of implementation of this model and the selected roles of users. Section 4 shows the approach to security in Object-attribute system. At the end of the paper, conclusions on this work and plans for the further study are given.

2. A survey of the available research

Access permission is one of the main problems appearing after the development of the required functionality of the program. Therefore, there are a lot of researches representing different approaches to solving this problem. In [1], the authors propose an approach called business-oriented development (Business-Driven Development), in which the key role is given to the security configuration in the application. The authors use the Model-Driven Architecture (MDA) of architecture of the program. They introduce the concepts of business processes and models at the model level, and then determine the security policies and templates specifying certain rules for them. The present research describes principles of access permission assignment at the level of platform-independent models and the further transformation into platform-dependent models. As a result, the authors present a set of templates for access control providing that their configuration can be adjusted if necessary. This solution is tested using a service-oriented architecture (SOA). To improve the efficiency of the description of the software product life cycle and the corresponding access permissions, the authors propose to make several changes in the languages of software development, such as UML and BPEL. An advantage of the paper is the presence of a number of charts illustrating the proposed solution, as well as many code fragments represented as XML.

The research [2] is more practical and special. It describes a model of adaptive security for multi-agent information systems used by the authors in the medical information system called HealthAgents. The authors start from describing the classical model of access control based on Role-based access control (RBAC) and extend it to be used in multi-agent systems. In their research, the authors present a meta-model that allows one to manage access control by using the UML class diagram. To interact with the security role, the authors introduce the base class Subject attributed with different user permissions. The derived class represents users, organizations and agents. An analysis of research shows that the object-oriented approach for describing access rights is implemented. To describe the process of applying the security policies, the authors depict the Interaction Diagram and present, in the XML-code, an example of test description of access rights of certain users, stored in the system.

The research [3] presents the simulation of multi-level security, integrated within a service-oriented application. In a service-oriented architecture (SOA) that allows one to develop different Web applications, the security is critical. The security is provided by the Web service WS-Security controlled by SOAP messages. These messages may be attacked either by anonymous customers or by trusted clients. In addition, there are other possible types of attacks, for example, the so-called denial of service (DoS), which can exhaust the computer resources and make the Web service unavailable. The described security model consists of three levels. Attention is paid to each of the levels. The obtained multi-level security architecture is presented graphically, namely, various security domains, as well as the composition and structure of the software installed on each of them are depicted. After this, various types of possible attacks at each of the levels are discussed. They are described using the UML Class Diagram. This allows one to analyze the results obtained by the authors and then to design the desired security models based on the results.

The framework for describing the security model of service-oriented applications (SOA) is presented in [4]. The authors focusing on the process of modeling business processes use the BPEL notation. The security model is used with the model of business processes. The authors argue that the difference in approaches of a Business analyst and an Expert to solving the security problems leads to certain permission assignment that ultimately compromise the safety of user data. The authors developed several annotations that allow the security Experts to specify the security model. The proposed approach is demonstrated by an example of business processes of a service-oriented information system providing data about the progress of students. The paper describes a possible implementation of the framework, its basic modules and rules of interaction between the experts and the system.

The paper [5] presents model-oriented templates (patterns) of application security obtained by the authors by an analysis of phases of the application development. The authors examine the applications working in Internet. The templates contain descriptions of solutions to common security problems. The selection of an

appropriate pattern depends not only on the situation but on other templates applied earlier, i.e. the dependence between the patterns is taken into account. The authors present an analysis of such dependencies for the first time. The technology of changes of General security templates is proposed on the basis of a rule transformation model based on previously used patterns. This allows one to avoid inappropriate application of the security templates. The authors identify two levels of abstraction: 1) the analysis Phase; 2) the design Phase. Certain modules are responsible for each of them. The software structure and the functions of the modules are considered in detail by the authors. In conclusion, the authors present the syntax of the language used to describe the transformation rules of different patterns. This is similar to languages such as SQL, OCL, LINQ. To demonstrate the obtained results, the authors describe the test information system containing information about the patients of a hospital. The use chart (Use Case) shows the different categories of users and the types of the applied security patterns. Then the structure of the template and the class diagram of the subject area after the application of this decision are illustrated in the form of a UML Class Diagram. This approach is applied to all selected templates, and the complexity of manual and automated applications is evaluated.

In [6], the model-oriented approach to the security applied in the information system of electronic voting is presented. The necessary security requirements, illustrated as the Use Diagrams of UML, were represented as functional requirements at the requirement formalization stage. After this, the authors describe the step-by-step algorithm for identifying and implementing the security requirements and then describe each key element in detail. The paper presents the application architecture and the main computing nodes (computers) which play a certain role. This allows the authors to determine possible vulnerability and attacks against which the system should be projected. The authors also present an approach to the security model implementation in the information system of electronic voting. The model is illustrated by the Sequence Diagram of language UML.

3. The model of access control

Currently, the classical model access control based on roles (Role-based access control, RBAC) has been widely used. Appeared in operating systems, it has the form presented in fig. 1.

This model is popular due to its plain architecture whose functions are as follows. The security system (model) creates multiple roles represented by the Role class. Each role is assigned certain access permissions represented by the Permission class. Permissions are assigned to different objects in the system, which is represented by the class Object. The user described by the User class is attached to at least one role. Moreover, these roles can be inherited, and this can simplify the process of assigning permissions to objects. This scheme is optimal for delineation of rights for objects of one type, for instance, for managing the permissions of access to file system objects (files, directories) in an operating system.

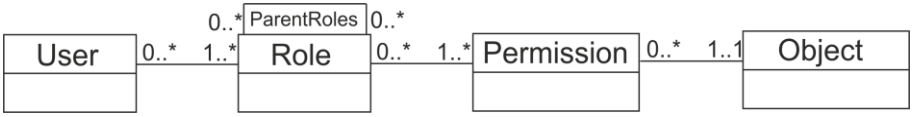


Fig. 1. Classical Role-based Access Control, (RBAC) model

Software applications written in object-oriented programming languages require another security means because it are several types of objects that can be attributed by rights. For the optimal systems design the following optimality criteria (OC) for features are selected:

- access rights for classes (OC1);
- access rights for class properties (OC2);
- access rights for objects (instances of classes) (OC3).

Fig. 2 shows the structure of an optimal model of access rights management for object-oriented applications.

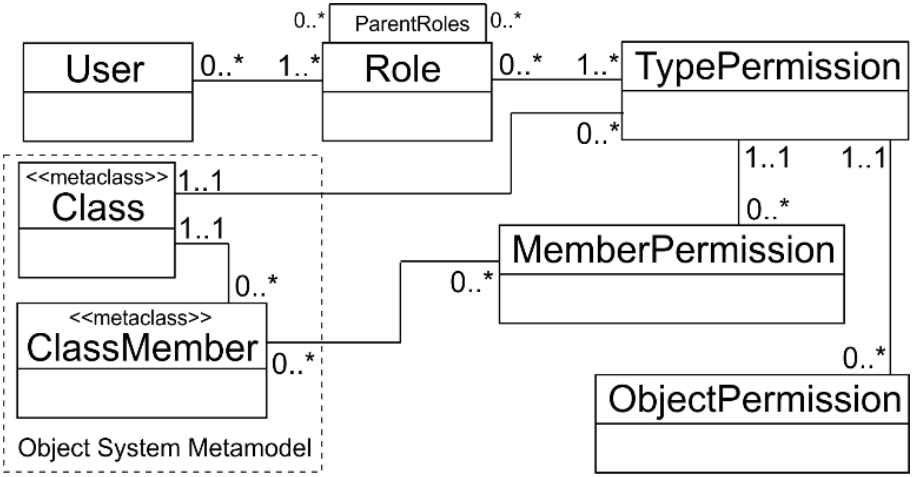


Fig. 2. Classical Role-based Access Control, (RBAC) model

We will examine this figure in more detail. To describe the objects which can be assigned the access rights, an advanced meta-model of the object system is used. In our case, it is enough to have information about the class and attributes (properties) of classes. To match the selected OC1, the class TypePermission which allows differentiating the access rights for the classes is designed. To differentiate the rights according to the properties of classes (see OC2), the class MemberPermission

is introduced. The Class ObjectPermission is used to set permissions on the class copies corresponding to the OC3 requirement.

After clearing the structure and concept implementation, we begin to study of the final system. Fig. 3 shows the implemented-by-authors model of access control for object-oriented applications in the form of class diagrams.

We will consider fig. 3 in more detail. All base classes implementing the key functionality of the security system have names ending by the suffix Base. So the SecuritySystemRoleBase and SecuritySystemUserBase classes form the root class for representing the roles of security and the system user respectively. The TypePermissionMatrixItem class is used to specify the data type (class name) which needs the access rights. The following permission types are used for the classes:

- AllowCreate allows the user to create objects (class instance);
- AllowCreate allows the user to delete objects (class instance);
- AllowNavigate allows the user to display a menu item to view the class instance;
- AllowRead allows the user to view objects of the class;
- AllowWrite allows the user to replace some objects of the class by other.

The class SecuritySystemMemberPermissionsObject allows one to describe the rights to some individual properties and to implement a complex security policy in which the user is prohibited from reading certain attributes of the class.

The class SecuritySystemObjectPermissionsObject is used to distinguish the rights between individual objects of the class which satisfy some predicate. This condition holds in the property Criteria.

The UML diagram shows the relationship between associations which allows one to understand the relationship between classes. In the end, it should be noted that the developed security system allows an unlimited description of the types of access rights in an object-oriented system, which corresponds to the previously identified optimality criteria.

4. Examples of using the model of access control

To implement the above-described model of access control, it is very important to have the meta-information of the object system. The model is physically stored in a relational database according to the principles described in [7]. When designing a meta-model, the key challenge was to develop a hierarchy of meta-classes which allows one to save information about literal types and different classes of domain entities [8-9]. The design of the developed meta-model allows one to realize the subject-oriented approach to designing database applications for different fields [11-13]. In [14-16], the use of the metamodel in the design of information systems is described.

Then paper [16] describes the previously-used security model for access rights applied to an information system used to carry out scientific conferences. The model

was repeatedly employed to manage the conference "Object system" (objectsystems.ru). Attention was paid to the security issues at the design stage. For this, the following roles were allocated to the users in the system:

1. The organizer of the conference. He is the main person and the user of the system. His responsibilities include the following tasks:

1. to register the publications;
2. to appoint the reviewer;
3. to verify the corrections made by the authors according to the reviewer comments;
4. to check the payments;
5. to prepare the journal;
6. to send the proceeding books and certificates to the authors of the papers.

2. **The author** writes a paper and sends it to the conference. The author's responsibility is also to revise the paper according to the reviewer's comments about the paper and, if necessary, to pay the registration fee.

3. **The reviewer** checks the author's paper and evaluates its quality. The review includes: to write a review indicating the observations and recommendations for its improvement; to formulate the review result (to accept the paper for publication or to reject it or to send it back for revision). During the preparation of the conference proceedings, the reviewers award nominations to the best papers submitted to the conference. However, in the general case, there are several reviewers.

On the basis of this information, classes and types of access are detected for different roles. Next, instances of classes presented in Figure 3 are created.

The paper [17] describes an information system of a beauty salon. Studying the business logic in this field shows that the system must implement a variety of different financial calculations determining the costs and profitability of the salon. This information can be presented only to the owner of the salon. The following roles are emphasized:

1. **Master**. Main task of the master is to provide services to clients. Therefore, each master can only view (read) the main system directories such as: Operating Schedule, Record/Visit, Schedule of visits, Customer, Leave/Sick leave/Compensatory leave/Absence, Service, Commodity, Certificate, Price, Interest, Master, master Category, room Category, Remnants of goods, Work schedule, Working hours;

2. **Salon administrator**. The main task of a manger is to monitor the activities of the salon. Namely, an administrator registers clients and monitors progress of master work. In the system, an administrator has right to add/edit/delete data from the directories: Visiting Schedule, Customer Master, work Schedule, Record/Attendance, Vacation/Sick leave/Compensatory leave/Absence, Service,

Commodity, Certificate, Discount, goods Receipt, Inventory, Price, Stock, Percent, client Category, master Category, service Category, Document, Movement of goods, remaining Stock, Sales, Salon, Working hours, Working time;

3. **Owner of the Salon** has all the same rights as the Administrator of the salon. In addition, he has right to view information from processed forms such as: Wages, Profit, and Profitability. The salon owner can also introduce new users in the system and add them only to the existing roles.

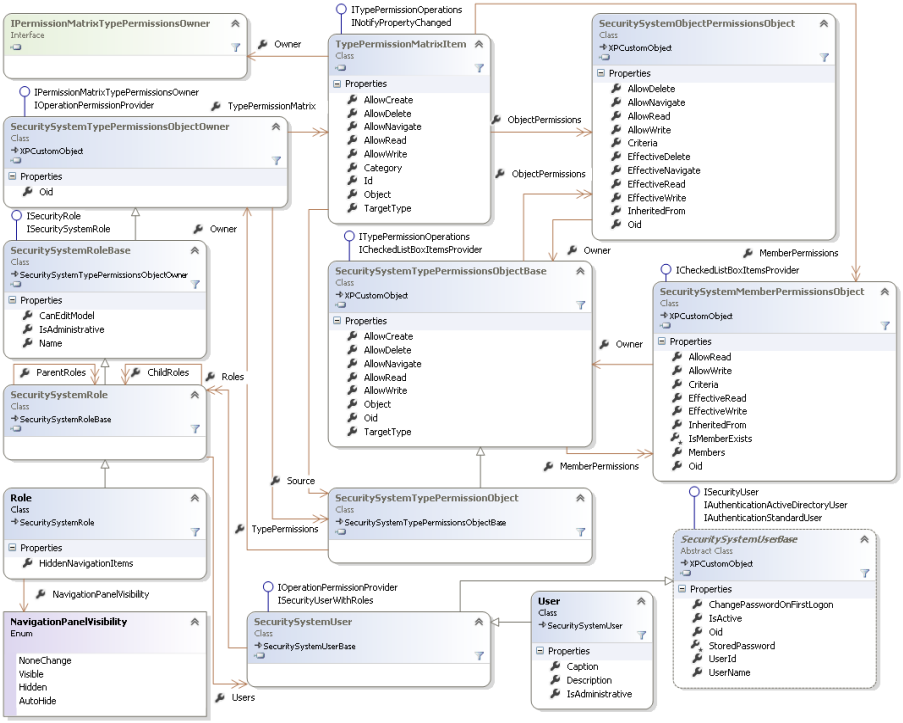


Fig. 3. UML class diagram of the implemented model of access rights differentiation

The papers [18-20] describe the information system architecture of fast food restaurants. The key feature of application of this class is that they are used in the places of public service with a large number of clients. In such software products, the critical maintenance time is very important, and so the graphical interface of the user must be ergonomic. The monoblocks with touch screens are often used as the hardware platform in such systems. Therefore, in such applications, attention is paid

to the graphical interface of the user and to the principles of security settings. In this case, the following roles are selected:

- **Waiter**. The waiter's main task is to create purchase orders, to add the goods purchased by clients to the orders, and to arrange the payment;
- **Cashier**. A cashier cannot create new orders but can remove erroneous orders, view all orders issued in the current and previous shifts, and also issue the payment orders;
- **Manager**. His main task is to form consolidated reports on the work of a shift and to add new waiters and cashiers to the system;
- **Merchandiser**. The main task of the merchandiser is to introduce information about new food into the system.

When designing each of the above-described applications, the role of system administrator, who sets permissions for the existing roles and creates new roles, was also assigned. In fact, this role corresponds to the system administrator of a domain of the Windows operating system.

5. Information security in OA-systems

The OA-approach to organization of the data structure and the computational process is currently being developed. The approach implements the object-oriented (OO) programming principle with a few other features [21,22,23]. The OA-approach requires new methods for the information security organization.

Unlike the OO paradigm, in OA, there is no distinction between the concepts of class and object. Instead of the class, a semantic network template, which is copied to generate a new semantic network, is used [24]. Also, there is no such a concept as the field of an object: a data and a program are represented as an information capsule (IC). Therefore, in the OA-system, the data security is focused on an information capsule (IC), and the OA-graph is protected through it. Let us explain it. The functional unit (FU) processes an OA-graph. Let us call it a processing FU. The processing FU usually takes reference to one of the IC (starting IC) of the OA-graph and produces a traversal from the IC. The traversal is performed as follows. A FU looks for the information pair (IP) in the IC with a specific attribute and goes by the link contained in its load to another IC of the OA-graph. Thus, the OA-graph security is provided through the security of the starting IC. Any other IC may be secured in the OA-graph similarly to the protection of the object field in the OO paradigm.

For the implementation of information security, a specialized FU, called the "Guard", is required. The functions of the FU are the control of the user accounts and roles (if the RBAC approach is used) and the creation and control of the access control list (ACL) for IC contained in the OA-graph. The Guard integrated to the processing FU controls the access permissions to a IC. The control is ensured as follows: operating FU before the analysis, the IC passes a reference to the access

controller that checks the access permission to IC. If the access is denied, then the Guard blocks the FU performing the OA-graph traversal.

The access permissions information is stored in the ACL (fig. 4). The ACL can be attributed to the IC of the OA-graph by adding IP, called the security IP, with the attribute "ACL", the load of the IP contains a pointer to the ACL (one ACL can be assigned to one or several IC.). To prevent unauthorized access to the ACLs, the manipulation protection of security IP is included in the algorithm for controlling the processing FU: prohibition to remove the secure IP (the IP can only be removed during the removal of the IC, where the IP is located), prohibition to use the reference of the secure IP load, etc. The ACL is processed (creation, destruction and modification) by the Guard.

The proposed mechanism well emulates the protection class in the OO paradigm. If the secure IP is contained in the OA-graph, then when copying the OA-graph, the secure IP with the load containing the reference to the access rights matrix is copied too.

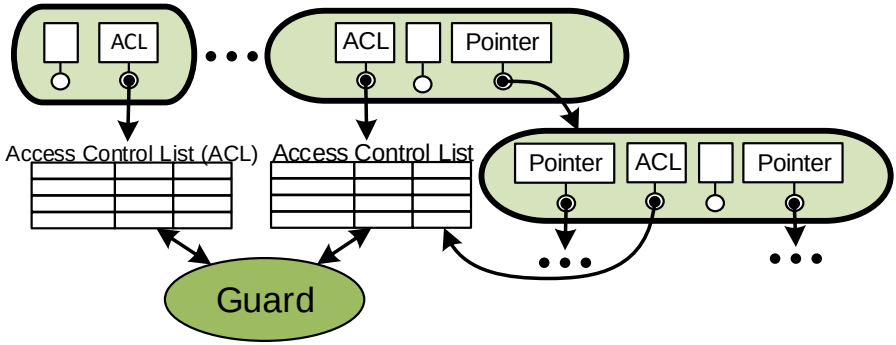


Fig. 4. The mechanism of data security in OA-computing system

The proposed methodology provides maximum flexibility of the security mechanism of the OA-graph and corresponds to all three criteria (OC1, OC2, OC3) applicable to the security of OO systems, i.e., protection of OA-graph (similar to object), a separate IC (similar to object fields), and OA-graphs copied from the OA-graph template (similar to the class protection). Moreover, all criteria are satisfied with a single protection mechanism.

6. Conclusions and further research

The above description shows that the established model of differentiation of access rights can successfully be used in applications in various domains, i.e. it is universal. Several applications where the security comes first are currently designed and implemented. This allows testing the proposed model completely and modifying it in accordance with the discovered drawbacks.

The model was developed in the OA-approach. The model is quite simple and satisfies all criteria for the security in the OO approach.

References

[1]. Nagaratnam N., Nadalin A., Hondo M., McIntosh M., Austel P. Business-driven application security: from modeling to managing secure applications. IBM Systems Journal, vol. 44, issue 4, 2005, pp. 847-867.

[2]. Xiao L., Peet A., Lewis P., Dashmapatra S., Saez C., Croitoru M., Vicente J., Gonzalez-Velez H., Lluch i Ariet M. An Adaptive Security Model for Multi-agent Systems and Application to a Clinical Trials Environment. 31st Annual International Computer Software and Applications Conference, COMPSAC 2007, 24-27 July 2007, Beijing, China, 2007, pp. 261-268.

[3]. Fengyu Zhao, Xin Peng, Wenyun Zhao. Multi-Tier Security Feature Modeling for Service-Oriented Application Integration. Eighth IEEE/ACIS International Conference on Computer and Information Science, ICIS 2009, 1-3 June 2009, Shanghai, China, 2009, pp. 1178-1183.

[4]. Saleem M.Q., Jaafar J., Hassan M.F. Model Driven Security Framework for Definition of Security Requirements for SOA Based Applications. 2010 International Conference on Computer Applications and Industrial Electronics (ICCAIE), 5-8 Dec. 2010, Kuala Lumpur, 2010, pp. 266-270.

[5]. Shiroma Y., Washizaki H., Fukazawa Y., Kubo A., Yoshioka N. Model-Driven Security Patterns Application Based on Dependences among Patterns. ARES '10 International Conference on Availability, Reliability, and Security, 15-18 Feb. 2010, Krakow, Poland, 2010, pp. 555-559.

[6]. Salini P., Kanmani S. Application of Model Oriented Security Requirements Engineering Framework for Secure E-Voting. 2012 CSI Sixth International Conference on Software Engineering (CONSEG), 5-7 Sept. 2012, Indore, 2012, pp. 1-6.

[7]. Oleynik P.P. Resentating metamodel of object system in a relational database. Izvestiya vysshikh uchebnykh zavedeniy. Severo-Kavkazskiy region [UNIVERSITY NEWS. NORTH-CAUCASIAN REGION]. Spetsvypusk «Matematicheskoe modelirovanie i komp'yuternye tekhnologii» [Special Issue "Mathematical modeling and computer technologies»], pp. 3-8, 2005 (in Russian).

[8]. Oleynik P.P. Implementation of the Hierarchy of Atomic Literal Types in an Object System Based of RDBMS. Programming and Computer Software, vol. 35, no.4, pp. 235-240, 2009.

[9]. Oleynik P.P. Class Hierarchy of Object System Metamodel. Ob'ektnye sistemy – 2012: materialy VI Mezhdunarodnoj nauchno-prakticheskoj konferencii, Rostov-na-Donu, 10-12 maja 2012 g. [Object Systems – 2012: Proceedings of the Sixth International Theoretical and Practical Conference. Rostov-on-Don, Russia, 10-12 May, 2012]. pp. 37-40 (In Russian). Available at: http://objectsystems.ru/files/2012/Object_Systems_2012_Proceedings.pdf

[10]. Oleynik P.P. Domain-driven design of the database structure in terms of object system metamodel. Ob'ektnye sistemy – 2012: materialy VI Mezhdunarodnoj nauchno-prakticheskoj konferencii, Rostov-na-Donu, 10-12 maja 2012 g. [Object Systems – 2014: Proceedings of the Eighth International Theoretical and Practical Conference, Rostov-on-Don, 10-12 May, 2014], pp. 41-46 (In Russian). Available at: http://objectsystems.ru/files/2014/Object_Systems_2014_Proceedings.pdf

[11]. Oleynik P.P. Using metamodel of object system for domain-driven design the database structure // Proceedings of 12th IEEE East-West Design & Test Symposium (EWDTS'2014), Kiev, Ukraine, September 26 – 29, 2014, pp. 79-86. DOI: 10.1109/EWDTS.2014.7027052

[12]. Oleynik P.P. Unified Metamodel of Object System. Ob'ektnye sistemy – 2015: materialy X Mezhdunarodnoj nauchno-prakticheskoj konferencii, Rostov-na-Donu, 10-12 maja 2015 g. [Object Systems – 2015: Proceedings of X International Theoretical and Practical Conference, Rostov-on-Don, 10-12 May, 2015], pp. 79-85. Available at: http://objectsystems.ru/files/2015/Object_Systems_2015_Proceedings.pdf

[13]. Oleynik P.P. The Elements of Development Environment for Information Systems Based on Metamodel of Object System. Biznes-informatika [Business Informatics], №4(26), pp. 69-76, 2013 (In Russian). http://bijournal.hse.ru/data/2014/01/16/1326593606/1BI%204(26)%202013.pdf

[14]. Oleynik P.P., Kurakov Yu.I. The Concept Creation Service Corporate Information Systems of Economic Industrial Energy Cluster. Prikladnaja informatika [Applied Informatics], №6. pp. 5-23, 2014 (In Russian).

[15]. Kurakov Y. I., Oleynik P. P. Implementation method a unified information system of economic production and energy cluster in coal industry. Gornyj informacionno-analiticheskij bjulleten' [Mining information-analytical Bulletin, no. 6, pp. 260-273, 2015 (In Russian).

[16]. Borodina N.E., Oleynik P.P., Galiaskarov E.G. Reengineering of Object Model by the Example of Information System for Cataloging Scientific Articles for International Conferences. Ob'ektnye sistemy – 2014 (zimnjaja sessija): materialy IX Mezhdunarodnoj nauchno-prakticheskoj konferencii, Rostov-na-Donu, 10-12 dekabrja 2014 g. [Object Systems – 2014 (Winter session): Proceedings of IX International Theoretical and Practical Conference, Rostov-on-Don, 10-12 December, 2014], pp. 17-23 (In Russian). Available at: http://objectsystems.ru/files/2014WS/Object_Systems_2014_Winter_session_Proceedings.pdf

[17]. Kozlova K.O., Borodina N.E., Galiaskarov E.G., Oleynik P.P. Domain-Driven Design of Information System of a Beauty Salon in Terms of Unified Metamodel of Object System. Ob'ektnye sistemy – 2015: materialy X Mezhdunarodnoj nauchno-prakticheskoj konferencii, Rostov-na-Donu, 10-12 maja 2015 g. [Object Systems – 2015: Proceedings of X International Theoretical and Practical Conference, Rostov-on-Don, 10-12 May, 2015], pp. 86-90 (In Russian). Available at:http://objectsystems.ru/files/2015/Object_Systems_2015_Proceedings.pdf

[18]. Oleynik P.P, Yuzefova S.Yu., Nikolenko O.I. Experience in Designing an Information System for Fast Food Restaurants. Ob'ektnye sistemy – 2014 (zimnjaja sessija): materialy IX Mezhdunarodnoj nauchno-prakticheskoj konferencii, Rostov-na-Donu, 10-12 dekabrja 2014 g. [Object Systems – 2014 (Winter session): Proceedings of IX International Theoretical and Practical Conference, Rostov-on-Don, 10-12 December, 2014], pp. 12-16 (In Russian). Available at: http://objectsystems.ru/files/2014WS/Object_Systems_2014_Winter_session_Proceedings.pd

[19]. Nikolenko O.I., Oleynik P.P, Yuzefova S.Yu. Prototyping and Implementation of Graphical Order Form for the Information System of Fast Food Restaurants. Ob'ektnye sistemy – 2015: materialy X Mezhdunarodnoj nauchno-prakticheskoj konferencii, Rostov-na-Donu, 10-12 maja 2015 g. [Object Systems – 2015: Proceedings of X

International Theoretical and Practical Conference, Rostov-on-Don, 10-12 May, 2015], pp. 68-72 (In Russian). Available at: http://objectsystems.ru/files/2015/Object_Systems_2015_Proceedings.pdf

[20]. Pavel P. Oleynik, Olga I. Nikolenko, Svetlana Yu. Yuzefova. Information System for Fast Food Restaurants. Engineering and Technology, vol. 2, no. 4, 2015, pp. 186-191. Available at: http://article.aascit.org/file/pdf/9020895.pdf

[21]. P. B. Panfilow, S. M. Salibekyan Dataflow Computing and its Impact on Automation Applications. Procedia Engineering, vol. 69, 2014., pp. 1286-1295. URL: http://www.sciencedirect.com/science/article/pii/S1877705814003671

[22]. Pavel P. Oleynik, Sergey M. Salibekyan. The Approaches to Implementation of Patterns of Static Object Models for Database Applications: Existing Solutions and Unified Testing Model. International Journal of Applied Engineering Research, vol. 10, no. 24 2014, pp 45513-45516.

[23]. Salibekyan S.M., Panfilov P. B Object-Attribute Architecture is a New Approach to Object Systems Developing. Informacionnye tehnologii [Information technologies], no.2, 2012, pp 8-14.

[24]. Salibekyan S. M., Belousov, A. Yu., Graph Database Implemented by Object-Attribute Approach. Ob'ektnye sistemy – 2014 (zimnjaja sessija): materialy IX Mezhdunarodnoj nauchno-prakticheskoj konferencii, Rostov-na-Donu, 10-12 dekabrja 2014 g. [Object Systems – 2014 (Winter session): Proceedings of IX International Theoretical and Practical Conference, Rostov-on-Don, 10-12 December, 2014], pp. 70-75 (In Russian). Available at: http://objectsystems.ru/files/2014WS/Object_Systems_2014_Winter_session_Proceedings.pdf

Модель разграничения прав доступа для объектно-ориентированных и объектно-атрибутных приложений

[1] *П.П. Олейник <xsl@list.ru>*

[2] *С.М. Салибекян <ssalibekyan@hse.ru>*

[1] *Шахтинский институт (филиал) Южно-Российского государственного политехнического университета (НПИ) им. М.И. Платова,*
346500, Россия, Ростовская обл., Шахты, пл. Ленина, 1.

[2] *Национальный исследовательский университет «Высшая школа экономики», Московский институт электроники и математики,*
101000, Россия, г. Москва, ул. Мясницкая, д. 20.

Аннотация. В статье приводится описание двух методик разграничения прав доступа, основанных ролевом подходе (RBAC) и применении таблиц/списков прав доступа. Вначале приводится обзор современных подходов к организации безопасности и разграничения прав доступа пользователей в приложениях различной архитектуры. Далее приводится описание двух методик защиты информации. Первая разработана для защиты объектно-ориентированных приложений, вторая приложений объектно-атрибутных, применяемых для управления сетевыми базами данных и базами знаний. Далее внимание уделяется первой авторской методике, основанной на описании прав доступа для классов, атрибутов классов и объектов, удовлетворяющих определенному критерию. Подход, разработанный первым автором, реализован с помощью иерархии классов, состав и структура которых детально описана в работе. Также приводятся примеры конкретных информационных систем, разработанных первым автором: информационная система проведения научных конференций, используемая многократно при проведении конференции «Объектные системы» (objectsystems.ru), а также информационная система салона красоты. Далее приводится описание второй методики, потребовавшей разработки новых подходов к организации защиты информации. Вторая методика, разработанная вторым автором, полностью дублирует функциональность первой. В частности, она обеспечивает копирование прав доступа при копировании части сетевой структуры данных, подобно тому, как в объектно-ориентированной парадигме происходит передача свойств родителя к потомку класса; в статье приводится подробное описание такого механизма. Для управления правами доступа в такой методике применяется специальное виртуальное устройство, а информация о правах доступа привязывается узлу сети (графа), если необходимо ограничить доступ к нему.

Ключевые слова: защита информационной системы, объектно-ориентированные приложения, объектно-ориентированная метамодель, модель разграничения прав, объектно-атрибутный подход.

DOI: 10.15514/ISPRAS-2016-28(3)-3

Для цитирования: Олейник П.П, Салибекян С.М. Модель разграничения прав доступа для объектно-ориентированных и объектно-атрибутных приложений. Труды ИСП РАН, том 28, вып. 3, 2016 г. стр. 35-50 (на английском). DOI: 10.15514/ISPRAS-2016-28(3)-3

Список литературы

[1]. Nagaratnam N., Nadalin A., Hondo M., McIntosh M., Austel P. Business-driven application security: from modeling to managing secure applications. IBM Systems Journal, vol. 44, issue 4, 2005, pp. 847-867

[2]. Xiao L., Peet A., Lewis P., Dashmapatra S., Saez C., Croitoru M., Vicente J., Gonzalez-Velez H., Lluch i Ariet M. An Adaptive Security Model for Multi-agent Systems and Application to a Clinical Trials Environment. 31st Annual International Computer Software and Applications Conference, COMPSAC 2007, 24-27 July 2007, Beijing, China, 2007, pp. 261-268

[3]. Fengyu Zhao, Xin Peng, Wenyun Zhao. Multi-Tier Security Feature Modeling for Service-Oriented Application Integration. Eighth IEEE/ACIS International Conference

on Computer and Information Science, ICIS 2009, 1-3 June 2009, Shanghai, China, 2009, pp. 1178-1183

[4]. Saleem M.Q., Jaafar J., Hassan M.F. Model Driven Security Framework for Definition of Security Requirements for SOA Based Applications. 2010 International Conference on Computer Applications and Industrial Electronics (ICCAIE), 5-8 Dec. 2010, Kuala Lumpur, 2010, pp. 266-270

[5]. Shiroma Y., Washizaki H., Fukazawa Y., Kubo A., Yoshioka N. Model-Driven Security Patterns Application Based on Dependences among Patterns. ARES '10 International Conference on Availability, Reliability, and Security, 15-18 Feb. 2010, Krakow, Poland, 2010, pp. 555-559

[6]. Salini P., Kanmani S. Application of Model Oriented Security Requirements Engineering Framework for Secure E-Voting. 2012 CSI Sixth International Conference on Software Engineering (CONSEG), 5-7 Sept. 2012, Indore, 2012, pp. 1-6

[7]. Олейник П.П. Представление метамодели объектной системы в реляционной базе данных. Известия высших учебных заведений. Северо-Кавказский регион. Спецвыпуск «Математическое моделирование и компьютерные технологии», стр. 3-8, 2005.

[8]. Oleynik P.P. Implementation of the Hierarchy of Atomic Literal Types in an Object System Based of RDBMS. Programming and Computer Software, vol. 35, no.4, 2009, pp. 235-240.

[9]. Олейник П.П. Иерархия классов метамодели объектной системы. Объектные системы – 2012: материалы VI Международной научно-практической конференции (Ростов-на-Дону, 10-12 мая 2012 г.), стр. 37-40. Доступно по ссылке: http://objectsystems.ru/files/2012/Object_Systems_2012_Proceedings.pdf

[10]. Олейник П.П. Предметно-ориентированное проектирование структуры базы данных в понятиях метамодели объектной системы. Объектные системы – 2014: материалы VIII Международной научно-практической конференции (Ростов-на-Дону, 10-12 мая 2014 г.), стр. 41-46. Доступно по ссылке: http://objectsystems.ru/files/2014/Object_Systems_2014_Proceedings.pdf

[11]. Oleynik P.P. Using metamodel of object system for domain-driven design the database structure. Proceedings of 12th IEEE East-West Design & Test Symposium (EWDTS'2014), Kiev, Ukraine, September 26 – 29, 2014, pp. 79-86. DOI: 10.1109/EWDTS.2014.7027052

[12]. Oleynik P.P. Unified Metamodel of Object System. Объектные системы – 2015: материалы X Международной научно-практической конференции (Ростов-на-Дону, 10-12 мая 2015 г.), стр. 79-85. Доступно по ссылке: http://objectsystems.ru/files/2015/Object_Systems_2015_Proceedings.pdf

[13]. Oleynik P.P. Элементы среды разработки программных комплексов на основе организации метамодели объектной системы. Бизнес-информатика, №4(26), 2013, стр. 69-76. Доступно по ссылке: http://bijournal.hse.ru/data/2014/01/16/1326593606/1BI%204(26)%202013.pdf

[14]. Олейник П. П., Кураков Ю. И. Концепция создания обслуживающей корпоративной информационной системы экономического производственно-энергетического кластера. Прикладная информатика, №6, 2014, стр. 5-23

[15]. Кураков Ю.И., Олейник П.П. Методика реализации унифицированной информационной системы экономического производственно-энергетического кластера угольной промышленности. Горный информационно-аналитический бюллетень, № 5, 2015, стр. 260-273.

[16]. Бородина Н.Е., Олейник П.П., Галиаскаров Э.Г. Опыт выполнения реинжиниринга объектной модели на примере информационной системы каталогизирования научных статей при проведении международных конференций. Объектные системы – 2014 (зимняя сессия): материалы IX Международной научно-практической конференции (Ростов-на-Дону, 10-12 декабря 2014 г.), стр. 17-23 Доступно по ссылке: http://objectsystems.ru/files/2014WS/Object_Systems_2014_Winter_session_Proceedings.pdf

[17]. Козлова К.О., Бородина Н.Е., Галиаскаров Э.Г., Олейник П. П. Предметно-ориентированное проектирование информационной системы салона красоты. Объектные системы – 2015: материалы X Международной научно-практической конференции (Ростов-на-Дону, 10-12 мая 2015 г.), стр. 86-90. Доступно по ссылке: http://objectsystems.ru/files/2015/Object_Systems_2015_Proceedings.pdf

[18]. Олейник П.П., Юзефова С.Ю., Николенко О.И. Опыт проектирования информационной системы для ресторанов быстрого питания. Объектные системы – 2014 (зимняя сессия): материалы IX Международной научно-практической конференции (Ростов-на-Дону, 10-12 декабря 2014 г.), стр. 12-16. Доступно по ссылке: http://objectsystems.ru/files/2014WS/Object_Systems_2014_Winter_session_Proceedings.pdf

[19]. Николенко О.И., Олейник П.П. Прототипирование и реализация графической формы заказа для информационной системы ресторанов быстрого питания. Объектные системы – 2015: материалы X Международной научно-практической конференции (Ростов-на-Дону, 10-12 мая 2015 г.), стр. 68-72. Доступно по ссылке: http://objectsystems.ru/files/2015/Object_Systems_2015_Proceedings.pdf

[20]. Pavel P. Oleynik, Olga I. Nikolenko, Svetlana Yu. Yuzefova. Information System for Fast Food Restaurants. Engineering and Technology, vol. 2, no. 4, 2015, pp. 186-191. Доступно по ссылке: http://article.aascit.org/file/pdf/9020895.pdf

[21]. P. B. Panfilow, S. M. Salibekyan Dataflow Computing and its Impact on Automation Applications. Procedia Engineering, vol. 69, 2014, pp. 1286-1295. doi:10.1016/j.proeng.2014.03.121. Доступно по ссылке: http://www.sciencedirect.com/science/article/pii/S1877705814003671

[22]. Pavel P. Oleynik, Sergey M. Salibekyan. The Approaches to Implementation of Patterns of Static Object Models for Database Applications: Existing Solutions and Unified Testing Model. International Journal of Applied Engineering Research, vol. 10, no. 24, 2015, pp 45513-45516.

[23]. Салибекян С. М., Панфилов П. Б. Объектно-атрибутная архитектура – новый подход к созданию объектных систем. Информационные технологии, № 2, 2012, стр. 8-13.

[24]. Салибекян С.М., Белоусов А.Ю. Сетевая база данных, построенная по объектно-атрибутному принципу. Объектные системы – 2014 (зимняя сессия): материалы IX Международной научно-практической конференции (Ростов-на-Дону, 10-12 декабря 2014 г.), стр. 70-75. Доступно по ссылке: http://objectsystems.ru/files/2014WS/Object_Systems_2014_Winter_session_Proceedings.pdf

Dynamic key generation according to the starting time

A.S. Kiryantsev < reyzor2142@gmail.com>
I.A. Stefanova <aistvt@mail.ru>
Volga Region State University of Telecommunications and Informatics,
77 Moskovskoe sh., Samara, Russia

Abstract. The article analyses the problem of data persistence while transmitting the messages and looks into possible solutions. The central part of the article describes the algorithm of data encryption and digital signature algorithm according to the starting time of the session. In the algorithm the session key is symmetrically generated for each pair of subscribers; further the data are encrypted with this key. In its turn the session key is also encrypted with a public asymmetric key of a recipient and with an asymmetric encryption algorithm. Then the decrypted session key with the decrypted message are sent to the recipient. This client employs the same asymmetric encryption algorithm and his/her secret decryption key to decrypt the asymmetric session key. The decrypted session key is used for decryption of the received message. Thus, every time new symmetric keys are generated according to the starting time of a session, which enables high speed of encryption along with an open to public temporary encryption keys transmitting. Besides, the article contains examples of Diffie-Hellman protocol work and the hash-function algorithm MD5. They are used for encryption of generated temporary keys and for transmitting common private key to both clients. According to the suggested algorithm, the prototype of key and signature generation has been created and probated. The article illustrates the stages of Diffie-Hellman and MD5 protocol work. The prototype was tested with the help of a computer and two phones (2013 and 2015 production years)**.**

Keywords: Diffie-Hellman protocol; MD5-functoin; cryptography; encryption; decryption; digital protection; digital signature; symmetric and asymmetric cryptosystems.

DOI: 10.15514/ISPRAS-2016-28(3)-4

For citation: Kiryantsev A.S., Stefanova I.A. Dynamic key generation according to the starting time. Trudy ISP RAN / Proc. ISP RAS, 2016, vol. 28, issue 3, pp. 51-64. DOI: 10.15514/ISPRAS-2016-28(3)-4.

1. *Introduction*

The necessity of serious approach to information security brings us to the basic concepts of cryptography: digital protection, digital signature and encryption. As you know, cryptography is engaged in the search for solutions to such important

security issues as confidentiality, authentication, integrity and control of participants in the interaction.

Encryption is the process of converting data into a form, which is not possible to read the keys. It uses the encryption – decryption keys. The encryption process of the original message helps to ensure privacy by keeping information secret from someone it is not addressed. A set of conversion algorithms and keys used by these algorithms for encryption, key management system, as well as the original and the encrypted text form a cryptographic system. In turn, cryptosystems ensure the secrecy of transmitted messages as well as their authenticity and a user's authentication. The article offers new ideas for dynamic generation of keys and signatures depending on the starting time of the interaction between two subscribers.

2. *Approaches to the construction of cryptosystems*

There are two methods of cryptographic information processing with the keys – symmetric and asymmetric [1]. A symmetric (private) method implies that the sender and receiver use the same key, which they agree before the interaction for both encryption and decryption. If the key has not been compromised, then decrypt database automatically authenticates the sender, since it is only the sender who has the key, which he/she can use to encrypt information, and it is only the recipient who has the key to decrypt the information.

The symmetric encryption algorithms use keys that are not very long and can quickly encrypt large amounts of data. Symmetric encryption systems have a common drawback – that is the complexity of the keys distribution. When an external party intercepts the key, the system of cryptographic protection will be compromised. When it is necessary to replace a key, it should be sent confidentially to the participants of the encryption. Obviously, this method is not suitable when one needs to establish a secure connection with a large number of Internet subscribers. The main problem of this method is how to generate and securely transmit keys to the participants of the interaction. How is it possible to establish a secure communication channel between the participants of interaction while sending keys through insecure communication channels? The lack of a secure key exchange method limits the expansion of symmetric methods of encryption in the Internet.

This problem is resolved in an asymmetric (public) encryption method. In an asymmetric system, the document is encrypted with one key and decrypted with another one. Each participant of the information transfer generates two random numbers (private and public keys). The public key is transferred through public communication channels to another participant of the encryption, but the private key is kept in secret. The sender encrypts the message with the public key of the recipient, and it is only the private key owner who may decrypt the message. This method is suitable for a wide usage. If each Internet user is assigned to his/her own pair of keys and the public keys are published as the numbers in the phone book, almost all users can exchange encrypted messages with each other.

All asymmetric cryptosystems are the object of direct attacks through the direct key enumeration, and, therefore, they must use much longer keys than those used in symmetric cryptosystems to provide an equivalent level of protection. This immediately affects the calculation resources required for encryption.

There is the necessity to verify that there is no distortion into the information in an e-document. Digital signature is used for this sake. Digital signature in a cryptosystem protects a document from changes or substitution and, thereby, guarantees its validity. It is a line, where the attributes of the document (for example, checksum of a file, etc.) and its contents are encoded, so that any change in the file even with the unchanged signature may be detected. When a document is protected by a digital signature, it verifies the document itself along with the private key of the sender, and the recipient's public key. The owner of a private key is the only one who can sign the document correctly. To verify the digital signature of the document, the recipient uses the sender's public key. No other key pair is suitable for verification. Thus, unlike an ordinary signature, digital signature depends on the document and the sender's public key. Therefore, it is several times safer than an ordinary signature and a seal.

Despite the fact that digital signature certifies the authenticity of the document, it does not protect it from unauthorized reading. Both symmetric and asymmetric encryption systems have their advantages and disadvantages. The shortcomings of symmetric encryption are in the complexity of replacing a compromised key, and the disadvantages of asymmetric encryption are in a relatively low speed of work.

These problems are addressed to the encryption systems that use the combined algorithm, which enables high-speed encryption and sending of the encryption keys through the public channels. In order to avoid low-speed of asymmetric encryption algorithms, a temporary symmetric key is generated for each message. The message is encrypted with a temporary symmetric session key. Then this session key is encrypted with a public asymmetric key of a recipient and an asymmetric encryption algorithm. Due to the fact that a session key is much shorter than a message itself, the time of encryption will be relatively short. After that this encrypted session key is transferred to the recipient along with the encrypted message. The recipient uses the same asymmetric encryption algorithm and his/her private key to decrypt the session key and the received session key is used to decrypt the message.

The mentioned above makes it obvious that combined encryption algorithms currently have a promising line of development in modern cryptosystems.

3. *Algorithm description*

It is time to consider the operation principle of the suggested method to data encryption with the session symmetric key, generated at the moment of interaction between the two subscribers. The session key is encrypted with the exposed asymmetric key of the recipient and Diffie-Hellman’s algorithm [2]. The algorithm allows two sides to get common private key, using the channel that is unprotected

from discreet listening, but protected from the channel substitution. The received key can be used for message exchange through symmetric encryption.

Diffie-Hellman's algorithm uses one-sided function *F(X)* with two attributes:

- there is a polynomial algorithm of values *F(X)*,
- there is not a polynomial algorithm of inverted function *F(X)*.

To put simply, this function doesn't include decryption of the encrypted text.

The function with a secret is the function *Fk*; it depends on *k* and has the following properties: there is a polynomial algorithm of calculation *Fk(X)* value for any *k* and *X*, and there is not a polynomial algorithm of the inverted *Fk* for unknown *k*; but there is a polynomial algorithm of inverted *Fk* for the known *k* parameter.

Fig. 1 presents encryption's block diagram according to the Diffie-Hellman's algorithm.

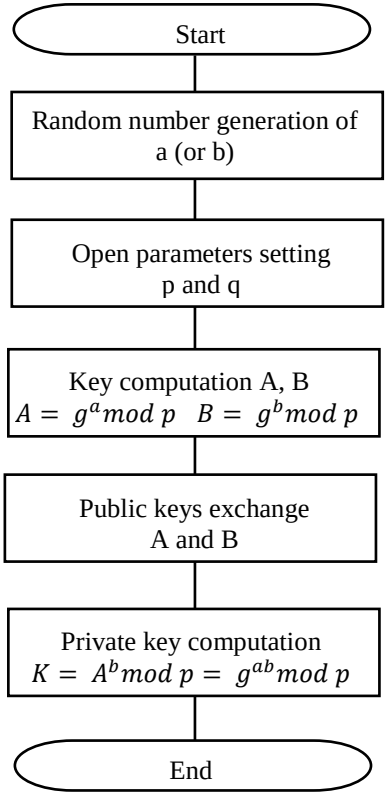


Fig. 1. Block diagram of Diffie-Hellman algorithm.

The algorithm operation is presented in the following example. Andrew defines variables g and p which are large numbers. And he also conceives his private number *a* and calculates the value *A* using the formula

$$A = g^a mod\ p \quad (1)$$

Then he transmits it to Natasha along with the conceived values of g and p. Natasha conceives her private number b. Through the same formula as Andrew does, she calculates her public number

$$B = g^b mod\ p \quad (2)$$

and sends to Andrew. It is possible that the malicious user can get both values, but he will not modify them, as he is unable to interfere in broadcasting process.

At the second stage Natasha calculates the value of *K* having number *B* and the received number *A*:

$$K = A^b mod\ p = g^{ab} mod\ p\ , \quad (3)$$

That is the key for encryption. Then, Andrew calculates his key using number *B* received from Natasha and his calculated number *A*

$$K = B^a mod\ p = g^{ab} mod\ p. \quad (4)$$

You can see in examples (3) and (4) that Andrew gets the same number k, as Natasha. As a result, there is a root key that will be used in generating temporary key and message's signature in the future.

If the root key is used as a private key, a malefactor will be forced to meet with a practically undecidable (for a reasonable period of time) problem of calculating the number $g^{ab} mod\ p$ having numbers $A = g^a mod\ p$ and $B = g^b mod\ p$, intercepted in the public channel if p, a and b are large enough numbers.

Now it is time to explain the process of temporary key generation. It follows the same HMAC (hash-based message authentication code) algorithm [3] and its standard RFC2104. According to them, information integrity is verified with private key. This standard allows to ensure that transmittable or stored at unreliable environment data were not change by unknown persons.

The HMAC algorithm contains the standard, describing the process of data exchange, the process of data integrity verification with the help of private key and hash-function. Depending on the hash-function, we can distinguish HMAC-MD5, HMDC-SH1 etc.

In the article, the hash-function is generated from the root key by the suggested algorithm, for example: md5 (rootKey + Time). The function md5 is a modification of hash-function MD5. At the generation of hash-function, the time, particularly its second value, will be rounded. As it is known, time is presented in the format HH:MM:SS and rounding happens in the last format's unit. If there are more than 30 sec. in the value of starting time SS, then they round upward, if there are less than 30 sec., then they round downward. The message will be encrypted exactly with this key, and also through this algorithm one can generate digital signature of message to verify the message. As a result, we get a resistant system of dynamic keys for messages encryption and signature, where participants do not need to exchange some data for generation and root key generally.

A generalized algorithm of messages encryption in cryptosystem with the key and signature generation is presented on fig. 2.

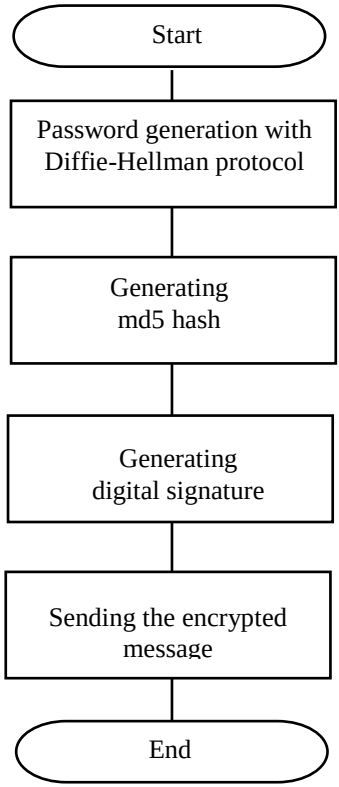


Fig 2. Block diagram of encryption by the key and signature generation algorithm.

4. Working prototype

Web technologies and JavaScript language were chosen for prototype realization. Due to it, the program will become a cross-platform and can be loaded everywhere, when there is a support of JavaScript specification (EcmaScript 5) and HTML 4 support. The JavaScript language was not chosen by chance, as at this moment it is the only "native" language for browser and it is supported by all browsers on default.

Below we can consider fragments of prototype code as an example.

Mass Math.random is used for generation of a large number p by Diffie-Hellman algorithm.

This approach is justified by the fact that the JavaScript language cannot work with large numbers (BigInt), as the algorithm requires it.

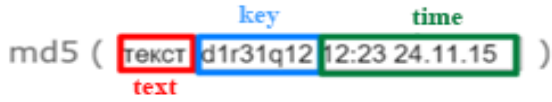
The code generation example of a large number in JavaScript language looks as this:

```
    random(1000000000,9999999999) + " + random(1000000000,9999999999) + "
+   random(1000000000,9999999999) + " + random(1000000000,9999999999) + "
+   random(1000000000,9999999999) + " + random(1000000000,9999999999) + "
+   random(1000000000,9999999999) + " + random(10000000,99999999);
```

Then the code of message's generation to JavaScript language seems:

```
    $scope.getSign = function()
      {
       return md5($scope.msg + $scope.username + bigInt2str(a_sec,
10).toString() + datetime);
      }
```

Function md5(arg) returns the hash line from argument arg. Function bigInt2str is a function that allows to work with large numbers in JavaScript. $scope.username allows to insert a username. In this way we get a unique signature for each user. There is a screenshot of text values' substitution and the result of the performed program:



5c733932c8910c2c6cb1432d1eb6f117

The time test script execution was conducted through the prototype. In this test the following e-devices were used:

1. The computer – INTRL i5 (Windows 10/chrome)
2. The phone – Nexus 5 (android 6.0.1/ chrome)
3. The phone – Samsung galaxy ace (android 4.2.2/ browser).

In table 1 the results of the algorithm individual steps are provided. The steps are applied in different application. In fig. 3 there is a diagram that visualizes experiment results. From the table analysis it is obvious that the algorithm works very fast on the mobile phones.

Hash-function algorithm MD5 is not selected occasionally, it is the fastest, the most common one. It has the simplest hashing algorithm that may be used for signature generation. Besides MD5 possesses a very interesting property. For instance, if at least one byte in a line is changed, the view of the resulting hash line will change dramatically.

Table 1. Time of algorithm application in different devices at different stages (msec).

devices / algorithm	Intel i5 (Windows/chrome)	Nexus 5 (android 6.0.1/ chrome)	Samsung galaxy ace (android 4.2.2/ browser)
Diffie-Hellman generation	20,915	166,706	220,53
MD5 generation	0,81	3,315	6,21
Sign generation	0,27	0,48	0,72
Total time	22,883	170,89	229,416

Fig 3. Histogram of algorithm performance time by different e-devices.

The logic of the encryption algorithm can be considered in five steps. After the data are received there is the process of preparing the data flow to the calculations.

Step 1. First, the flow line requires alignment for hashing. At the end of the stream one on-bit and the necessary number of off bits are registered. After the input data alignment, the length of the stream should be equal to $512 \cdot N + 448$.

Step 2. At the end of the message, one should add 64-bit result for alignment. There are 4 low-order bits that are put first, then high-order bits follow. If the stream length exceeds $2^{64} - 1$, only low-order bits are written down. After that, the stream length becomes 512-fold. The calculations are made with data flow presented as an array of 512-bit words.

Step 3. Then it is necessary to initialize 4 32-bit variables (A, B, C, D) and to set their initial values with hex numbers: "low-order byte comes first". For example,

A = 01 23 45 67; // 67452301h

B = 89 AB CD EF; // EFCDAB89h

C = FE DC BA 98; // 98BADCFEh

D = 76 54 32 10. // 10325476h

The results of intermediate calculations will be stored in these variables. Then it is time to initialize constants and functions required in further calculations.

Four laps will require 4 functions with the logical operators XOR (⊕), AND (∧), OR (∨), NOT (¬):

$$FunF(X,Y,Z) = (X \wedge Y) \vee (\neg X \wedge Z),$$

$$FunG(X,Y,Z) = (X \wedge Z) \vee (\neg Z \wedge \mathrm{Y}),$$

$$FunH(X,Y,Z) = (X \oplus \mathrm{Y} \oplus Z),$$

$$FunI(X,Y,Z) = Y \oplus (\neg Z \vee \mathrm{X}).$$

The 64-element table of invariables is structured as follows:

$$T[n] = int(2^{32} \cdot |\sin(n)|)$$

Each 512-bit block of the flow passes through 4 stages of calculation, 16 laps each. For this the block is presented as an array X of 16 32- bit words. All the laps are of the same type, but they differ in the rotate shift by *s* bits of a 32-bit argument. The number s is defined for each lap.

Step 4. Steps in loop calculations. Base n element into the block from an array of 512-bit blocks. The values A, B, C, D, remain after operations with the previous blocks (or their values in case the array goes first).

AA = A

BB = B

CC = C

DD = D

Sum the values with the result of the previous loop:

A = AA + A

B = BB + B

C = CC + C

D = DD + d

After the loop ends, check if there are any blocks for calculations left. If there are some, go to the next array element (n+1) and the loop repeats.

Step 5. The result of the hash-function calculation is formed in ABCD buffer. If the result starts with the low-order byte A, one gets MD-5 hash.

Fig. 4 presents a screenshot of md5 hash function working prototype in the CRYP2CHAT app [4]. It resorts to a modified MD5 hash function.

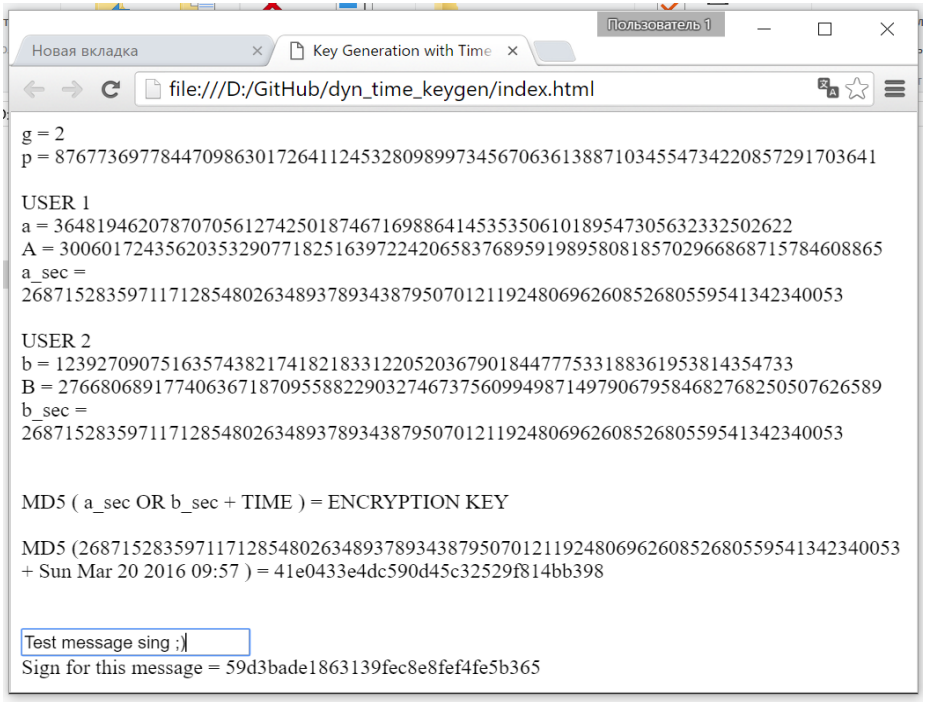


Fig 4. Prototype of Application work.

5. *Evaluation of algorithm effectiveness*

The cryptographic strength of the proposed algorithm for key and signatures generation depends on the encryption method that combines the algorithms of symmetric and asymmetric encryption.

The cryptographic strength is a quantitative characteristic of encryption algorithms – intrusion into a particular algorithm requires a certain number of resources. This is the amount of information and time required to perform the attack, as well as the memory required to store information used in the attack.

An attacking encryption algorithm typically aims at solving the following tasks:

- to get public text version from the encrypted one,
- to calculate the encryption key.

The second task is usually more challenging than the first one. However, having the encryption key the cryptanalyst can later decrypt all the data encrypted with a key.

The algorithm is considered to be secure, if a successful attack at it requires from an attacker unattainable calculating resources in practice, or open intercepted and encrypted messages, or if decryption is so time-consuming that currently protected information would lose its relevance. In most cases, the cryptographic strength

cannot be mathematically proven, you can only prove the vulnerability of the algorithm or to calculate the time required to find the key. For this sake, one should take into consideration the difficulty of a given mathematical problem that serves as the basis for the encryption algorithm.

To estimate the time of the password configuration to gain an unauthorized access to the channel of two subscribers, we have the equation [5]:

$$t = \frac{N^0+N^1+N^2+N^L}{V} \quad (5)$$

It estimates time in the worst case. Here t is the time required for the guaranteed password configuration, V is the number of combinations per second in brute search, N is the number of characters in the configurated password, L is the length of the password.

In case with the md5 algorithm, the number of characters is 36. This number includes the 26 symbols-letters in the Latin alphabet (a...z) and 10 symbols of Arabic numerals (0..9). The number of symbols in the secure key for encryption or signing is 32. To calculate speed of the brute symbol search, we'll take an Intel i7 and a video card Radeon HD5850 1024 MB. Their power equals to 65 000 passwords per second, calculated empirically.

As a result of substitution of values in (5) the estimated time will be:

$$t = \frac{36^0 + 36^1 + 36^2 + 36^{32}}{6500} = 9.745\text{x}10^{44}\text{c.}$$

Converting the seconds into a larger value, we get the result 3.09×10^{37} years.

Conclusion: this algorithm can be considered secure from attack and the encryption key calculation, as the time for the key search outweighs the actual time of work with data.

In sources [5, 6] an algorythm of dynamic key generation is offered. It is presented as a self-authenticated method with timestamp. In the patent the author employs asymmetric encryption-decryption algorithm. In contrast in this article the described algorithm is symmetric. This helps exclude sending and receiving any key, which increases security of data transmission. Moreover, Google team uses slightly similar algorithm of key generation. However, its development group employs another hash function that is not connected with encryption. Additionally, password configuration is a part of the algorithm that we provide.

6. *Conclusion*

The algorithm for temporary keys and signatures generation can be used to teach students the basics of cryptography, and used in real projects. Coupled with a VPN or TOR networks it becomes more secure due to the new encryption level [7].

References

[1]. Mikhail Adamenko. The basics of classical cryptology. The secrets of ciphers and codes. DMK Press [DMK Publishing], 256 p., 2014 (in Russian).
[2]. Diffie, W. and Hellman, M. E. New directions in cryptography. IEEE Transactions on Information Theory, vol. 22, issue 6, 1976, pp. 644-654.
[3]. Maurer U.M, Wolf S. The Diffie-Hellman Protocol. Retrieved. Designs, Codes and Cryptography. Special Issue: Public Key Cryptography, № 19, 2000, pp.147-171.
[4]. The construction of the password generator. Retrieved from www.scritub.com/limba/rusa/194620205.php, 2013-08-02 (accessed February, 2016) (in Russian).
[5]. Self-authenticated method with timestamp. Patent US 20140325225 A1. Retrieved from http://www. google.com/patents/US20140325225 (accessed Oct. 30, 2014).
[6]. SELF-AUTHENTICATED METHOD WITH TIMESTAMP - DIAGRAM, SCHEMATIC, AND IMAGE. Retrieved from http://www.faqs.org/patents/imgfull/20140325225_06 (accessed Oct. 30, 2014 Sheet 5 of 5).
[7]. Kiryantsev A.C., Stefanova I.A. Constructing Private Service with CRYP2CHAT application. Trudy ISP RAN / Proc. ISP RAS], vol. 27, issue 3, 2015, pp. 279-290. DOI: 10.15514/ISPRAS-2015-27(3)-19

Генерация динамических ключей и подписей с зависимостью от времени

А.С. Кирьянцев < reyzor2142@gmail.com>
И.А. Стефанова <aistvt@mail.ru>
Поволжский государственный университет телекоммуникаций и информатики, Самара, Московское шоссе, 77

Аннотация. В статье рассмотрена проблема сохранности переписки при передаче и пути решения этой проблемы. Центральную часть статьи составляет описание алгоритма генерации паролей для шифрования данных и генерации подписей для сообщений с зависимостью от времени начала взаимодействия двух абонентов. В предлагаемом комбинированном алгоритме используется генерация временного симметричного сеансового ключа для каждой пары абонентов с последующим шифрованием этим ключом передаваемого сообщения. В свою очередь сам сеансовый ключ шифруется с помощью открытого асимметричного ключа получателя и асимметричного алгоритма шифрования. Далее зашифрованный сеансовый ключ вместе с зашифрованным сообщением передается получателю, который использует тот же самый асимметричный алгоритм шифрования и свой секретный ключ для расшифровки симметричного сеансового ключа, а полученный сеансовый ключ используется для расшифровки самого принятого сообщения. Таким образом, симметричные ключи генерируются каждый раз новые, в зависимости от времени установки связи между парой абонентов, что позволяет при высокой скорости шифрования использовать открытую пересылку временных ключей шифрования. А для их сохранности уже использовать асимметричные методы шифрования. Кроме того в статье рассмотрены примеры работы протокола Диффи-Хеллмана и алгоритма

хеширования MD5, используемые для шифрования генерируемых временных ключей и позволяющие двум сторонам получить общий секретный ключ. По предложенному алгоритму был создан прототип с реализаций генерации пароля и генерации подписи, который наглядно показывает этапы работы протокола Диффи-Хеллмана и MD5. С помощью прототипа было проведено тестирование на предмет времени исполнения алгоритма на трёх устройствах: на одном стационарном компьютере с видеокартой и двух телефонах (2013 и 2015 годов выпуска).

Ключевые слова: Протокол Диффи-Хеллмана; MD5 функция; криптография; шифрование; дешифрование; цифровая защита; цифровая подпись; симметричные и асимметричные криптосистемы.

DOI: 10.15514/ISPRAS-2016-28(3)-4

Для цитирования: Кирьянцев А.С., Стефанова И.А. Генерация динамических ключей и подписей с зависимостью от времени. Труды ИСП РАН, том 28, вып. 3, 2016 г. стр. 51-64 (на английском). DOI: 10.15514/ISPRAS-2016-28(3)-3.

Список литературы

[1]. Михаил Адаменко. Основы классической криптологии. Секреты шифров и кодов, Издательство ДМК 2014 - 256 с.
[2]. Diffie, W. and Hellman, M. E. New directions in cryptography. IEEE Transactions on Information Theory, vol. 22, issue 6, 1976, pp. 644-654).
[3]. Maurer U.M, Wolf S. The Diffie-Hellman Protocol. Retrieved. Designs, Codes and Cryptography, Special Issue: Public Key Cryptography, № 19, 2000, p.147-171.
[4]. Как устроен генератор паролей?: www.scritub.com/limba/rusa/194620205.php, 2013-08-02 (доступ февраль, 2016).
[5]. Self-authenticated method with timestamp. Patent US 20140325225 A1. Retrieved from http://www. google.com/patents/US20140325225 (accessed Oct. 30, 2014)
[6]. Self-authenticated method with timestamp - diagram, schematic, and image 06. Retrieved from http://www.faqs.org/patents/imgfull/20140325225_06 (accessed Oct. 30, 2014 Sheet 5 of 5)
[7]. Kiryantsev A.C., Stefanova I.A. Constructing Private Service with CRYP2CHAT application. Trudy ISP RAN / Proc. ISP RAS], vol. 27, issue 3, 2015, pp. 279-290. DOI: 10.15514/ISPRAS-2015-27(3)-19.

Automatic Code Generation from Nested Petri nets to Event-based Systems on the Telegram Platform

D.I. Samokhvalov <disamokhvalov@edu.hse.ru>
L.W. Dworzanski <leo@mathtech.ru>
National Research University Higher School of Economics,
Myasnitskaya st., 20, Moscow, 101000, Russia

Abstract. Nested Petri net formalisms is an extension of coloured Petri net formalism that uses Petri Nets as tokens. The formalism allows creating comprehensive models of multi-agent systems, simulating, verifying and analyzing them in a formal and rigorous way. Multi-agent systems are found in many different fields — from safety critical systems to everyday networks of personal computational devices; and, their presence in the real world in increasing with the increasing number of mobile computational devices. While several methods and tools were developed for modelling and analysis of NP-nets models, the synthesis part of multi-agent systems development via NP-nets is still under active development. The widely used method of automatic generation of target system code from designed and verified formal models ensures obtaining correct systems from correct models. In this paper, we demonstrate how Nested Petri net formalism can be applied to model search-and-rescue coordination systems and automatically generate implementation in the form of the executable code for event-driven systems based on the Telegram platform. We augment the NP-nets models with Action Language annotation, which enables us to link transition firings on the model level to Telegram Bot API calls on the implementation level. The suggested approach is illustrated by the example annotated model of a search and rescue coordination system.

Keywords: nested petri nets; telegram bot api; action language; event-based systems; code-generation.

DOI: 10.15514/ISPRAS-2016-28(3)-5

For citation: Samokhvalov D.I., Dworzanski L.W. Automatic Code Generation from Nested Petri nets to Event-based Systems on the Telegram Platform. *Trudy ISP RAN / Proc. RAS*, vol. 28, issue 3, 2016. pp. 65-84. DOI: 10.15514/ISPRAS-2016-28(3)-5

1. Introduction

Messengers have become the integral part of our life in the recent years; and, almost all the people who have Whatsapp, Viber or Telegram installed on their mobile

devices use them in everyday life. That is all because of hands-on approach in terms of receiving and sending information. Telegram Bot API (TBA)[1] appeared not so long time ago has made a breakthrough in the messengers evolution; and, many IT and business *experts* see the great potential in appliance of the tool for both business and computer science domains.

The variety of TBA usage shows the great diversity of different applied domains starting with service bots, which are designed in order to meet customers requirements, ending with Artificial Intelligence bots (e.g. YandexBot), which can answer different kinds of queries and even strike up and sustain a coherent conversation. The one sphere where TBA could be applied in — people coordinating in different types of special operations. These operations turn out to be extremely difficult to plan and support when it comes to coordination of big squads; especially, in the state of emergency cases. A thorough planning of search and rescue or military operations is rather struggling to deal with, because of the lack of time to create a detailed schedule of part-taking for each agent and deprecated methods for sending and receiving notifications from the agents who are involved in such operations. TBA provides a great opportunity for that purpose because it is extremely easy to use when the bot logic is designed according to a consecutive and well-structured scheme. However, it is not easy to create a coherent TBA logic, because it requires programming skills and is time-consuming. As the time factor plays a crucial role, this makes such system much less attractive and unsuitable in the fast changing context of emergency and rescue operations. Nested Petri Nets (NP-nets) are a well-known formalism which provides an approach for modelling multi-agent systems [2], [3], [4], [5]. NP-nets are generally used to describe the complex processes with dynamic hierarchical structure. NP-nets are convenient for specification of that kind of processes because of the visible and coherent structure [6]. A number of methods for the analysis and verification of NP-nets were developed [7], [8], [9]. However the practical application is impeded by the necessity of manual implementation of the constructed model. Even if the model correctness is verified, code defects can be introduced on the error-prone implementation phase of software construction process. The reasons for such defects: different understanding of the model by a software architect and software developers; the complex behaviour of multi-agent systems with dynamic structure; the distributed systems testing and debugging problems. The alternative to manual coding is automatic code generation from a model to the executable implementation of the modelled system. Automatic generation provide considerable saving of the project resources, reproducible quality of the generated code, better support for round-trip development by regenerating code after model changes. The approach does not guarantee zero-defect implementation, but, after long term usage, a code generation system becomes reliable and allows to obtain code with reproducible quality.

The goal of the project is to develop a code generation system which allows to automatically construct multi-agent systems on the Telegram platform from NP-nets

models. The generated software is designed according to the event-base paradigm and consists of a complex Telegram Bot and mobile Telegram applications. The main purpose of the Telegram bot is to coordinate and communicate with the agents according to the original NP-net model.

The section 2 contains basic notation and definitions. In the section 3, a motivating example of Search and Rescue coordination system modelled with the NP-nets formalism is given. In the section 4, we provide the architecture and technical details on the implementation of the automatic code generation. The section 5 contains the suggested Action Language description. In the section 6, we discuss the application of the suggested technology to the motivating example. The section 7 concerns the related work, the previous studies on NP-nets translations, and further directions.

2. Preliminaries

At first, we provide the classical definition of a Petri Net. A *Petri net (P/T-net)* is a 4-tuple (P, T, F, W) where

- P and T are disjoint finite sets of *places* and *transitions*, respectively;
- $F \subseteq (P \times T) \cup (T \times P)$ is a set of *arcs*;
- $W: F \to N \backslash \{0\}$ – an *arc multiplicity function*, that is, a function which assigns every arc a positive integer called an *arc multiplicity*.

A *marking* of a Petri net (P, T, F, W) is a multiset over P, i.e. a mapping $M: P \to N$. By $M(N)$ we denote the set of all markings of the P/T-net N.

We say that a transition t in the P/T-net $N = (P, T, F, W)$ is *active* in a marking M if for every $p \in \{p | (p,t) \in F\}: M(p) \geq W(p,t)$. An active transition may *fire*, resulting in a marking M'; such that, for all $p \in P: M'(p) = M(p) - W(p,t)$ if $p \in \{p | (p,t) \in F\}$, and $M'(p) = M(p)$ otherwise.

For simplicity, we consider here only two-level NP-nets, where net tokens are classical Petri nets.

A *nested Petri net* is a tuple $NPN = (Atom, Expr, Lab, SN, (EN_1, \dots, EN_k))$, where

- $Atom = Var \cup Con$ – a set of atoms;
- Lab is a set of transition labels;
- $(EN_1, \dots, EN_k)$, where $k \geq 1$ – a finite collection of P/T-nets, called element nets;
- $SN = (P_{SN}, T_{SN}, F_{SN}, \nu, W, \Lambda)$ is a high-level Petri net where
  - P_{SN} and T_{SN} are disjoint finite sets of *system places* and *system transitions* respectively;
  - $F_{SN} \subseteq (P_{SN} \times T_{SN}) \cup (T_{SN} \times P_{SN})$ is the set of *system arcs*;
  - $\nu: P_{SN} \to \{EN_1, \dots, EN_k\}$ is a *place typing* function;
  - $W: F_{SN} \to Expr$ is an *arc labelling* function, where $Expr$ is the

arc expression language;

- $\Lambda: T_{SN} \to Lab \cup \{\tau\}$ is a *transition labelling* function, τ is the special "silent" label.

Let Con be a set of *constants* interpreted over $A = A_{net} \cup \{°\}$; and, $A_{net} = \{(EN, m) | \exists i = 1, \dots, k: EN = EN_i, m \in (EN_i)\}$ is a set of marked element nets. Let Var be a set of *variables*. Then the expressions of $Expr$ are multisets over $Var \cup Con$. The arc labelling function W is restricted such that: constants or multiple instances of the same variable are not allowed in input arc expressions of transitions; constants and variables in the output arc expressions correspond to the types of output places; and, each variable in an output arc expression of a transition occurs in one of the input arc expressions of the transition.

A marking M of an NP-net $NPN = (Atom, Expr, Lab, SN, (EN_1, \dots, EN_k))$ is a function mapping each $p \in P_{SN}$ to a multiset $M(p)$ over A. The set of all markings of an NP-net NPN is denoted by $M(NPN)$ Let $Vars(e)$ denote a set of variables in an expression $e \in Expr$ For each $t \in T_{SN}$ we define $W(t) = \{W(x, y) | (x, y) \in F_{SN} \wedge (x = t \vee y = t)\}$ – all expressions labelling arcs incident to t. A *binding* b of a transition t is a function $b: Vars(W(t)) \to A$, mapping every variable in the t-incident arcs expressions to a token. We say that a transition t is *active* in a binding b iff: $\forall p \in \{p | (p, t) \in F_{SN}\}: b(W(p, t)) \subseteq M(p)$. An active transition t may *fire* yielding a new marking $M'(p) = M(p) - b\big(W(t, p)\big) + b(W(t, p))$ for each $p \in P_{SN}$ (denoted as $M \xrightarrow{t[b]} M'$).

A behaviour of an NP-net consists of three kinds of steps. A *system-autonomous step* is the firing of a transition, labelled with τ, in the system net without changing the internal markings of the involved tokens. An element-autonomous step is a transition firing in one of the element nets according to the standard firing rules for P/T-nets. An autonomous step in a net token changes only this token inner marking. An autonomous step in a system net can move, copy, generate, or remove tokens involved in the step, but doesn't change their inner markings.

A *(vertical) synchronization step* is a simultaneous firing of a transition labelled with some $\lambda \in Lab$ in a system net with firings of transitions labelled with the same λ in all consumed net tokens involved in the system net transition firing. For further details see [5]. Note, however, that here we consider a typed variant of NP-nets, when a type of an element net is instantiated to each place.

3. Motivating example

Search and rescue operations is what happens all over the world; they require the well-trained and skilled employees, well-structured planning, and knowledgeable human management. There were 2447 emergency callouts registered in Russia throughout 2005–2014 [10], and about 100 times more in USA [11]. Earthquakes,

water floods, and hurricanes hit the earth rarely than ordinary emergency cases like fires or gas leaks, but they leave whole regions and even countries devastated, thousands of people killed or lost without a trace. Therefore, the crucial goal of rescuers is to treat such cases quickly and cohesively.

In this example, we will explain how a particular search and rescue operation in an earthquake could be handled with a multi-agent model based on the nested Petri net formalism. First, we need to introduce the purposes of the basic components which we will use further to design our search and rescue coordination plan. Our model consists of the following basic components:

- **System net** – the main component of an NP-net which is a high level Petri net. It will be used to define the activity coordination of the agents involved in the operation. The system net will be implemented as a bot on the Telegram platform to receive the notifications from agents and to process them with the Action Language (AL) event handlers assigned to the transitions of the system net;
- **Element net** – represents the activity of a particular agent type that is supposed to be performed by an agent while taking part in the operation. There are two element nets in our example. The first one corresponds to the acting plan for medical workers involved in the operation; the second one will provide the plan for the rescuers participating in the operation.

The system net in Fig.1 represents the main model of our operation. Basically, it reflects the dependence of the agent actions on server responses. In other words, it describes how an operation coordinator interacts with the rescuers and medics and reacts on their signals to the server. The model deals only with those agent requests where coordinators answer is essential for the further operation progress. An actions happen when the particular agent reaches the state and the coordinator response expected is defined with AL code assigned to the system net transition. To understand how the model works, we need to understand how the agents intercommunicate with the operation coordinating server.

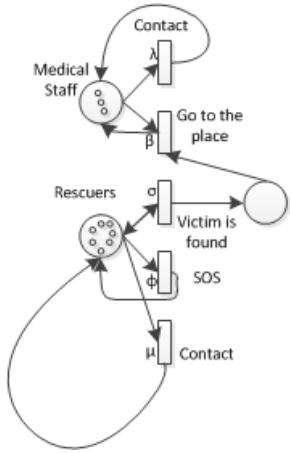


Fig. 1: The search and rescue system net example.

In the initial marking of the system net places “Medical Staff” and “Rescuers”, there are all the agents – rescuers and medical staff respectively. The transitions have the following functions:

- Transitions T0 and T4 **”Contact”** — handle communication between the rescuers and the coordinator;
- Transition T1 **”Go to the place”** — represents the event when a rescuer has found a victim, and a medical agent is supposed to go to the place where the victim is found;
- Transition T2 **”Victim is found”** – represents the event when a rescuer agent has found a victim. It precedes the T1 event, as the medical agent needs to start acting only when the victim is found by the rescuer;
- Transition T3 **”SOS”** – a rescuer has stuck in emergency;

The agents behaviour is determined by two element nets. The medical staff element net is depicted in Fig. 3; and, the rescuer element net — in Fig. 5.1. In the real Search and Rescue operations there are usually more element nets and they are more detailed.

Medical staff element net represents what kind of actions should a medical agent performs while taking part in the operation. At first, the medical agent needs to get the medicine and learn about the operation. He will not be allowed to the next stage of the operation before he performs both of these actions. After doing that, he is supposed to wait until he receives the notification on the accidence. Then he has to send his arrival time, and start making his way to the place where the accidence had happened. The next two steps are to report the victim condition and to transport the victim to the infirmary. The medical agent also may contact coordinator at any time.

Rescuers element net is the model of part-taking for rescuers. Before entering the operation, each agent is required to do the following: get the equipment; obtain the

information about other agents; and, get briefing about the operation. The equipment consists of three parts; and, the agent must equip them all. After entering the operation, the agent has to go to the exploration area. If a victim is found, the agent is supposed to send a photo, a description, and the accurate coordinates of the victim location. If something goes wrong, the agent can just send the location and the coordinator will handle it. Once the exploration is completed, he can receive the coordinates of the new area to explore.

4. Architecture

The way this system is designed relies on three basic components:

- NPNtool (Eclipse plugin) [7] for creating Nested Petri Nets models and linking AL code to the transitions. The main purpose of this tool is to model a system net and element nets which will represent the model of the bot. The AL code will be linked to the transitions and then compiled to the executable file according to the model;
- AL Java-library consists of the AL-compiler and the AL-linker. AL-linker traces the system and element nets, collects all the code from the transitions, and eventually converts in to text files that will be compiled by the AL-compiler. AL-compiler is created with the ANTLR[12] tool. AL-compiler gets an input text file and translates it to the executable artifact that actually represents the Telegram bot;
- Telegram Bot API library that consists of the code for requesting data via HTTP-requests from the Telegram Bot API server.

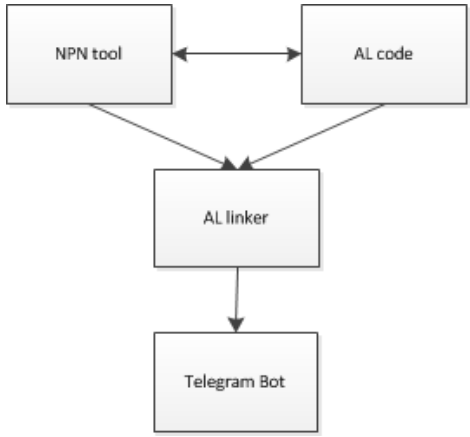


Fig. 2: The code generation scheme

The overall technology chain is as follows. At first, a developer creates and verifies the NP-net model of a system via NPNtool. When the model is constructed, the developer inscribes AL code to the transitions according to the expected logic of the Bot. Then, the developer launches AL-linker which traverses the constructed NP-net collecting the textual representation of the transitions AL code into a text file. After that, AL-compiler reads the artifacts generated by AL-linker and generates a Telegram Bot code. The codegeneration of distributed systems from NP-nets models has been studied in [13]. Once a JAR file is compiled from that code, it could be executed. All the actions of the agents are displayed on the Bot host and could be processed at real-time (saved or directly answered) by the coordinator who ran the Bot.

Telegram bot consists of TBA library and several Java classes. Each Java class corresponds to an element net or a system net and stores a number of methods corresponding to the transitions with AL-code inscribed to them. These methods will use TBA to interchange the information. There is also a class that links all the element and system nets libraries together and proceeds the logic using event-based paradigm and asynchronous requests.

It shall be noticed that the compiled Telegram Bot is a server that communicates with the software clients — the rescue and medical staff software mobile clients. The bot is connected to the Telegram server via the webhook technology; namely, all the requests that agents send to the Telegram server via Telegram mobile applications are redirected to and served on the deployed bot server.

The fragment of code in Table 1 represents the method which corresponds to one of the Medical Staff element net transition:

Table 1. Medical staff victim condition report action.

```
public void taskReportVictimsCondition(String
    mes, String chatId) throws
    TelegramApiException{
       SendMessage message = new SendMessage();
       String[] tasks = {"Report about the
           victim's condition"};
       ReplyKeyboardMarkup replyKeyboardMarkup
           = makeKeyboard(tasks);
       message.setReplayMarkup(
           replyKeyboardMarkup);
       message.setText(mes);
       sendTo(message, chatId);
    }
```

5. Action language

AL compiler has been developed with ANTLR compiler which enables to define a grammar in a ANTLR grammar language and compile it to the Java classes which represent the lexer and the parser of AL. The code generated by ANTLR parses the

AL code and apply specified semantical rules to the constructed syntax tree. To translate the target Java code while traversing the nodes of this tree, the syntax tree visitors were created that generate Java code from the initial AL code.

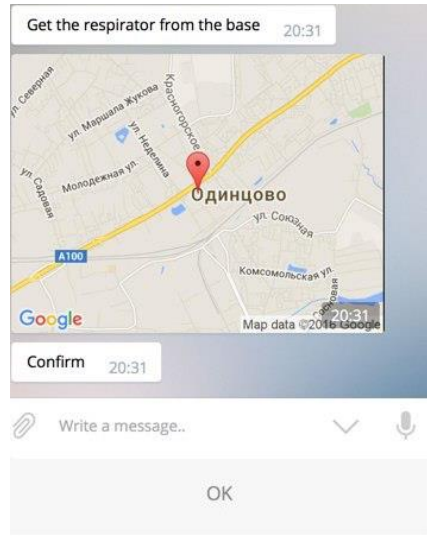


Fig. 3: The agent is confirming the task implementation

Fig. 4: The rescuer agent Telegram mobile client interface.

5.1 AL grammar

Here we provide the formal definition of suggested Action Language. The suggested Action Language is specific to Rescue and Search operations domain, and shall be reconsidered for other applications of the technology.

- < SystemNet> ::= < SP> : < name> — the name of system net
- < ElementNet> ::= < EP> : < name> — the name of element net
- < file> ::= file(< text>) – loads file from file-system
- < initialization > ::= < variable> = < value> — it is possible to assign variables of the types < file>, < float>, < string>
- < sendMessage> ::= sendMessage(< file> | < text> | < variable>) — sends a message from a transition of an element net
- < sendPhoto> ::= sendPhoto(< file> | < variable>) – send a Photo from a transition of an element net
- < sendLocation> ::= sendLocation(longitude : < variable> | < float>, latitude : < variable> | < float> — sends Location
- < sendVideo> ::= sendVideo(< file> | < variable>) – sends Video
- < sendAudio> ::= sendAudio(< file> | < variable>) – sends Audio
- < transition element net> :: = < name> = < text> response: (< sendAudio> | < sendVideo> | < sendLocation> | < sendPhoto> | < sendMessage>)* – this is the structure of the code which should be inscribed on the distinct transition of an element net.
- < connect> ::= connect (< name>.< transition>) – links a transition from an element net to a transition of a system net.
- < display> ::= display() – displays the object received on a transition of a system net
- < save> ::= save(< file> | < text> | < variable>) – saves the object received on the transition of a system net
- < transition system net> ::= < name> = (receive (photo | video | audio | message | location) : (save | display)*)*
- < loop> ::= forall < variable> in < variable botVariable.add(< variable>)

The AL example on Table 1 is from our search-and-rescue system model which was provided in the motivating example. It illustrates what kind of code must be inscribed to the transition of the Medical staff element net (Fig. 5) and the coordinator system net (Fig. 5). We will not provide the code for Rescue element net because it follows the same pattern of coding as for the Medical staff element net.

6. The application of the Telegram bot code generation technology

In this section, we examine the application of the suggested technology to the motivating example provided in the section 3. The main components of the system are modelled with system net and element nets. Then the codegenerator translates NP-nets into Telegram bots components of the target Telegram-based multi-agent system being constructed.

The bot server serves the received requests according to the NP-net system net behaviour and sends the answers to the agents. All the actions, except the actions described on the system net transitions, of the developed Search and Rescue operation are handled by the Bot automatically. However, it is possible to interact ad-hoc during the operation, i.e. if an agent sends any kind of request that was not described by AL, the coordinator will be notified and will be able to answer this request with the standard Telegram client interface. All the event that were described with the system net transitions require the direct interaction of the coordinator. The agent will not be allowed to proceed to the next stage of operation, unless he receives the answer from the coordinator.

As soon as we launch the compiled bot, all the rescuers and medicals that were loaded to the system will receive notifications from the Telegram bot. The concurrent transitions (e.g. Helmet, Respirator, Gloves) from the Rescuer element net allow that all the actions inscribed on them could be executed by agents in any order. An agent will not be allowed to the next stage unless he performed all of them. After performing an action, the agent must confirm that in the mobile client by pressing the «OK» button (Fig. 3). The button appears on the screen when the agent has actions-transitions to fire.

When an agents reaches the “Begin the operation” action, the bot moves to the awaiting state and notifies the coordinator, that the agent has reached the state and waits till the next instructions will be provided. As soon as the coordinator fill the form and submit the answer, the agent will be allowed to move to the next state of his plan. That is due the «Begin the operation» transition is synchronized with the T2 system net transition.

7. Related works and further directions

The codegeneration from models to executable software artifacts has attracted attention when model driven development became industrial popular and valuable approach [14]. The codegeneration from Petri net like models to executable software systems is studied for many formalisms and semi-formal industrial modelling languages like UML[15], [16] and SDL[17]. In [18], [19] the code generation tool for Input-Output Place-Transition Petri Nets was developed. In [20] the application of Sleptsov nets for modelling and implementation of hardware systems is studied. In [21], the technology to construct embedded access control systems from coloured Petri nets models is suggested. The approach to generate

C++ code from SDL models is developed in [8]. The code generation from the UML state machines[15] and sequence [16] diagrams to executable code was studied. These are a lot of studies in the field, so we only cited a few.

The translation from NP-nets to coloured Petri nets was developed in [8]. The translation from NP-nets to PROMELA models to verify the correctness of LTL properties is studied in [22]. The automatic translation from NP-nets models to distributed systems components that preserve liveness, conditional liveness, and safety properties was studied in [13]. In the current work, we adopted the translation scheme developed in the latter work to obtain executable code from the structure of NP-nets models.

The further research concerns theoretical as well as practical aspects of the developed automatic codegeneration system. From the theoretical point of view, it is interesting to study preservation of different behavioural properties by the implemented translation and securing different behavioural consistencies of generated systems and initial models. As the underlaying technologies are too large to conduct exhaustive formal verification, the both dynamic and static behavioural analyses techniques should be applied to study the correctness of the translation. From the practical point of view, there are lot of attractive features that are to be implemented. For example, it is not possible to change the deployed bots at runtime in the tool. However, such function could be of use for long term operations, when new actions should be integrated into an operating Telegram system without recompiling the whole system. The runtime deployment will be considered in the future research. Also, the scalability of generated Telegram systems and possible schemes of agents distribution in the system are the subjects of the further research.

8. Conclusion

The developed technology enables developers to create Telegram Bots according to a visually clear model that could be verified and tested with help of the developed methods [22], [8], [9]. It allows to create distributed event-based Telegram Bots systems that operate on the Telegram platform and the AL language supports all the features provided by Telegram Bot API up to the moment.

The automatic code-generation reduces the risk of introducing defects on the implementation phase of software development process and improves the quality of the resultant code. It not only reduces the cost of software production, but also makes the quality of developed systems more predictable. The suggested technology is demonstrated with the example of a Search and Rescue system.

The authors would like to thank the anonymous referees for valuable and helpful comments.

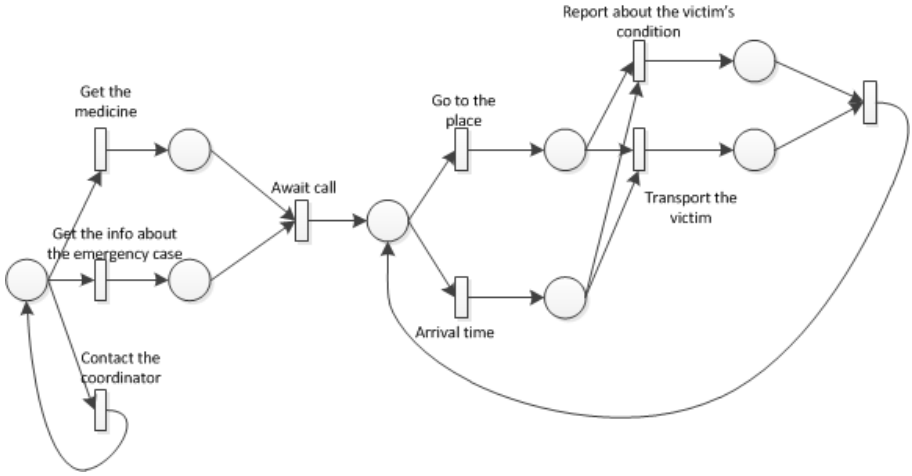


Fig. 5. The Medical Staff element net

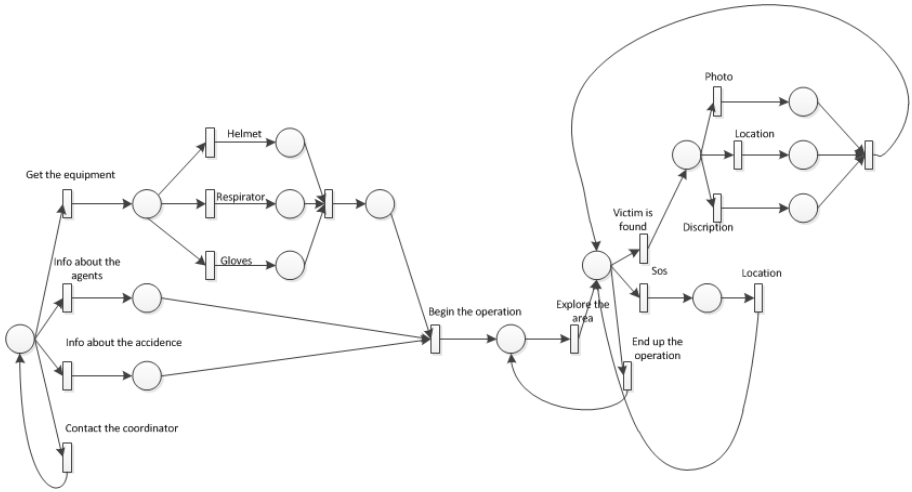


Fig. 6. The Rescuer element net

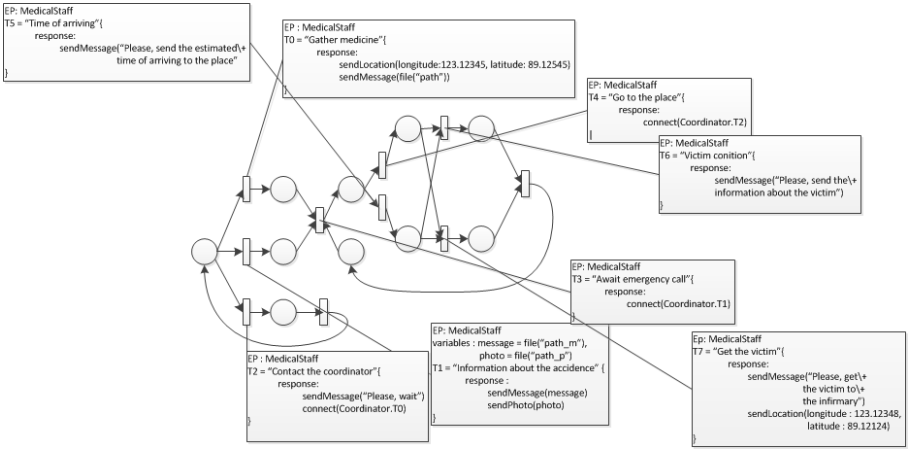


Fig. 7. The element net augmented with code

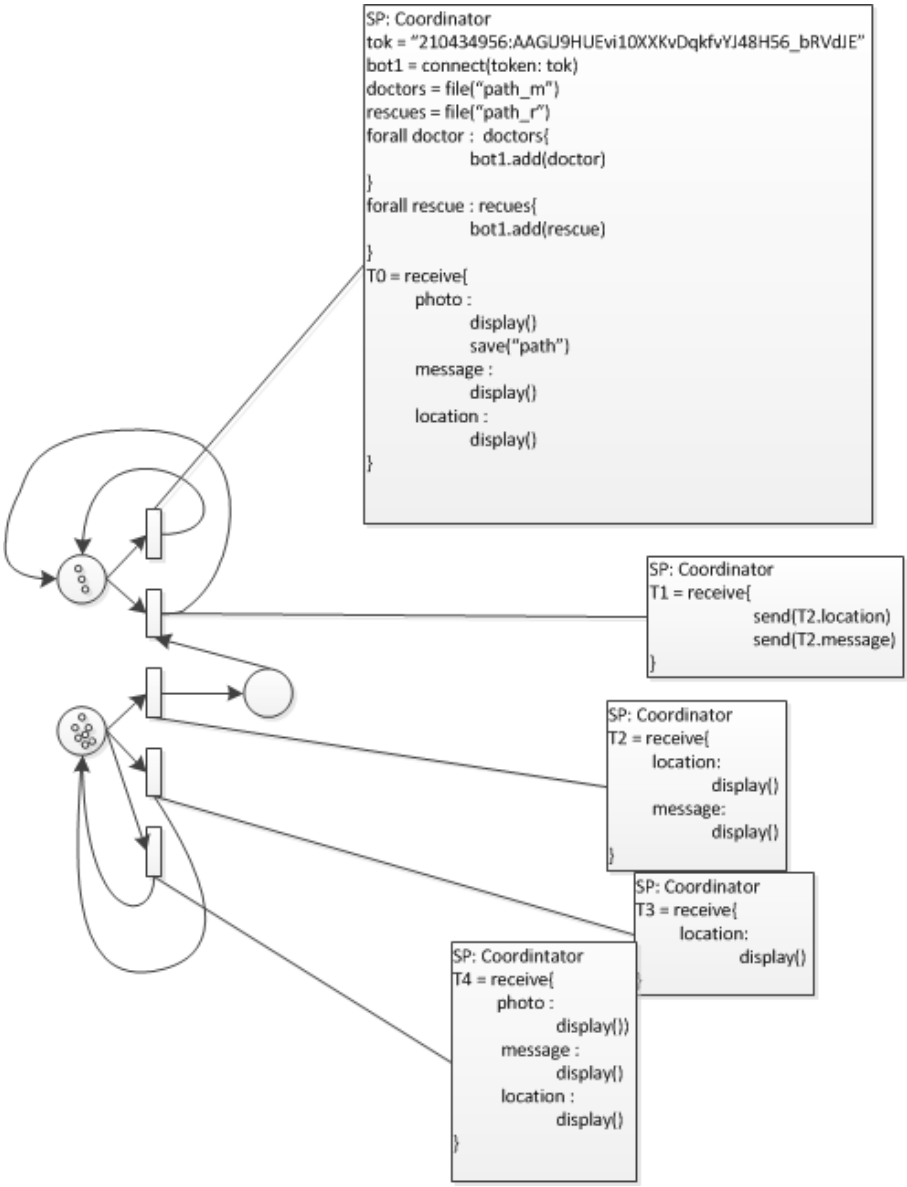


Fig. 8. The system net augmented with code

Acknowledgement

This work is supported by the Basic Research Program at the National Research University Higher School of Economics and Russian Foundation for Basic Research, project No. 16-01-00546.

References

[1]. Telegram Bot API online documentation. [Online]. Available at: https://core.telegram.org/bots/api
[2]. L. Chang, X. He, J. Li, and S. M. Shatz. Applying a Nested Petri Net Modelling Paradigm to Coordination of Sensor Networks with mobile agents. In Proc. of Workshop on Petri Nets and Distributed Systems. Xian, China, 2008, pp. 132–145.
[3]. I. A. Lomazova, “Nested Petri Nets - a Formalism for Specification and Verification of Multi-Agent Distributed Systems,”Fundamenta Informaticae, vol. 43, no. 1, pp. 195–214, 2000.
[4]. Nested Petri nets: Multi-level and Recursive Systems. Fundamenta Informaticae, vol. 47, no. 3-4, pp. 283–293, Oct 2001.
[5]. Nested Petri Nets for Adaptive Process Modelling. In Pillars of Computer Science, ser. Lecture Notes in Computer Science, A. Avron, N. Dershowitz, and A. Rabinovich, Eds. Springer Berlin Heidelberg, 2008, vol. 4800, pp. 460–474.
[6]. K. Hoffmann, H. Ehrig, and T. Mossakowski. High-Level Nets with Nets and Rules as Tokens. In ICATPN, 2005, pp. 268–288.
[7]. D. Frumin and L. Dworzanski. NPNtool: Modelling and Analysis Toolset for Nested Petri Nets. In Proceedings of the 7th Spring/Summer Young Researchers Colloquium on Software Engineering, 2013, pp. 9–14.
[8]. L. Dworzanski and I. A. Lomazova. CPN Tools-Assisted Simulation and Verification of Nested Petri Nets. Automatic Control and Computer Sciences, vol. 47, no. 7, pp. 393–402, 2013.
[9]. L. Dworzanski and I. A. Lomazova. On Compositionality of Boundedness and Liveness for Nested Petri Nets. Fundamenta Informaticae, vol. 120, no. 3-4, pp. 275–293, 2012.
[10]. The Ministry of the Russian Federation for Civil Defence. Emergencies and Elimination of Consequences of Natural Disasters. Emergency Cases Registered in Russia. [Online]. Available at: http://25.mchs.gov.ru/document/2644168
[11]. (2013) United States Coast Guard Search and Rescue Summary Statistics. [Online]. Available at: https://www.uscg.mil/hq/cg5/cg534/SARfactsInfo/SAR%20Sum%20Stats%2064-13.pdf
[12]. T. Parr. The Definitive ANTLR 4 Reference. 2nd ed. Pragmatic Bookshelf, 2013.
[13]. L. Dworzanski and I. A. Lomazova. Automatic Construction of Distributed Component System from Nested Petri Nets. In print, Programmirovanie, vol.6, 2016 (in Russian).
[14]. B. Selic. The Pragmatics of Model-Driven Development. IEEE Software, vol. 20, no. 5, p. 19, 2003.
[15]. A. Knapp and S. Merz. Model Checking and Code Generation for UML State Machines and Collaborations. Proc. 5th Wsh. Tools for System Design and Verification, pp. 59–64, 2002.
[16]. D. Kundu, D. Samanta, and R. Mall. Automatic Code Generation From Unified Modelling Language Sequence Diagrams. Software, IET, vol. 7,no. 1, pp. 12–28, 2013.

[17]. P. Morozkin, I. Lavrovskaya, V. Olenev, and K. Nedovodeev. Integration of SDL Models into a SystemC Project for Network Simulation. In SDL 2013: Model-Driven Dependability Engineering: 16th International SDL Forum, Montreal, Canada, June 26-28, 2013. Proceedings. Springer Berlin Heidelberg, 2013, pp. 275–290.

[18]. L. Gomes, J. P. Barros, A. Costa, and R. Nunes. The Input-Output Place-Transition Petri Net Class and Associated Tools. In Industrial Informatics, 2007 5th IEEE International Conference on, vol. 1. IEEE, 2007, pp. 509–514.

[19]. R. Campos-Rebelo, F. Pereira, F. Moutinho, and L. Gomes. From IOPT Petri Nets to C: An Automatic Code Generator Tool. In Industrial Informatics (INDIN), 2011 9th IEEE International Conference on.

[20]. D. Zaitsev and J. Jurjens. Programming in the Sleptsov Net Language For Systems Control. Advances in Mechanical Engineering, vol. 8, no. 4, p. 1-11, 2016. DOI: 10.1177/1687814016640159.

[21]. K. H. Mortensen. Automatic Code Generation Method Based on Coloured Petri Net Models Applied on an Access Control System. In Application and Theory of Petri Nets 2000. Springer, 2000, pp. 367–386.

[22]. M. L. F. Venero and F. S. C. da Silva. Model Checking Multi-Level and Recursive Nets. Software & Systems Modeling, pp. 1–28, 2016.

Автоматическая генерация кода по вложенным сетям Петри для систем на основе событий на платформе Telegram

Д.И Самохвалов<disamokhvalov@edu.hse.ru>
Л.В. Дворянский <leo@mathtech.ru>
Национальный исследовательский университет Высшая школа экономики,
ул. Мясницкая., 20, Москва, 101000, РФ.

Аннотация. Вложенные сети Петри – это расширение формализма раскрашенных сетей Петри, которые используют сети Петри в качестве фишек. Данный формализм позволяет создавать подробные модели мультиагентных систем, осуществлять имитационное моделирование, верифицировать и анализировать их свойства на формальном и строгом уровне. Мультиагентные системы находят применение во многих областях – начиная системами, для которых безопасность играет критическую роль, заканчивая повседневными системами, работающими на персональных вычислительных устройствах. Число таких систем в современном мире растет вместе с увеличивающимся числом мобильных вычислительных устройств. На данный момент разработаны инструменты и методы моделирования и анализа вложенных сетей Петри, но синтез мультиагентных систем по моделям вложенных сетей Петри еще недостаточно исследован и находится в стадии активного изучения. Метод автоматической генерация исполняемого кода целевой системы по спроектированной и верифицированной модели вложенной сети Петри обеспечивает получение корректных системы из корректных спецификаций на языке вложенных сетей Петри. В данной работе, демонстрируется применение формализма вложенных сетей Петри для построения модели системы управления поисковыми и спасательными операциями и

автоматической генерации реализации в виде исполняемого кода событийно-управляемых систем основанных на платформе Telegram. Мы добавляем возможность аннотировать модели вложенных сетей Петри с помощью Action Language, который позволяет связывать срабатывания переходов на модельном уровне с вызовами Telegram Bot API на уровне реализации. Предложенный подход продемонстрирован на примере аннотированной модели системы координирования спасательной операции.

Ключевые слова: вложенные сети Петри;, telegram bot api; язык действий; событийно-управляемые системы; кодогенерация.

DOI: 10.15514/ISPRAS-2016-28(3)-5

Для цитирования: Самохвалов Д.И., Дворянский Л.В. Автоматическая генерация кода по вложенным сетям Петри для систем на основе событий на платформе Telegram. *Труды ИСП РАН,* том 28, вып. 3, 2016. стр. 65-84 (на английском). DOI: 10.15514/ISPRAS-2016-28(3)-5

Список литературы

[1]. Telegram Bot API online documentation. [Online]. Доступно по ссылке: https://core.telegram.org/bots/api
[2]. L. Chang, X. He, J. Li, and S. M. Shatz. Applying a Nested Petri Net Modelling Paradigm to Coordination of Sensor Networks with mobile agents. In Proc. of Workshop on Petri Nets and Distributed Systems. Xian, China, 2008, pp. 132–145.
[3]. I. A. Lomazova, “Nested Petri Nets - a Formalism for Specification and Verification of Multi-Agent Distributed Systems,”Fundamenta Informaticae, vol. 43, no. 1, pp. 195–214, 2000.
[4]. Nested Petri nets: Multi-level and Recursive Systems. Fundamenta Informaticae, vol. 47, no. 3-4, pp. 283–293, Oct 2001.
[5]. Nested Petri Nets for Adaptive Process Modelling. In Pillars of Computer Science, ser. Lecture Notes in Computer Science, A. Avron, N. Dershowitz, and A. Rabinovich, Eds. Springer Berlin Heidelberg, 2008, vol. 4800, pp. 460–474.
[6]. K. Hoffmann, H. Ehrig, and T. Mossakowski. High-Level Nets with Nets and Rules as Tokens. In ICATPN, 2005, pp. 268–288.
[7]. D. Frumin and L. Dworzanski. NPNtool: Modelling and Analysis Toolset for Nested Petri Nets. In Proceedings of the 7th Spring/Summer Young Researchers Colloquium on Software Engineering, 2013, pp. 9–14.
[8]. L. Dworzanski and I. A. Lomazova. CPN Tools-Assisted Simulation and Verification of Nested Petri Nets. Automatic Control and Computer Sciences, vol. 47, no. 7, pp. 393–402, 2013.
[9]. On Compositionality of Boundedness and Liveness for Nested Petri Nets. Fundamenta Informaticae, vol. 120, no. 3-4, pp. 275–293, 2012.
[10]. The Ministry of the Russian Federation for Civil Defence. Emergencies and Elimination of Consequences of Natural Disasters. Emergency Cases Registered in Russia. [Online]. Доступно по ссылке: http://25.mchs.gov.ru/document/2644168
[11]. (2013) United States Coast Guard Search and Rescue Summary Statistics. [Online]. Доступно по ссылке:: https://www.uscg.mil/hq/cg5/cg534/SARfactsInfo/SAR%20Sum%20Stats%2064-13.pdf

[12]. T. Parr. The Definitive ANTLR 4 Reference. 2nd ed. Pragmatic Bookshelf, 2013.
[13]. Дворянский Л.В., Ломазова И.А. Автоматическое построение распределенной компонентной системы по вложенной сети Петри. В печати: Программирование, no. 6, 2016 (in Russian).
[14]. B. Selic. The Pragmatics of Model-Driven Development. IEEE Software, vol. 20, no. 5, p. 19, 2003.
[15]. A. Knapp and S. Merz. Model Checking and Code Generation for UML State Machines and Collaborations. Proc. 5th Wsh. Tools for System Design and Verification, pp. 59–64, 2002.
[16]. D. Kundu, D. Samanta, and R. Mall. Automatic Code Generation From Unified Modelling Language Sequence Diagrams. Software, IET, vol. 7,no. 1, pp. 12–28, 2013.
[17]. P. Morozkin, I. Lavrovskaya, V. Olenev, and K. Nedovodeev. Integration of SDL Models into a SystemC Project for Network Simulation. In SDL 2013: Model-Driven Dependability Engineering: 16th International SDL Forum, Montreal, Canada, June 26-28, 2013. Proceedings. Springer Berlin Heidelberg, 2013, pp. 275–290.
[18]. L. Gomes, J. P. Barros, A. Costa, and R. Nunes. The Input-Output Place-Transition Petri Net Class and Associated Tools. In Industrial Informatics, 2007 5th IEEE International Conference on, vol. 1. IEEE, 2007, pp. 509–514.
[19]. R. Campos-Rebelo, F. Pereira, F. Moutinho, and L. Gomes. From IOPT Petri Nets to C: An Automatic Code Generator Tool. In Industrial Informatics (INDIN), 2011 9th IEEE International Conference on.
[20]. D. Zaitsev and J. Jurjens. Programming in the Sleptsov Net Language For Systems Control. Advances in Mechanical Engineering, vol. 8, no. 4, p. 1-11, 2016. DOI: 10.1177/1687814016640159.
[21]. K. H. Mortensen. Automatic Code Generation Method Based on Coloured Petri Net Models Applied on an Access Control System. In Application and Theory of Petri Nets 2000. Springer, 2000, pp. 367–386.
[22]. M. L. F. Venero and F. S. C. da Silva. Model Checking Multi-Level and Recursive Nets. Software & Systems Modeling, pp. 1–28, 2016.

Mining Hierarchical UML Sequence Diagrams from Event Logs of SOA Systems while Balancing between Abstracted and Detailed Models

K.V. Davydova <kvdavydova@edu.hse.ru>
S.A. Shershakov<sshershakov@hse.ru>
National Research University Higher School of Economics,
PAIS Lab at the Faculty of Computer Science,
20 Myasnitskaya st., Moscow, 101000, Russia

Abstract. In this paper, we consider an approach to reverse engineering of UML sequence diagrams from event logs of information systems with a service-oriented architecture (SOA). UML sequence diagrams are graphical models quite suitable for representing interactions in heterogeneous component systems; in particular, the latter include increasingly popular SOA-based information systems. The approach deals with execution traces of SOA systems, represented in the form of event logs. Event logs are created by almost all modern information systems primarily for debug purposes. In contrast with conventional reverse engineering techniques that require source code for analysis, our approach for inferring UML sequence diagrams deals only with available logs and some heuristic knowledge. Our method consists of several stages of building UML sequence diagrams according to different perspectives set by the analyst. They include mapping log attributes to diagram elements, thereby determining a level of abstraction, grouping several components of a diagram and building hierarchical diagrams. We propose to group some of diagram components (messages and lifelines) based on regular expressions and build hierarchical diagrams using nested fragments. The approach is evaluated in a software prototype implemented as a Microsoft Visio add-in. The add-in builds a UML sequence diagram from a given event log according to a set of customizable settings.

Keywords: event logs; UML sequence diagram; reverse engineering; process mining.

DOI: 10.15514/ISPRAS-2016-28(3)-6

For citation: DavydovaK.V., ShershakovS.A. Mining Hierarchical UML Sequence Diagrams from Event Logs of SOA systems while Balancing between Abstracted and Detailed Models. *Trudy ISP RAN / [Proc. ISP RAS]*, vol.28, issue 3, 2016. pp. 85-102. DOI: 10.15514/ISPRAS-2016-28(3)-6

1. Introduction

Nowadays there are a lot of information systems. They are developed by people, which are error-prone. Systems also can have a structure which is difficult to understand. Thus, models are necessary to understand systems and find errors. When there is no complete model of a system, reverse engineering techniques can be applied to extract necessary information from the system and build an appropriate model. There are a number of tools for this purpose, they analyze source code of the system and build a model.

There are some types of models, which are useful to analyze in software engineering. For example, state machines are able to model a large number of software problems. However, they have a weakness in describing an abstract model of computation. Another example of a software model is Petri nets which can describe processes with concurrent execution. Furthermore, there are a number of models described by a standard of Unified Modeling Language (UML) for visualizing design of information systems. UML 2.4.1 [1] has two groups of diagrams, structural and behavioral ones. In particular, such kind of UML diagrams as *state class diagrams*, *statecharts* and *sequence diagrams* are widely applied to reverse engineering domain.

Almost every information system has an ability to write results of its execution to event logs. We propose approaches to mine UML sequence diagrams (UML SD) from these logs. Event logs of information systems with a service-oriented architecture (SOA) are considered and UML SDs are applied to modeling interaction between SOA information system components.

In contrast to existing reverse engineering tools, which use source code, we work with *system execution traces* in the form of event logs. A technique that allows analysis of business processes based on event logs is called process mining [2]. It uses specialized algorithms for extracting knowledge from event logs recorded by an information system. Moreover, process mining helps to check the conformance of a derived model with its earlier specification. Using execution traces works even if there is no access to the source code of an information systems. Also, not all code versions are normally stored. Moreover, large information systems tend to be distributed. Different components of a system are often implemented in different programming languages. Such a problem is solved by considering event logs instead of source code.

1.1. Motivating example

There is an event log written by a SOA-based banking information system (Table 1). We are interested in building a model in the form of a UML sequence diagram reflecting processes in the system. We have only some of the runs of the process, so one of the problems is to build an as feasible model as possible. The log contains a number of execution traces. Each trace consists of a sequence of events ordered by Timestamp attribute. Columns represent attributes of the log and rows

represent its events. System executions are maintained by different components of the system. They are grouped in attributes such as *Domain*, *Service/Process* and *Operation*. *Domains* group *Services* and *Processes*, and the latter consist of *Operations* [3].

Interaction between program system components can be represented at different abstraction levels. For example, by mapping some log attributes onto structural elements of UML SDs, such as lifelines and messages, one can get a UML SD diagram such as on Fig. 1. Specific values of these attributes appear with head names such as "Domain::Service/Process". Similarly, values of *Operation* and *Payload* attributes, which are mapped onto messages parameters appear with message arrows. Timestamp attribute sets an order of calls (time goes from the top to the bottom of a diagram).

It can also be useful to merge some messages or lifelines in order to reduce the size of a diagram and avoid "spaghetti-like" models. A regular expression suits it and an example of their usage is depicted on Fig. 2.

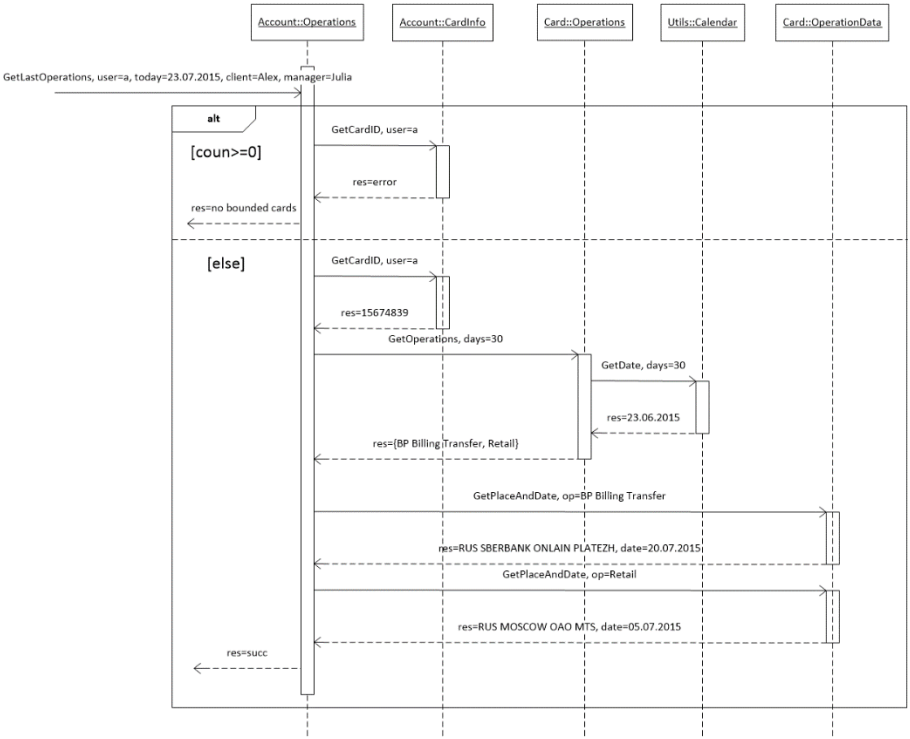


Fig. 1. Mapping log attributes onto UML sequence diagram components

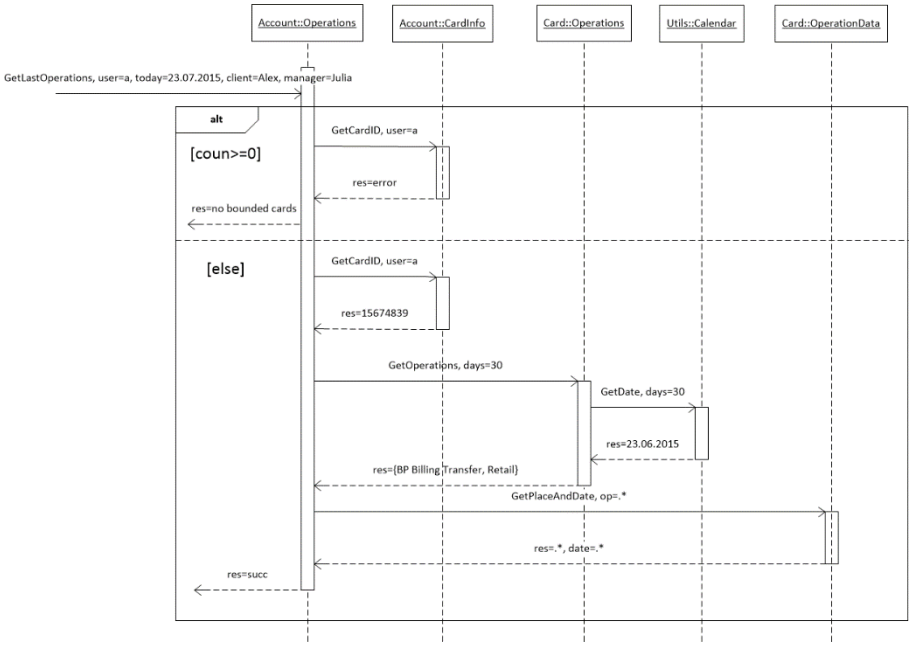


Fig. 2. Merge of diagram components based on a regular expression

Some interaction sometimes can be useful to represent on one diagram and other interactions on a nested diagram. Those both diagrams use an interaction fragment labeled *ref*. An example of a hierarchical diagram is on Fig. 3.

It would be good to have a tool which can do mapping of event log attributes on UML sequence diagram elements with ability to set an abstraction level for seeing different perspectives of the system execution. An approach approved in VTM4Visio framework is applied, which allows building these diagrams.

1.2. Related work

Reverse engineering of UML sequence diagrams is not a new problem. There are a number of works such as [4], [5], [6], [7] applied static approaches (getting models from source code without execution) for solving this problem. Moreover, there are a number of CASE tools for reverse engineering of UML sequence diagrams and other types of UML diagrams. However, most of them use static program analysis without execution of a program. Static program analysis usually uses source code or object code (a result of source code compilation). Some of these tools analyze source code, some of these tools analyze both source code and object code.

However, event logs are execution traces of source code. Thus, we do not need access to source code.

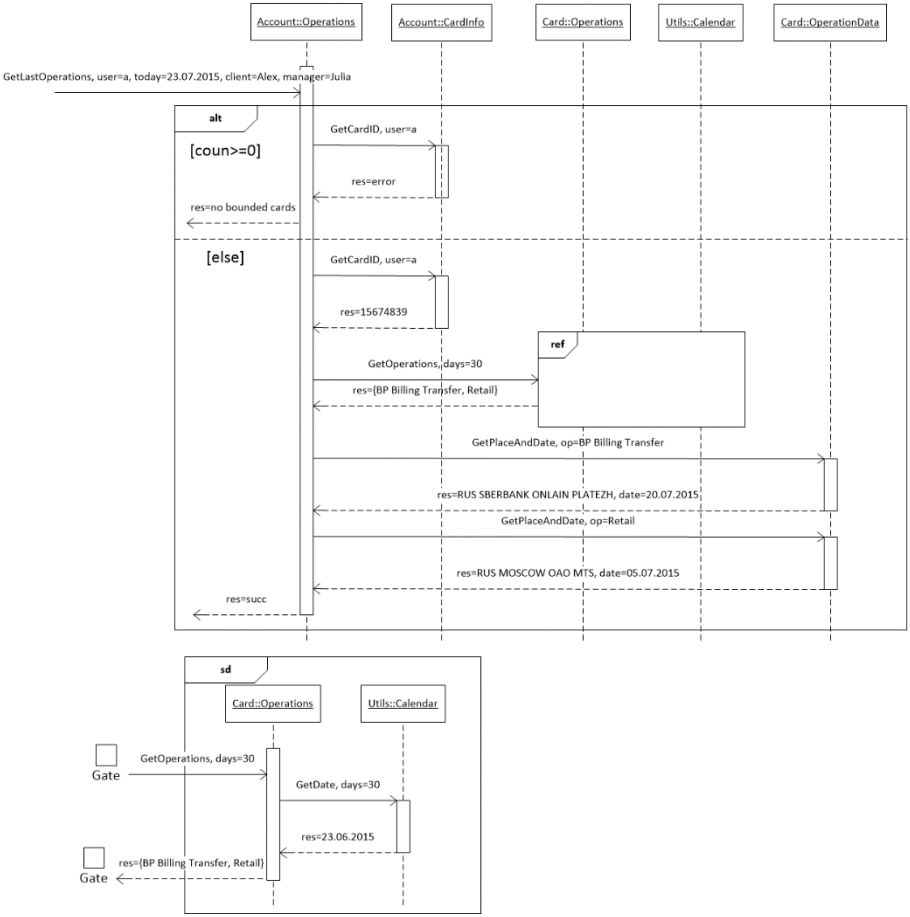


Fig. 3. Hierarchical UML sequence diagram using nested fragments

The most popular CASE tools are Sparx Systems' Enterprise Architect [8], IBM Rational Software Architect [9], Visual Paradigm [10], Altova UModel [11], MagicDraw [12], StarUML [13], ArgoUML [14]. There are both tools for end-to-end design and simple UML editors. The former include Sparx Systems' Enterprise Architect, IBM Rational Software Architect, Visual Paradigm, Altova UModel and MagicDraw, the latter include StarUML and ArgoUML. Beside that, the main aim of these tools is to get models from source code. Table 2 [15] contains CASE tools and program languages, for which models can be built. As we can see, none of these tools is able to infer models from the most popular languages used for developing

SOA information systems. Moreover, a SOA architecture can be developed with various programming languages. For example, some modules can be written in C#, others can be developed in Java, they can interact with LAMP service, so a single CASE tool cannot produce models for that system. Mining diagrams from event logs solves this problem.

Table 1. Log fragment L1. Banking SOA-system

CaseID	Domain	Service/Process	Operation	Action	Payload	Timestamp
23	Account	Operations	GetLastOperations	REQ	user=a, today=23.07.2015, client=Alex, manager=Julia	17:32:15 135
23	Account	CardInfo	GetCardID	REQ	user=a	17:32:15 250
23	Account	CardInfo	GetCardID	RES	res=15674839	17:32:15 297
23	Card	Operations	GetOperations	REQ	days=30	17:32:15 378
23	Utils	Calendar	GetDate	REQ	days=30	17:32:15 409
23	Utils	Calendar	GetDate	RES	res=23.06.2015	17:32:15 478
23	Card	Operations	GetOperations	RES	res={BP Billing Transfer, Retail}	17:32:15 513
23	Card	OperationData	GetPlaceAndDate	REQ	op=BP Billing Transfer	17:32:15 589
23	Card	OperationData	GetPlaceAndDate	RES	res=RUS SBERBANK ONLAIN PLATEZH, date=20.07.2015	17:32:15 601
23	Card	OperationData	GetPlaceAndDate	REQ	op=Retail	17:32:15 638
23	Card	OperationData	GetPlaceAndDate	RES	res=RUS MOSCOW OAO MTS, date=05.07.2015	17:32:15 735
23	Account	Operations	GetLastOperations	RES	res=succ	17:32:15 822
25	Account	Operations	GetLastOperations	REQ	user=a, today=23.07.2015, client=Alex, manager=Julia	17:40:18 345
25	Account	CardInfo	GetCardID	REQ	user=a	17:40:18 408
25	Account	CardInfo	GetCardID	RES	res=error	17:40:18 489
25	Account	Operations	GetLastOperations	RES	res=no bounded cards	17:40:18 523

Table 2. Programming languages of reverse engineering tools

Tools	Programming languages						
	PHP	C++	Java	Ruby	Python	VB	C#
Sparx Systems' Enterprise Architect	+	+	+	-	+	+	+
IBM Rational Software Architect	-	+	-	-	-	+	+
Visual Paradigm	+	+	+	+	+	-	+
Altova UModel	-	-	+	-	-	+	+
MagicDraw	-	+	+	-	-	-	+
StarUML	-	+	+	-	-	-	+
ArgoUML	-	+	+	-	-	-	+

There are some works, such as [16], [17], [18], [19], where approaches are applied for building UML sequence diagrams from program system execution traces (dynamic approaches). One of related works [16] analyzes one trace using a meta-

model of the trace and a UML SD. The trace includes information not only about invocation of methods but also about loops and conditions, which makes easier recognition of fragments such as iteration, alternatives and option. However, program systemsloggingdoesnotusuallyincludethisinformation,so it is necessary to change source code to apply this approach. In opposite to this approach, our approach recognizes fragments as conditions based on traces' difference.

There is a dynamic approach to build a UML sequence diagram based on multiple execution traces in [18]. The authors apply an approach to build a Labeled Transition System (LTS) from a trace and an algorithm to merge some LTSs into one. After that, the LTS is transformed into a UML sequence diagram. In opposite to this approach, we propose not to use other data structures to represent traces and merge them. We propose to map traces onto a UML sequence diagram directly without intermediate models, which is more efficient.

In [19] the authors pay more attention to analysis of derived models. They describe an approach briefly, without details. They mention that diagrams of one trace are merged into one UML sequence diagram. However, there is no mathematically strict definition of a trace or a UML sequence diagram and it is not clear how they merge several diagrams.

The rest of the current paper is organized as follows. Section 2 gives definitions. Section 3 introduces our approach to mining UML sequence diagrams. Section 4 discusses results of some experiments on deriving models with the help of the developed tool. Section 5 concludes the paper and gives directions for further research.

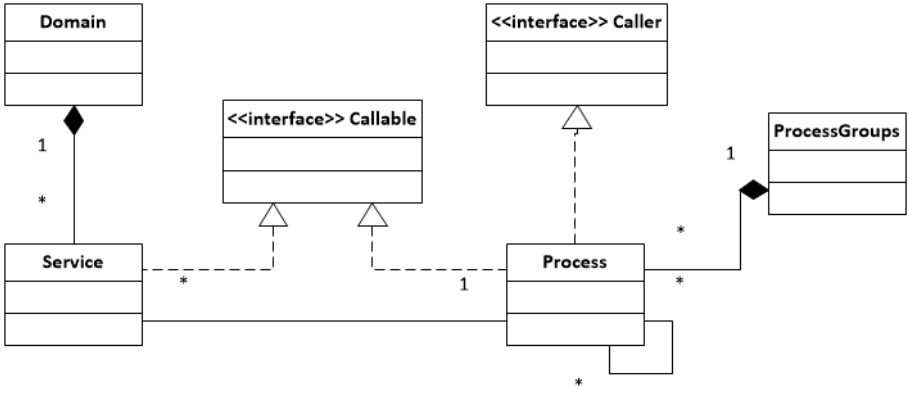


Fig. 4. Meta-model of a SOA system

2. Preliminaries

Definition 1. (Event log) Let E be a set of events. An event is a tuple $e = (a_1, a_2, \dots, a_n)$, where n is a number of attributes. $\sigma =< e_1, e_2, \dots, e_k >$ is an event trace (i.e. an ordered set of events which normally belongs to one case). $Log = \mathrm{P}(E)$ is an event log which is a multi-set of traces.

In the paper, we consider primarily event logs written by SOA information systems. The logs have a structure according to a SOA system standard. A meta-model of such a system is depicted on Fig. 4. The model complies with a Service Oriented Architecture standard (Fig. 5) proposed by Object Management Group [20].

We introduce a formal definition of a UML sequence diagram as follows.

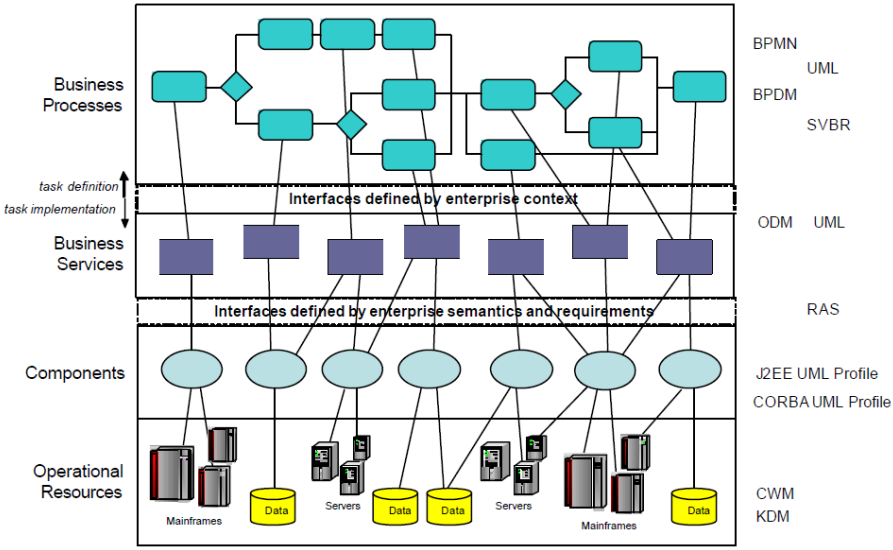


Fig. 5. Service-Oriented Architecture structure

Definition 2. (UML Sequence Diagram) A UML sequence diagram is a tuple $U_{SD} = (L, A, M, T, P, Ref, \delta)$, where:

- L is a set of lifelines, they represent objects whose interaction is shown on the diagram.
- A is a set of activations (emit and take messages) mapped onto lifelines. $A \subseteq (L \times T \times T)$
- T is time, it goes from the top of the diagram to the bottom. $\forall t \in T$, $\tau(t) = y$, where $y \in \mathbb{Z}$

- M is a set of messages (call and return) with its parameters and is ordered by time. $M \subseteq \big((A \cup Ref) \times T \times P \times (A \cup Ref)\big), m \in M: m = (a_1, t, p, a_2)$, where $a_1 \in A \cup Ref, t \in T, p \in P, a_2 \in A \cup Ref$
  $a_1 = (l_1, t_{11}, t_{12}), a_2 = (l_2, t_{21}, t_{22}): t_{11} < t_{21}, t_{11} < t_{12}, t_{21} < t_{22}$
- P is a set of parameters of messages.
- Ref is a set of *ref* fragments which group lifelines and hide their interaction. The interaction is shown on another diagram.
- $\delta: U_{SD} = (U_{SD1}, U_{HSD} | L^{'} \subseteq L, L_1 \subseteq L$
  $A^{'} \subseteq A, A_1 \subseteq A, A_1 \cap A^{'} = \emptyset$
  $M^{'} \subseteq M, M_1 \subseteq M, M_1 \cap M^{'} = \{m = (a_1, t, p, a_2) | a_1 \in A_1, a_2 \in A'\}$
  $P^{'} \subseteq P, P_1 \subseteq P, P_1 \cap P^{'} = \{p | m = (a_1, t, p, a_2),$
  $a_1 \in A_1, a_2 \in A^{'}, p \in P_1, p \in P'\}$
  $Ref^{'} \subseteq Ref, Ref_1 \subseteq Ref, Ref_1 \cap Ref^{'} = \emptyset),$
  where:
  $U_{SD} = (L, A, M, T, P, Ref)$ – a detailed diagram.
  $U_{SD1} = (L_1, A_1, M_1, T, P_1, Ref_1)$ – a diagram with *ref* fragment.
  $U_{HSD} = (L^{'}, A^{'}, M^{'}, T, P^{'}, Ref^{'})$ – a nested diagram.

3. Approach to balance between abstraction and detalization

We propose an approach to mining UML sequence diagrams from an event log with a various degree of detalization. The approach consists of three steps derived one from another. It is necessary to map attributes of the log onto elements of a diagram prior to begining a mining procedure. Some mapping functions are therefore needed. First, it is necessary to define which interaction of SOA components (*Services*, *Processes*, *Domains* etc.) must be depicted on the diagram. Function α (1) maps events of the log with their attributes onto lifelines of diagrams. It allows choosing attributes to be represented on the diagram as lifelines.

$$E = (e_1, e_2, \dots, e_k), k - a\ number\ of\ events$$
$$\alpha: U(E) \to L \qquad (1)$$

3.1. Mapping log attributes onto UML sequence diagram components

The first step allows getting diagrams with different abstraction levels by choosing log attributes for mapping onto lifelines and attributes for mapping onto parameters. To map attributes onto lifelines function α is used. Values of attributes *Domain* and *Service* are mapped onto composite lifeline objects with head names such as "Domain::Service/Process" on Figure 1. Also, function γ (2) is introduced for

mapping attributes onto message parameters. *Operation* and *Payload* attributes are mapped onto messages parameters on Figure 1 such as "Operation, Payload".

$$\gamma: U(E) \to P \tag{2}$$

The diagram depicted on Fig. 1 demonstrates interaction of services. The model represents one of the possible configurations of abstraction for the event log in table 1. For example, another possible configuration includes *Service/Process* and *Operation* attributes as diagram objects. Choosing such attributes allows inferring diagrams with different abstraction levels.

3.2. Merge of diagram components

On Figure 1 we see that the last two invocations of *GetPlaceAndDate* function are almost equal except for operation parameters. The second step of our approach performs merging some parts of a diagram. We propose to merge similar parts by using regular expressions. A regular expression contains a common part of a number of merged parts. The approach allows reducing the size of a model by merging similar parts. It increases generalization of the model.The approach involves a Cartesian square of a log with filtering. Function β (3) is used to map a filtered Cartesian square of the log on the set {1, 0} so that the element of the square is a pair "event" - "event from a set of next events". If the pair satisfies a regular expression then it is marked as 1, otherwise as 0. We introduce η (4) to compare elements of the square. η considers events as equal to each other if their corresponding attributes are equal. In this case, attributes are equal if they can be matched as a single regular expression. Functions α and γ are used in this approach for mapping event attributes onto UML sequence diagram elements. There is also introduced function ξ (5) which determines a family of messages that are satisfied with pair event attributes. A message can be just a value of attributes or a regular expression applicable to single event attributes.

$$\beta: E \times E \to \{0,1\} \tag{3}$$

$$e_i = \left(a_{i,1}, a_{i,2}, \dots a_{i,n}\right) - an\ event\ with\ n\ attributes$$
$$(e_1, e_2) \in E \times E$$
$$\widetilde{e_1} = \left(a_{1,1}, a_{1,3}, \dots, a_{1,p}\right) - an\ event\ with\ p\ sample\ attributes, p < n \tag{4}$$
$$\widetilde{e_2} = \left(a_{2,1}, a_{2,3}, \dots, a_{2,p}\right) - an\ event\ with\ p\ sample\ attributes, p < n$$
$$\eta: \widetilde{e_1} = \widetilde{e_2} \Rightarrow a_{1,1} = a_{2,1} \& a_{1,3} = a_{2,3} \& \dots \& a_{1,p} = a_{2,p}$$

$$\forall m \in M \exists \ddot{e} \in E \times E: \xi(\ddot{e}) = m \,\&\, \beta(\ddot{e}) = 1, \quad M - set\ of\ messages \tag{5}$$

If one looks at the example introduced above on Figure 2, the diagram is obtained through applying this function and regular expressions. It is noticeable that two invocations of operation *GetPlaceAndDate* are merged in one invocation with regular expressions in message parameters. Regular expression ".*" means that any sequence of symbols can be inserted instead of this expression. It is also possible to merge lifelines by using regular expressions. It can be useful if class *A* is invoked only by class *B*; so, these classes can be merged into one lifeline.

3.3. Mining a hierarchical UML sequence diagram using nested fragments

One of the ways to represent a complex model is creating a hierarchical model. The UML standard [1] allows us to divide a complex diagram into more abstract and detailed models interacting through *gates*.

In order to define a hierarchy in a UML sequence diagram we introduce a definition of a selection criterion as follows. The definition of a hierarchical UML sequence diagram is given in Definition 2.

Definition 3. (Selection criterion) Let *k* be a number of hierarchical levels and *RE* be a regular expression defined in [21] with an added symbol "." as an any symbol designation. Then, $c =< c_i | c_i \in RE >$, $c_i$is a selection criterion of events for *i*-hierarchical level. $c = c_1 \cup c_2 \cup ... \cup c_k$and $c_1 \cap c_2 \cap ... \cap c_k = \emptyset$. The regular expressions defined in [21] as selection criteria are Boolean expressions because their abstract syntax includes Boolean operations.

The components of SOA systems described by a meta-model depicted on Figure 4 have hierarchical relationship with each other. According to the SOA model there is a hierarchy in L1 event log because processes invoke different subprocesses or services.

It is also possible to distinguish some technical sublevels from main level by applying regular expressions. We propose a previously defined step with regular expressions to group elements.

Each hierarchical level is able to be encapsulated into another level on a UML sequence diagram. We propose to use nested fragments labeled as *ref*,which is defined in [1]. It allows combining high-level and detailed views of diagrams at the same time.

For applying the approach, a number of hierarchical levels and selection criteria, which are defined in Definition 3, need to be specified. Function β defines whether two events can be grouped into a single sublevel. If events match a selection criterion then they are moved to a nested diagram. For this case, values of some attributes must be equal or match a single regular expression. Function δ (Definition 2) maps some part of a UML sequence diagram considered as nested on a separate UML SD. The mapping uses *interaction use*which is shown as a *combined fragment* with operator *ref* [1]. This fragment hides some details of a

high-level diagram moved to a nested diagram while the referred diagram allows seeing the details.

On Figure 3, a hierarchical UML sequence diagram for event log L1 is depicted there. There is some elements' interaction on the high-level diagram and some interaction is abstracted as *ref* fragment and depicted on the nested one. A selection criterion used for building the diagrams is *"Operation=GetDate"* which defines a part to be abstracted.

4. Evaluation

This section discusses our evaluation of the approach presented in this paper.

4.1. VTM4Visio Framework

Microsoft Visio is a professional drawing tool for making business charts and diagrams. It also supports some of UML diagrams. Besides, Visio has reverse engineering of databases, but it does not support UML reverse engineering. One of its flexible features is that it can be expanded by add-ins. It is possible to use Visio SDK [22] for having access to a Visio object model. Thus, it is a good solution to implement our tool for visualizing results (UML sequence diagrams) of our mining algorithm.

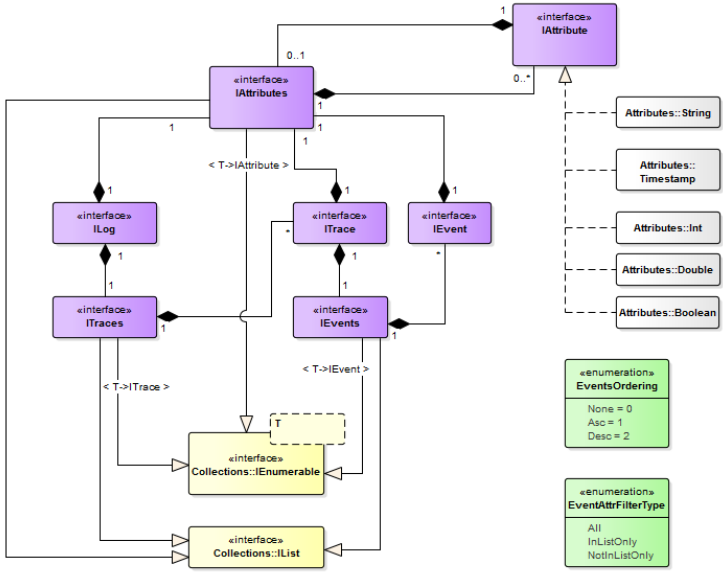


Fig. 6. Class diagram of Event log object model library

VTM4Visio is an extensible framework aimed at process mining purposing. It is implemented as an add-in for Microsoft Visio 2010. Our tool is implemented as a

plug-in, which is supported by one of the VTM4Visio components called Plugin Manager.

This framework was chosen because it provides useful instruments for accessing Microsoft Visio object models. It also has a convenient GUI.

4.2. Log pre-processing

It is necessary to have an event log in a definite format to apply our algorithm. A lot of information systems write logs in their own format. Our algorithm requires the event log to contain attributes which can be used as a case ID, timestamp and activity attributes. It is necessary to format and validate the event log before applying the algorithm.

4.3. Log library

Our algorithm requires an event log for mining a UML SD to be in some definite format. That is why it is necessary to have a library for working with event logs. We made the library and called it “Event Log Object Model Library”. Its UML class diagram is depicted on Fig. 6. The structure of our library is inspired by XES format [23]. It is not based on it but main components are taken from XES standard. We introduce special types such as *EvntsOrdering* and *EventAttrFilterType* for CSV and RDBMS-based event logs [24] because XML-based XES format is excessive. The library is written in C#. It is extensible, which allows working with different event log formats.

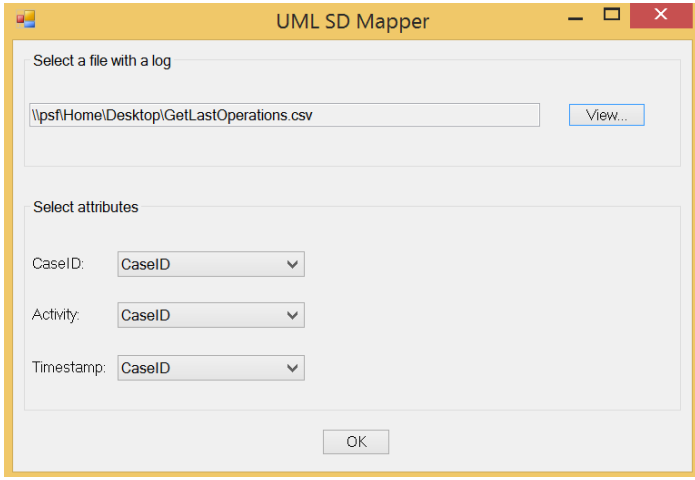


Fig. 7. Event log configuration

4.4. Prototype implementation

Our prototype was written in C# programming language as a plug-in for VTM4Visio framework. The prototype allows configuring parameters for our approaches as CaseID, Timestamp and Activity, names of lifelines and messages' parameters, a regular expression through some GUI forms (Fig. 7 and8). The configuration for reading of event logs from a file is implemented as shown on Figure 7. The configuration of the diagram is implemented as shown on Figure 8. This GUI form allows setting different perspectives and a regular expression for merging diagram elements and, hence, specifying hierarchy.

The processing result of the event log in Table 1 is depicted on Fig. 1, 2, 3.

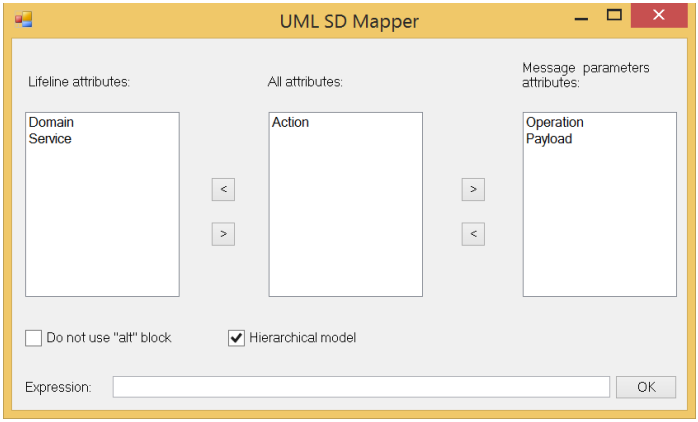


Fig. 8. Diagram configuration

5. Conclusion

This paper proposes a method of reverse engineering of UML sequence diagrams from event logs of SOA information systems. It contains three approaches to balance high-level diagrams and low-level ones.

Our method is a dynamic analysis of software because it uses only event logs. This is an advantage since source code is not always available. In addition, our approaches do not use intermediate models of an event log representation. The proposed method 1) maps log attributes onto diagram components, 2) merges diagram elements based on regular expressions and 3) builds hierarchical UML diagrams using a *ref* fragment.

Work with event logs of real-life SOA information systems shows that it is necessary to mine diagrams not only from single- threaded event logs but also from multi-threaded ones. Thus, it is a direction of our future work. UML sequence diagrams do not always show parallel interactions properly. Thus, we are going to mine hybrid diagrams as UML sequence diagrams with a *ref* fragment, which

abstracts parallel interactions and refers to UML activity diagramsillustrating parallel processes.

Acknowledgement

This work is supported by the Basic Research Program at the National Research University Higher School of Economics and the Russian Foundation for Basic Research, project No. 15-37- 21103.

References

[1]. OMG. OMG Unified Modeling Language (OMG UML), Superstructure, Version 2.4.1, August 2011.
[2]. W. M. P. van der Aalst. Process Mining: Discovery, Conformance and Enhancement of Business Processes. Springer Publishing Company, Incorporated, 1st edition, 2011.
[3]. V.A. Rubin, S.A. Shershakov System runs analysis with process mining. In Modeling and Analysis of Information Systems, pages 818–833, 2015.
[4]. A. Rountev, B.H. Connell. Object naming analysis for reverse-engineered sequence diagrams. In Proceedings of the 27th International Conference on Software Engineering, ICSE '05, pages 254–263, New York, NY, USA, 2005. ACM.
[5]. A. Rountev. Static control-flow analysis for reverse engineering of uml sequence diagrams. In Proc. 6th Workshop on Program Analysis for Software Tools and Engineering (PASTE' 05), pages 96–102. ACM Press, 2005.
[6]. P. Tonella, A. Potrich. Reverse engineering of the interaction diagrams from C++ code. pages 159–168. IEEE Computer Society, 2003.
[7]. E. Korshunova, M. Petkovic, M. G. J. van den Brand, M.R. Mousavi. Cpp2xmi: Reverse engineering of uml class, sequence, and activity diagrams from C++ source code. In WCRE, pages 297–298. IEEE Computer Society, 2006.
[8]. Sparx Systems' Enterprise Architect. http://www. sparxsystems.com.au/products/ea/.
[9]. IBM Rational Software Architect. https://www.ibm.com/ developerworks/downloads/r/architect/.
[10]. Visual Paradigm. https://www.visual-paradigm.com/ features/.
[11]. Altova UModel. http://www.altova.com/umodel.html.
[12]. MagicDraw. http://www.nomagic.com/products/magicdraw.html.
[13]. StarUML. http://staruml.io.
[14]. ArgoUML. http://argouml.tigris.org.
[15]. H. Osman, M. R. V. Chaudron. Correctness and completeness of CASE tools in reverse engineering source code into UML model. The GSTF Journal on Computing (JoC), 2(1), 2012.
[16]. L. C. Briand, Y. Labiche, J. Leduc. Toward the reverse engineering of uml sequence diagrams for distributed java software. IEEE Trans. Softw. Eng., 32(9):642–663, September 2006.
[17]. R. Delamare, B. Baudry, Y.L. Traon. Reverse-engineering of UML 2.0 sequence diagrams from execution traces. In Proceedings of the workshop on Object-Oriented Reengineering at ECOOP 06, Nantes, France, July 2006.

[18]. T. Ziadi, M.A.A. da Silva, L.M. Hillah, M. Ziane. A fully dynamic approach to the reverse engineering of UML sequence diagrams. In Isabelle Perseil, Karin Breitman, and Roy Sterritt, editors, ICECCS, pages 107–116. IEEE Computer Society, 2011.
[19]. Y.-G. Guéhéneuc. Automated reverse-engineering of UML v2.0 dynamic models. In Proceedings of the 6 th ECOOP Workshop on Object-Oriented Reengineering. http://smallwiki.unibe.ch/WOOR, 2005.
[20]. OMG. The OMG and Service Oriented Architecture, 2006
[21]. S. Owens, J. Reppy, A. Turon. Regular-expression derivatives re-examined. J. Funct. Program., 19(2):173–190, March 2009.
[22]. Visio 2010: Software Development Kit, 2010. https://www.microsoft.com/en-us/download/details.aspx?id=12365.
[23]. C. W. Günther and E. Verbeek. XES Standart Definition version 2.0, 2014.
[24]. S.A. Shershakov. VTMine framework as applied to process mining modeling, International Journal of Computer and Communication Engineering vol. 4, no. 3, pp. 166-179, 2015.

Метод автоматического построения иерархических UML-диаграмм последовательности с задаваемым уровнем детализации на основе журналов событий

К.В. Давыдова <kvdavydova@edu.hse.ru>
С.А. Шершаков <sshershakov@hse.ru>
Национальный исследовательский университет Высшая школа экономики, лаборатория ПОИС факультета компьютерных наук, 101000, Россия, г. Москва, ул. Мясницкая, д. 20

Аннотация. В данной статье мы предлагаем метод автоматического построения диаграмм последовательности UML на основе журналов событий информационных систем с сервис-ориентированной архитектурой (SOA). Диаграммы последовательности UML — графические модели, подходящие для представления взаимодействий в гетерогенных компонентных системах, в частности, в набирающих сейчас популярность информационных SOA-системах. Описываемый метод использует трассы исполнения SOA-систем, представленные в виде журналов событий. Почти все современные информационные системы имеют возможность записывать результаты своей работы в журналы событий, которые используются в основном для процесса отладки. По сравнению с традиционными техниками автоматического синтеза моделей, которые требуют не всегда имеющийся исходный код для своей работы, наш метод для автоматического построения диаграмм последовательности UML работает только с доступными журналами событий и некоторыми эвристическими данными. Метод состоит из нескольких этапов построения диаграмм последовательности UML в зависимости от разной перспективы, заданной аналитиком.Они включают отображение атрибутов журнала событий на элементы диаграммы с возможностью задать уровень абстракции через параметры, группировку некоторых компонент диаграммы и построение иерархических диаграмм последовательности. Мы предлагаем группировать некоторые компоненты (сообщения и линии жизни) на основе регулярных выражений и строить иерархические диаграммы,

используя вложенные фрагменты. Мы апробировали данный метод при помощи разработанного в виде плагина Microsoft Visio прототипа. Плагин строит диаграмму последовательности UML на основе заданного журнала событий в соответствии с набором настраиваемых параметров.

Ключевые слова: журнал событий; диаграмма последовательности UML; автоматическое выведение моделей; извлечение процессов.

DOI: 10.15514/ISPRAS-2016-28(3)-6

Для цитирования:Давыдова К.В., Шершаков С.А.Метод автоматического построения иерархических UML-диаграмм последовательности с задаваемым уровнем детализации на основе журналов событий.*Труды ИСП РАН,* том 28, вып. 3, 2016. стр. 85-102 (на английском). DOI: 10.15514/ISPRAS-2016-28(3)-6

Список литературы

[1]. OMG. OMGUnifiedModelingLanguage (OMGUML), Superstructure, Version 2.4.1, August 2011.
[2]. W. M. P. vanderAalst. Process Mining: Discovery, Conformance and Enhancement of Business Processes. Springer Publishing Company, Incorporated, 1st edition, 2011.
[3]. V.A. Rubin, S.A. Shershakov System runs analysis with process mining. In Modeling and Analysis of Information Systems, pages 818–833, 2015.
[4]. A. Rountev, B. H. Connell. Object naming analysis for reverse-engineered sequence diagrams. In Proceedings of the 27th International Conference on Software Engineering, ICSE '05, pages 254–263, New York, NY, USA, 2005. ACM.
[5]. A. Rountev. Static control-flow analysis for reverse engineering of uml sequence diagrams. In Proc. 6th Workshop on Program Analysis for Software Tools and Engineering (PASTE' 05), pages 96–102. ACM Press, 2005.
[6]. P. Tonella, A. Potrich. Reverse engineering of the interaction diagrams from C++ code. pages 159–168. IEEE Computer Society, 2003.
[7]. E. Korshunova, M. Petkovic, M. G. J. van den Brand, M. R. Mousavi. Cpp2xmi: Reverse engineering of uml class, sequence, and activity diagrams from C++ source code. In WCRE, pages 297–298. IEEE Computer Society, 2006.
[8]. Sparx Systems' Enterprise Architect. http://www. sparxsystems.com.au/products/ea/.
[9]. IBM Rational Software Architect. https://www.ibm.com/ developerworks/downloads/r/architect/.
[10]. Visual Paradigm. https://www.visual-paradigm.com/ features/.
[11]. Altova UModel. http://www.altova.com/umodel.html.
[12]. MagicDraw. http://www.nomagic.com/products/magicdraw.html.
[13]. StarUML. http://staruml.io.
[14]. ArgoUML. http://argouml.tigris.org.
[15]. H. Osman, M. R. V. Chaudron. Correctness and completeness of CASE tools in reverse engineering source code into UML model. The GSTF Journal on Computing (JoC), 2(1), 2012.

[16]. L. C. Briand, Y. Labiche, J. Leduc. Toward the reverse engineering of uml sequence diagrams for distributed java software. IEEE Trans. Softw. Eng., 32(9):642–663, September 2006.

[17]. R. Delamare, B. Baudry, Y. L. Traon. Reverse-engineering of UML 2.0 sequence diagrams from execution traces. In Proceedings of the workshop on Object-Oriented Reengineering at ECOOP 06, Nantes, France, July 2006.

[18]. T. Ziadi, M. A. A. da Silva, L. M. Hillah, M. Ziane. A fully dynamic approach to the reverse engineering of UML sequence diagrams. In Isabelle Perseil, Karin Breitman, and Roy Sterritt, editors, ICECCS, pages 107–116. IEEE Computer Society, 2011.

[19]. Y.-G. Guéhéneuc. Automated reverse-engineering of UML v2.0 dynamic models. In Proceedings of the 6 th ECOOP Workshop on Object-Oriented Reengineering. http://smallwiki.unibe.ch/WOOR, 2005.

[20]. OMG. The OMG and Service Oriented Architecture, 2006

[21]. S. Owens, J. Reppy, A. Turon. Regular-expression derivatives re-examined. J. Funct. Program., 19(2):173–190, March 2009.

[22]. Visio 2010: Software Development Kit, 2010. https://www.microsoft.com/en-us/download/details.aspx?id=12365.

[23]. C. W. Günther and E. Verbeek. XES Standart Definition version 2.0, 2014.

[24]. S.A. Shershakov. VTMine framework as applied to process mining modeling, International Journal of Computer and Communication Engineering vol. 4, no. 3, pp. 166-179, 2015.

Applying MapReduce to Conformance Checking

I.S. Shugurov <shugurov94@gmail.com>
A.A. Mitsyuk <amitsyuk@hse.ru>
National Research University Higher School of Economics, Laboratory of Process-Aware Information Systems, 20 Myasnitskaya St., Moscow, 101000, Russia

Abstract. Process mining is a relatively new research field, offering methods of business processes analysis and improvement, which are based on studying their execution history (event logs). Conformance checking is one of the main sub-fields of process mining. Conformance checking algorithms are aimed to assess how well a given process model, typically represented by a Petri net, and a corresponding event log fit each other. Alignment-based conformance checking is the most advanced and frequently used type of such algorithms. This paper deals with the problem of high computational complexity of the alignment-based conformance checking algorithm. Currently, alignment-based conformance checking is quite inefficient in terms of memory consumption and time required for computations. Solving this particular problem is of high importance for checking conformance between real-life business process models and event logs, which might be quite problematic using existing approaches. MapReduce is a popular model of parallel computing which allows for simple implementation of efficient and scalable distributed calculations. In this paper, a MapReduce version of the alignment-based conformance checking algorithm is described and evaluated. We show that conformance checking can be distributed using MapReduce and can benefit from it. Moreover, it is demonstrated that computation time scales linearly with the growth of event log size.

Key words: process mining; conformance checking; MapReduce; Hadoop; big data.

DOI: 10.15514/ISPRAS-2016-28(3)-7

For citation: Shugurov I.S., Mitsyuk A.A. Applying MapReduce to Conformance Checking. *Trudy ISP RAN / Proc. ISP RAS]*, vol.28, issue 2, 2016. pp. 103-122. DOI: 10.15514/ISPRAS-2016-28(3)-7

1. Introduction

Ever-increasing size and complexity of modern information systems force both researchers and practitioners to find novel approaches of formal specification, modeling, and verification. This process is essential for ensuring their robustness and for possible optimization and improvements of existing business processes.

Process mining is a research field, which offers such approaches [1]. *Process mining* is a discipline, which combines techniques from data analysis, data mining, and conventional process modeling. Typically, three main sub-fields of process mining are distinguished in the literature: (1) process discovery; (2) conformance checking and (3) enhancement [1].

The aim of *process discovery* is to build a process model based solely on the execution history of a particular process. Event logs are the most common and natural way of persisting and representing execution history. By an event log, we understand a set of traces where each trace corresponds exactly to one process execution. A typical process discovery algorithm takes an even log as an input parameter and constructs a process model which adequately describes the behavior observed in the event log.

The task of *conformance checking* is to measure how well a given process model and an event log fit each other. Furthermore, showing only the coefficient of conformance is usually insufficient for real-life application since analysts often need to see where and how often deviations happen in order to draw any conclusions. Therefore, it is often the case when conformance checking algorithms include computation of additional metrics as well as visualization of deviations.

Process enhancement deals with improvements of processes as well as corresponding process models.

One of the challenges of process mining, when applied in real life, is the size of data to be processed and analyzed [2], [3]. Since process discovery has drawn significant attention of researchers, there are a number of solutions which allow for fast process discovery from large event logs [4]. These solutions vary from using distributed systems and parallel computing [5] to applying more efficient algorithms, which require less data scans and manipulations [6], [7]. In contrast, conformance checking remains problematic to be made fast due to its theoretical and algorithmic difficulties. At the same time, efficient, easy-to-use and robust conformance checking is the key to better process improvement since enhancement approaches often rely heavily on measuring conformance (for example, see model repair approaches [8], [9]).

This paper concentrates on implementation details of distributed conformance checking rather than on its theoretical aspects. It describes a possible way of speeding up conformance checking. It implies improving one of the existing conformance checking algorithms so that it can be executed in a distributed manner by means of using MapReduce [10]. One of the very first papers discussing distributed conformance checking [11] was dedicated solely to theoretical foundations of process models and event logs decomposition. The author takes a look at the algorithmic side of distributed conformance checking and totally skips problems of its software implementation. In this paper, we consider practical aspects of distributed conformance checking. Furthermore, we prove viability of the proposed approach by demonstrating that it really allows measuring conformance of bigger event logs better than currently existing approaches.

This paper is structured as follows. Section 2 introduces foundational concepts we use in the paper. In section 3, the reader can find the main contribution. Section 4 proposes several improvements of the approach proposed in section 3. An implementation of the presented approach is described in section 5. Related work is reviewed in section 6. Finally, section 7 concludes the paper.

2. Preliminaries

In this paper, we consider process models in the Petri net (simple P/T-nets) notation. A *Petri net* is a bipartite graph, which consists of nodes of two types. In process mining, transitions, denoted by rectangles, are considered as process activities, whereas places, denoted by circles, designate the constraints imposed on the control-flow. String labels may be associated with transitions in order to show the correspondence between activities and transitions. Transitions without labels are called *silent*. It implies that silent transitions model behavior and constrains of an activity in a process, executions of which are not recorded into event logs. Each place denotes a causal dependence between two or more transitions. Places may contain so-called *tokens*. A transition may fire if there are tokens in all places connected to it via incoming arcs. When fired, it consumes one token from each input place and produces one token to each output place. *Marking* is a distribution of tokens over all places of a Petri net, thus a marking denotes the current state of a process.

An *event log* is a recorded history of process runs. Usually the execution of a process in some information system is recorded for documenting, administrative, security, and other purposes. The main goal of process mining is to explore and use these data for the diagnosis and improvement of actual processes.

We consider event logs of standardized nature as they are used in process mining. Formally, an event log is multiset of traces where each trace is a sequence of events. Each trace corresponds to exactly one process run. An event contains the name of associated activity, timestamp, performer name and may contain other additional properties. In this paper we consider simple event logs, in which events contains only names of activities. An example model and the corresponding event log are shown in fig. 1.

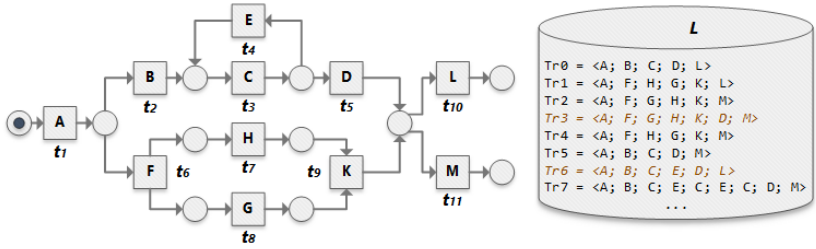


Fig. 1. Petri net and event log

2.1 Conformance checking

The conformance checking and its place in process mining are defined in [1]. Usually four dimensions of conformance are considered: fitness, precision, generalization, and simplicity. However, this paper focuses exclusively on fitness. By the term *fitness*, we understand the extent to which a model can reproduce traces from an event log. In other words, fitness shows how well the model reflects the reality. The fitness dimension is typically regarded as being the most frequently used and best-defined [1] among the dimensions.

Nowadays, the most advanced and refined conformance checking approach is the one using alignments [12]. The term *alignment* is used to denote the set of pairs where each pair consists of an event from an event log and a corresponding transition of a model. Such pairs are constructed sequentially for each event in a trace. A simple alignment for the trace $Tr3$ (see fig. 1) is depicted in fig. 2. However, it is allowed to pair an event with no transitions (a special ”no move“ symbol ≫). This means that the event is present in a log but cannot be replayed by any transition in the model. It is also possible to map a transition to no events (this is denoted by the same symbol ≫). In that case, the transition is fired but there is no evidence of this fact in the event log. Thus, there are two main types of steps composing any alignment: a synchronous move (a transition fired with the same label as an event name from the event log) and a non-synchronous move (a transition label and an event name are the different ones or a move is skipped either in the model or in log).

Fig. 2. Alignment

Alignments help to measure the difference between a trace from an event log and behavior specified by a model. In order to quantify the difference one has to calculate the number of non-synchronous moves and assess their significance. This assessment is accomplished by introducing a *cost function*, which is used for calculating *cost of an alignment*. By cost, we understand a number which somehow designates the significance. The general idea is that some deviations are more severe than others, thus these deviations have more impact on the overall conformance. Using cost function one can assign cost for each type of deviation for each transition and event. Thus, cost function maps a pair of an event and a transition to a number, which signifies a penalty for having such a pair in a trace. The more the cost is, the more significant this deviation is. Assuming that all costs are set to 1, the alignment shown in fig. 2 has the cost 1, because there is only one nonsynchronous move in it (event D in the trace has to be skipped during model run). Accumulating costs for all alignments of a particular event log, it is possible to derive the cost for the entire log.

It is possible that a particular run through the model and a particular trace have several possible alignments. In order to choose between them a cost function is used to evaluate the cost of each alignment. An alignment with the lowest cost is selected as the optimal alignment. According to [12], it makes sense to use only optimal alignments when calculating fitness. Alignment-based fitness can be measured using the metric defined in [13]:

$$f(L,N) = 1 - \frac{\sum_{tr \in L} \sum_{e \in tr} cost_{fn}^{\delta_{\text{opt}}}(e,N)}{\sum_{tr \in L} cost_{ai}}$$

where L is an event log, N is a model, $cost_{fn}^{\delta_{opt}}(e,N)$ is a cost of a pair $(e,(t_i,t_i^l))$ (e is an event, t_i is a transition from model run, t_i^l is its label) in the particular optimal alignment δ_{opt}, which depends on used cost function cf, $cost_{ai}$ is a total cost of the trace tr if all moves in it are considered as non-synchronous. Thus, fitness is a normalized ratio of the accumulated costs calculated for the optimal alignments to the accumulated costs for the worst possible alignments for a particular event log.

It is shown in [12] that construction of alignments and selection of optimal among them for each trace can be converted to solving the shortest path problem. Formally, a trace from the event log is represented as an event net, which is a special Petri net having the form of the sequence of transitions connected through places. Then the product of the model and this event net is constructed. It is shown in [12] that the problem of optimal alignment calculation can be viewed as a problem of finding a firing sequence in this product, which can be achieved by using a state-space exploration approach.

The proposed approach has a low computational performance when dealing with large models, large event logs or in case of low fitness because of the necessity to solve the shortest path problem, especially for model of certain types [12]. The author himself states in [12] that ”from a computational point of view, computing alignments is extremely expensive“. Moreover, its existing implementation keeps the processed models, event logs, event nets, and computed alignments in computer’s main memory. This approach allows for flexible configuration of visualization settings, and, in some cases, faster completion. However, this feature makes usage of existing implementation rather hard and inconvenient because the algorithm typically consumes several gigabytes of main memory even for processing relatively small models and small event logs (dozens of megabytes). Thus, it is not suitable for real-life usage.

This paper proposes a way of checking conformance between process models and big event logs of gigabyte sizes using MapReduce.

2.2 MapReduce

MapReduce is a computational model proposed and popularized in [10], although the idea dates back to the origins of functional programming. MapReduce is a

popular technology among practitioners and a research area among scientists. It has a good tool support; all major cloud platform vendors provide the possibility to execute MapReduce jobs on their cloud clusters.

The model simplifies parallel and distributed computing by allowing software developers to define only two quite primitive functions: *map* and *reduce*. At each invocation of a map function (also called *mapper*), it takes a key-value pair and produces an arbitrary number of key-value pairs. The aim of reduce functions (also called *reducers*) is to aggregate values with the same key and perform necessary computations over them. Thus, a reduce function takes a key-list pair as input parameters. Usage of such rather trivial functions makes their distribution straightforward. Last but not least, comes another important function allowed by MapReduce which is called *combine*. Its main purpose is to perform reduce-like computation between mappers and reducers. Combine functions (also known as *combiners*) are invoked on the same very computers as mappers. Combiners allow for further parallelizing computations and decreasing amount of data transferred to reducers and processed by them. It was pointed out even in the original article [10] that combiners may dramatically decrease computation time.

One of the most crucial advantages of MapReduce is that algorithms expressed in such a model are inherently deadlock-free and parallel. Another important advantage is the tendency to perform computations where required data resides. Generally, computation of map tasks take place where the required data is stored since its location is known beforehand. Such an approach ensures that data transfer between computers and latency, inflicted by it, are minimized. Ideally, data is transferred between computers where map tasks are executed and computers where reduce tasks are executed. Unfortunately, it is rarely achievable since all files are separated into smaller parts, called blocks, and distributed (and also replicated) over a cluster, thus data needed for execution of a single map task may reside in different data chunks — there will be a need to move a portion of data from one computer to another.

3. Fitness measurement using MapReduce

This section describes the approach we propose for checking conformance.

The few adjustments of the existing conformance checking algorithm with alignments need to be done in order to implement the proposed schema. It is expected that the algorithm will benefit if distribution is applied to traces. It means that traces are distributed over a cluster so that their alignments can be computed in parallel. Another possible option was to distribute computation of each alignment since efficient distributed graph algorithms for solving the shortest path problem are known. However, use of them seems excessive because they are aimed at solving problems on graphs consisting of thousands and millions of nodes, which is not the case for business process models. A process model consisting of more than a hundred nodes seems unrealistic.

The general schema is depicted in fig. 3. Map function takes traces one by one and computes their alignments. This process can easily be carried out in parallel since, by its definition, an alignment is computed individually for each trace. It is enough to use a single reduce function, which aggregates fitnesses of all traces and calculating fitness of the overall event log. Single reducer implies that key-value pairs emitted by all mappers have the same key. Single reducer can be considered as a bottleneck due to the reason that before it can start processing it waits for completion of all maps and transition of all costs to a single computer. To diminish the negative effect of a single reducer, a combiner function comes in handy. The problem is that calculating average is not an associative operation, thus it is impossible to use the basic reduce function instead of the combine function. We implemented it in a manner resembling the one described in [14]. The general idea is that calculating average can be easily decomposed into calculating a sum of all entries of some metric and counting a number of entries, where both of them are associative operations. It implies changing the structure of values used in key-value pairs. The modified version of values contains not only statistics (fitness and so on) but also a counter which shows how many traces describes a particular value. Given that, combiners only have to sum the values they receive and increment the counter.

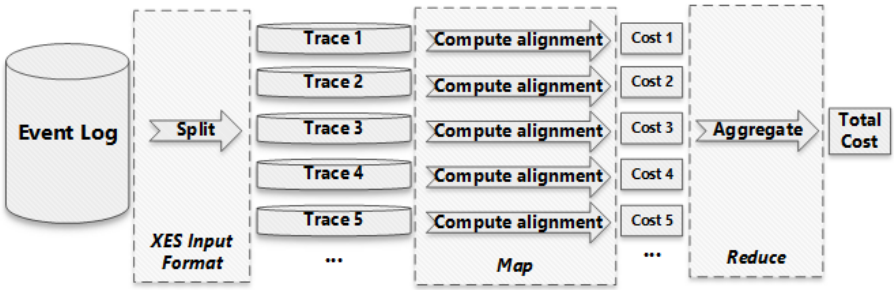


Fig. 3. Conformance checking with MapReduce

4. Potential improvements

One of the possible improvements of the algorithm is to enhance it by adding trace deduplication. When large event logs are considered, the possibility of the equivalent traces occurring several times is very high. Hence, it might be desired to find only unique traces, number of their occurrences and compute alignments only for them. It will allow for lessening the number of computed alignments. However, efficient MapReduce algorithm for deduplication of event sequences is far from trivial. Moreover, it is not guaranteed that time needed for deduplication and subsequent conformance checking will be shorter than in case of using the standard approach. This question can only be answered by conducting relevant experiments.

Even though process models is not prone to be large, a lot of time is still required for checking conformance. Another possible improvement, which aims at reducing model size, is to employ the ”divide and conquer“ principle. The way in which the

principle can be applied to cope with high computational complexity of conformance checking was proposed in [15] and [16]. The general idea is to divide a process model into smaller sub-parts. Next step is event log projection. This means that for each fragment of a model all events from the event log that correspond (names of events are equal to labels of activities) to a particular fragment are selected. As a result, we get as many projected event logs as decomposed Petri net fragment.

Once it is done, alignments and costs of each fragment can be computed. Then it is possible to sum costs of parts following specific rules to get a lower bound of the cost of the entire log. Having these costs, an upper bound of fitness can be computed. Performance gain is the most crucial motivation of this approach. Since time needed for computing alignments depends on trace size, usage of smaller parts of the model ensures faster computation. A wide range of model decomposition strategies have been proposed in [17], [15], [18], which leaves the user with the necessity to empirically choose between them. Last but not least, decomposition also incurs time overhead and projected event logs takes up disk space, so usage of the algorithm is not beneficial (or even feasible) in all the possible cases. Furthermore, there is no research done to establish when usage of which approach makes more sense.

It is possible to employ a similar approach in the MapReduce environment. There are two possible options: (1) computation of the overall event log fitness and (2) computation of fitness of each separate model part. In all the cases fitness is computed in a three-stage process as it is shown in fig. 4. The zero stage again is the splitting of the log by traces, which is followed by trace decomposition. Traces are decomposed using the maximal decomposition described in [15]. However, incorporation of other decomposition techniques [15], [17], [16] is also possible. At the second stage, alignments of sub-traces are computed and then aggregated. The final stage differs depending on the selected computation option. At this stage either fitness of the overall event log is computed at a single reducer or fitnesses of individual parts are computed at different reducers (the number of reducers can be up to the number of model parts). If fitness of individual process parts is calculated, after the second map unique identification of a model part is used as a key for emitted key-value pairs. When decomposition is applied, log deduplication's importance and potential benefit grow even more.

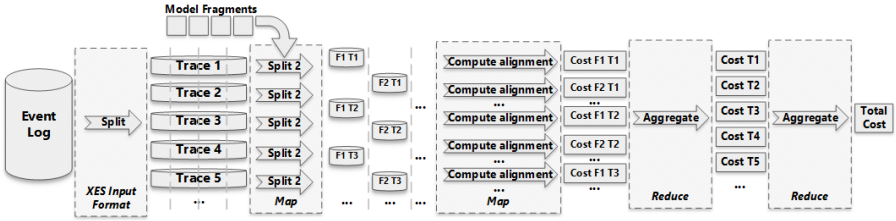


Fig. 4. Possible approach with vertical decomposition

5. Implementation and testing

This section describes the actual implementation[1] of the proposed approach and its experimental testing. Hadoop [19] was used for implementation and testing of the approach because it is a common and widely supported open source tool.

5.1 Implementation

The original algorithm was implemented as a ProM Framework plugin. The ProM Framework [20], [21] is a well-known tool for implementation of process mining algorithms. The ProM Framework consists of two main components:

- ProM core libraries which are responsible for the main functionality used by all users and extensions,
- extensions (typically called plugins) which are created by researchers and are responsible for import/export operations, visualization, and actual data processing.

The platform is written in such a way that it allows plugins to use data produced by other plugins. Furthermore, ProM encourages programmers to separate concerns: export plugins are only used for exporting data, visualization plugins are used for visualizing objects. As a result, a common usage scenario always consist of a chain of invocations of different plugins. Among main advantages of ProM are configurability, extensibility, and simplicity of usage. Last but not least, the platform allows researchers to easily create and share plugins with others thus extending the tool and contributing to the overall field of process mining. Despite all these positive sides, usage of ProM can be inconvenient and tedious, if the desired goal is unusual in any way.

XES [22] is often considered as a de facto standard for persisting event logs in the area of process mining. Technically, it is an XML-based standard, which means that it is tool-independent, extensible, and easy to use. Moreover, ProM fully supports this standard and has all required plugins for working with it.

Our approach involves usage of raw event logs stored in the format of XES only at the zero step of the algorithm. Before separate traces are available for the required computations, it is necessary to sequentially read XES files dividing them into separate traces. It is accomplished by using the *XMLInputFormat* from the *Mahout* project [23]. XMLInputFormat provides the capability of extracting file parts located between two specified tags. Moreover, the class is responsible for ensuring that the entire requested part (in our case — trace) is read, no matter in which blocks and on which data nodes it resides.

The fact that the initial algorithm was implemented for ProM inflicts several inconveniences for its distribution. First of all, it is assumed that the plugin is invoked by ProM via a special context. Essentially, it implies several things:

- the entire ProM distribution has to be sent to each computational node,

[1] The tool is available at https://sourceforge.net/p/distributedconformance/

- at each computational node, it is required to start up ProM (it may take up to couple of minutes on an average computer).

As a result, it may significantly increase latency and incur higher time needed for termination of computations. To avoid this, it was decided to alter implementation in such a way that a number of libraries the algorithm depends on in as minimal as it is possible to achieve. In other words, on the one hand it was desired to separate the implementation of the algorithm from ProM. On the other hand, usage of ProM could be useful for initial settings and visualization of final results. As a result, we achieved such a level of decoupling, that it is possible to launch the algorithm completely autonomously without the need of installation of the ProM Framework or any ProM plugins.

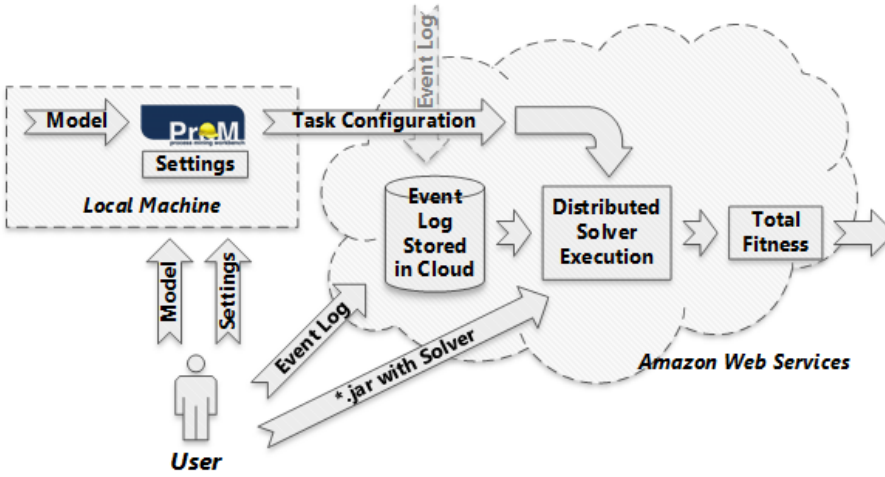


Fig. 5. Implementation of the approach

The resulting architecture is illustrated in fig. 5. Conformance measurement is done in two-step approach. At first step, the user loads a model, represented by a Petri net into a special ProM plugin, which serves for setting the options of the alignments-based conformance algorithm (mapping between transitions and events in event logs, costs of insertion and skipping in alignments). We use standard ProM classes for representing Petri nets because they allow for easier compatibility with other ProM plugins. Loading a model to a main memory should not be a problem because it is highly unlikely for such models to contain even hundreds of nodes, thus the size of process models is typically relatively small. Another possible option was to specify settings exclusively via XML files, though we found it less intuitive and convenient than visual settings. Once the algorithm is configured, settings are written to a file which later will be uploaded to a cluster. Last but not least, it is important to state that this ProM plugin depends neither on Hadoop nor on a chosen cloud cluster nor on any other auxiliary Hadoop libraries.

When Hadoop job is initiated, the user is asked to specify directories where event logs are placed, a path to a Petri net, and a path to conformance settings. A model and settings are then automatically added into the Hadoop distributed cache — the files are replicated to each data node, so they are available for fast access by any mapper. At a startup of each model, the files are loaded into main memory because they will be used for all the alignment computations.

After completion of conformance measurement, the results are written to a single file, which afterwards can be downloaded and viewed in ProM. Another sub-task is to find in which cases deduplication is worthwhile and how exactly it affects computational time.

5.2 Experimental results

The proposed algorithm was tested and evaluated using Amazon Web Services [24]. In our cluster, we used five m3.xlarge instances (one as a master node, four as data nodes). A local computer used for conducting experiments with the original algorithm had the following configuration: Intel Core i7-3630QM, 2.40 GHz, 8 GB of main memory, Windows 7 64 bit.

For testing purposes, we created a process model comprising some of the main workflow patterns: sequence, parallel split, synchronize, exclusive choice, and simple merge [25]. Afterwards, several models derived from the original were created — they all differ in fitness. Artificial event logs were generated using the approach proposed in [26]. Logs were generated only for the original model. All resulting logs were of different sizes.

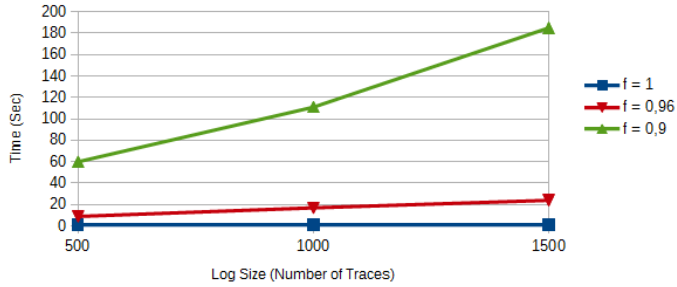


Fig. 6. Computation time of the standard approach

Fig. 6 illustrates how computation time depends on a number on traces and fitness. It is clear from the plots that computational complexity scales linearly with the growth of a number of traces. Moreover, it is seen that computation time highly depends on fitness. The lower the fitness, the slower the computations will be. It seems that computation time does not scale linearly with the decrease of fitness if the same quantity of logs is used. The clear indicators are the margins between lines representing fitness 1 and 0.96, and 0.96 and 0.9. Furthermore, we can conclude that

the lower fitness, the faster computation time increases with the rise of the number of traces.

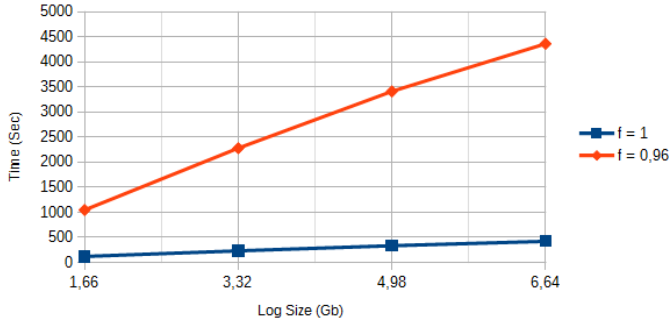


Fig. 7. Computation time with MapReduce

Fig. 7 provides an overview of how the algorithm scales when it is distributed using MapReduce. It is worth mentioning that 1.66 Gb of logs contain 500 thousand traces. As in the case of the not distributed algorithm, the graph shows that the algorithm scales linearly with the increase of a number of traces. Furthermore, similarly to the not distributed case, for non-fitting models computations take considerably longer than for perfectly fitting ones, and that computation time grows faster for non-fitting models.

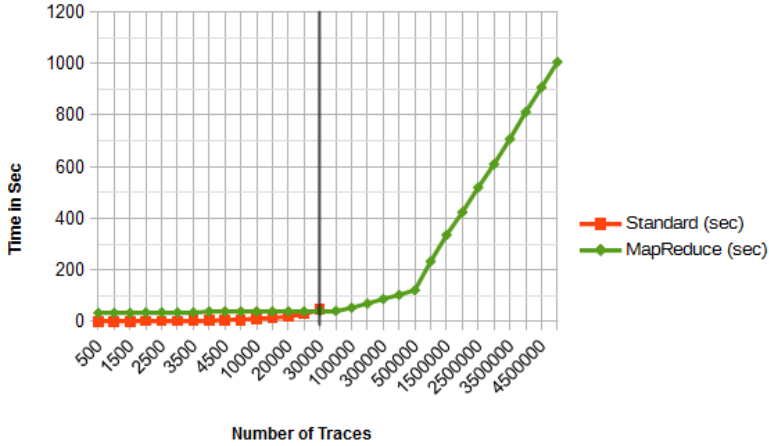


Fig. 8. Comparison of the standard and the distributed approaches

In fig. 8, a comparison of distributed and not distributed approaches is provided. Unfortunately, it is impossible to establish exactly when the distributed implementation beats the original in terms of performance since the original one cannot handle event logs of considerable size. In addition, the original algorithm

was not able of handling more than a hundred of Mbytes. On data of such small sizes, MapReduce and Hadoop fail to work efficiently because they are designed for processing much bigger files. In fact, Hadoop does not parallelize processing of files which are smaller than a single file block. It is clear from Figure 8 that for relatively small event logs the distributed version works more slowly. It is clear from the graph that our solutions can handle event logs of several dozens of GBs even on a small cluster used for conducting these experiments.

6. Related work

Although applicability of MapReduce or distributed systems for the tasks of process mining has not drawn significant attention yet, there are a few papers, which consider this subject.

In [27] the authors focus exclusively on finding process and events correlation in large event logs. According to them, MapReduce solution for such a computationally and data intensive task as events correlation discovery performs well and can be scaled to large datasets.

Other works where the authors study applicability of MapReduce to process mining are [28], [29]. In these articles, a thorough description of several popular discovery algorithms is provided (the alpha algorithm [30], and the flexible heuristics miner [31]). Every one of them consists of several consequent MapReduce jobs. First MapReduce job is responsible for reading event logs from the disc, splitting them into traces, and ordering event in each trace. The general idea of the second MapReduce all the implementations is that first step of process discovery typically requires extracting trivial dependencies between events called *log-based ordering relations*. Examples of those are:

- a > b — event a is directly followed by event b,
- a >> b — a loop of length two,
- a >>> b — event a is followed by event b somewhere in the log.

These relations can be found individually for each trace. Therefore, their computations are trivially parallelized using Mappers. Further MapReduce jobs vary but they somehow use mined primitive log-ordering relations to build a process model. The main potential problem of implementations is that these further MapReduce jobs typically compute relations for the overall event log. To achieve this, it is often the case when it is necessary for mappers to produce identical keys for all emitted pairs so that they all end up on the same computer and processed by the same reducer. Moreover, the proposed implementations extensively use identity mappers. It is a standard term for mappers, which emit exactly the same key-value pairs as they receive without performing any additional computations — all useful computations performed by combiners or reducers. They are used only because MapReduce paradigm requires presence of mappers. Despite these concerns, it is shown that performance and scalability provided by MapReduce are good enough for the task of process discovery from large volumes of data. Our solution, in

contrast to the described above, uses a more suitable file format. It allows measuring conformance without extra steps needed for preliminary log transformations.

In [32] the authors describe their framework for simplified execution of process mining algorithms on Hadoop clusters. The primarily focus of this work is to show how process mining algorithm can be submitted to a Hadoop cluster via the ProM user interface. In order to demonstrate viability of their approach, the authors claim that they implemented and tested the Alpha miner, the flexible heuristics miner, and the inductive miner [33]. We opted for not using the presented framework in order to simplify the usage of our ProM plugin and not to force the user to download all the codebase required by Hadoop and its ecosystem.

To sum up, these papers clearly demonstrate not only that process mining can benefit from using distributed systems and MapReduce, but also that such distributed process mining algorithms are needed and desired for usage in the real-life environment. Moreover, from these papers it is clear that some common approaches and techniques of process mining suit the MapReduce model well. Last but not least, analysis of the related work reveal that there are only theoretical considerations of parallel or distributed conformance checking and its usefulness.

7. Conclusions

This paper presents one of the possible ways of speeding up large-scale conformance checking. The paper provides a helicopter-view of distributed conformance checking and suggests ways for possible extensions and improvements. One of the proposed algorithms was implemented and evaluated on event logs, which were different in terms of size and fitness.

As a possible extension, it is worth considering implementing the algorithm using the Spark framework rather than Hadoop because as it is often claimed *Spark* might provide better performance due to its in-memory nature. Furthermore, the *XES* standard which defines how event logs should be structured for convenient process mining, but it seems that the XES standard is not the best option for using with Hadoop. Thus, it is possible to consider other storage formats such as Hadoop sequence files or the *Avro* format.

Acknowledgment

This work is supported by the Basic Research Program at the National Research University Higher School of Economics and Russian Foundation for Basic Research, project No. 15- 37-21103.

References

[1]. Wil M. P. van der Aalst, Process mining: discovery, conformance and enhancement of business processes. Springer, 2011. C. Lattner. LLVM: An Infrastructure for Multi-Stage Optimization. Master's thesis, Computer Science Dept., University of Illinois at Urbana-Champaign, Urbana, IL.

[2]. S. A. Shershakov and V. A. Rubin, “System runs analysis with process mining,” Modeling and Analysis of Information Systems, vol. 22, no. 6, pp. 818–833, December 2015.
[3]. S. A. Shershakov, “VTMine framework as applied to process mining modeling,” International Journal of Computer and Communication Engineering, vol. 4, no. 3, pp. 166–179, May 2015.
[4]. W. M. van der Aalst, “Process Mining in the Large: A Tutorial,” in Business Intelligence. Springer, 2014, pp. 33–76.
[5]. C. Bratosin, N. Sidorova, and W. van der Aalst, “Distributed Genetic Process Mining,” in Evolutionary Computation (CEC), 2010 IEEE Congress on, 2010, pp. 1–8.
[6]. S. J. J. Leemans, D. Fahland, and W. M. P. van der Aalst, “Discovering Block-Structured Process Models from Event Logs Containing Infrequent Behaviour,” in Business Process Management Workshops, ser. Lecture Notes in Business Information Processing, N. Lohmann, M. Song, and P. Wohed, Eds. Springer International Publishing, 2014, vol. 171, pp. 66–78.
[7]. A. A. Kalenkova, I. A. Lomazova, and W. M. P. van der Aalst, “Process Model Discovery: A Method Based on Transition System Decomposition,” in Petri Nets, ser. Lecture Notes in Computer Science, vol. 8489. Springer, 2014, pp. 71–90.
[8]. D. Fahland and W. M. P. van der Aalst, “Model Repair - Aligning Process Models to Reality,” Inf. Syst., vol. 47, pp. 220–243, 2015. [Online]. Available: http://dx.doi.org/10.1016/j.is.2013.12.007
[9]. I. S. Shugurov and A. A. Mitsyuk, “Iskra: A Tool for Process Model Repair,” Proceedings of the Institute for System Programming, vol. 27, no. 3, pp. 237–254, 2015.
[10]. J. Dean and S. Ghemawat, “MapReduce: Simplified Data Processing on Large Clusters,” Commun. ACM, vol. 51, no. 1, pp. 107–113, 2008. [Online]. Available: http://doi.acm.org/10.1145/1327452.1327492
[11]. W. M. P. van der Aalst, “Distributed Process Discovery and Conformance Checking,” in Fundamental Approaches to Software Engineering, ser. Lecture Notes in Computer Science, J. de Lara and A. Zisman, Eds. Springer Berlin Heidelberg, 2012, vol. 7212, pp. 1–25.
[12]. A. Adriansyah, “Aligning Observed and Modeled Behavior,” PhD Thesis, Technische Universiteit Eindhoven, Eindhoven, The Netherlands, 2014.
[13]. A. Adriansyah, B. van Dongen, and W. M. van der Aalst, “Conformance Checking using Cost-Based Fitness Analysis,” in IEEE International Enterprise Computing Conference (EDOC 2011), C. Chi and P. Johnson, Eds. IEEE Computer Society, 2011, pp. 55–64.
[14]. D. Miner and A. Shook, MapReduce Design Patterns: Building Effective Algorithms and Analytics for Hadoop and Other Systems, 1st ed. O’Reilly Media, Inc., 2012.
[15]. W. M. P. van der Aalst, “Decomposing Petri Nets for Process Mining: A Generic Approach,” Distributed and Parallel Databases, vol. 31, no. 4, pp. 471–507, 2013.
[16]. J. Munoz-Gama, “Conformance checking and diagnosis in process mining,” PhD Thesis, Universitat Politecnica de Catalunya, 2014.
[17]. W. M. P. van der Aalst, “Decomposing Process Mining Problems Using Passages,” in Application and Theory of Petri Nets, ser. Lecture Notes in Computer Science, S. Haddad and L. Pomello, Eds. Springer Berlin Heidelberg, 2012, vol. 7347, pp. 72–91.
[18]. J. Munoz-Gama, J. Carmona, and W. M. van der Aalst, “Single-Entry Single-Exit Decomposed Conformance Checking,” Information Systems, vol. 46, pp. 102–122, 2014.
[19]. “Apache hadoop,” http://hadoop.apache.org/, accessed: 2016-04-01.

[20]. W. M. P. van der Aalst and. B. van Dongen, C. Gunther, A. Rozinat, ¨ E. Verbeek, and T. Weijters, “ProM: The Process Mining Toolkit,” in Business Process Management Demonstration Track (BPMDemos 2009), ser. CEUR Workshop Proceedings, A. Medeiros and B. Weber, Eds., vol. 489. CEUR-WS.org, 2009, pp. 1–4.
[21]. “Prom framework,” http://www.promtools.org/doku.php, accessed: 2016-04-01.
[22]. IEEE Task Force on Process Mining, “XES Standard Definition,” www.xes-standard.org, 2013.
[23]. “Apache mahout,” http://mahout.apache.org/, accessed: 2016-04-01.
[24]. “Amazon EMR,” https://aws.amazon.com/ru/elasticmapreduce/, accessed: 2016-04-01.
[25]. W. M. P. van der Aalst, A. H. M. ter Hofstede, B. Kiepuszewski, and A. P. Barros, “Workflow Patterns,” Distrib. Parallel Databases, vol. 14, no. 1, pp. 5–51, Jul. 2003. [Online]. Available: http: //dx.doi.org/10.1023/A:1022883727209
[26]. I. S. Shugurov and A. A. Mitsyuk, “Generation of a Set of Event Logs with Noise,” in Proceedings of the 8th Spring/Summer Young Researchers Colloquium on Software Engineering (SYRCoSE 2014), 2014, pp. 88–95.
[27]. H. Reguieg, F. Toumani, H. R. Motahari-Nezhad, and B. Benatallah, “Using MapReduce to Scale Events Correlation Discovery for Business Processes Mining,” in Business Process Management. Springer, 2012, pp. 279–284.
[28]. J. Evermann, “Scalable Process Discovery using Map-Reduce,” IEEE Transactions on Services Computing, vol. PP, no. 99, pp. 1–1, 2014.
[29]. J. Evermann and G. Assadipour, “Big Data meets Process Mining: Implementing the Alpha Algorithm with Map-Reduce,” in Proceedings of the 29th Annual ACM Symposium on Applied Computing. ACM, 2014, pp. 1414–1416.
[30]. W. M. P. van der Aalst, A. J. M. M. Weijters, and L. Maruster, “Workflow Mining: Discovering Process Models from Event Logs,” IEEE Transactions on Knowledge and Data Engineering, vol. 16, no. 9, pp. 1128–1142, 2004.
[31]. A. J. M. M. Weijters and J. T. S. Ribeiro, “Flexible Heuristics Miner (FHM),” in Computational Intelligence and Data Mining (CIDM), 2011 IEEE Symposium on, April 2011, pp. 310–317.
[32]. S. Hernandez, S. Zelst, J. Ezpeleta, and W. M. P. van der Aalst, “Handling big (ger) logs: Connecting ProM 6 to Apache Hadoop,” in Proceedings of the BPM2015 Demo Session, ser. CEUR Workshop Proceedings, vol. 1418, 2015, pp. 80–84.
[33]. S. J. J. Leemans, D. Fahland, and W. M. P. van der Aalst, “Discovering Block-Structured Process Models from Incomplete Event Logs,” in Application and Theory of Petri Nets and Concurrency, ser. Lecture Notes in Computer Science, G. Ciardo and E. Kindler, Eds. Springer.

Применение MapReduce для проверки соответствия моделей процессов и логов событий

И.С. Шугуров <shugurov94@gmail.com>
А.А. Мицюк < amitsyuk@hse.ru >
Национальный Исследовательский Университет «Высшая Школа Экономики», Лаборатория процессно-ориентированных информационных систем, ул. Мясницкая, д. 20, 101000, г. Москва, Россия.

Аннотация. Process mining – это относительно новая область исследований, в рамках которой разрабатываются методы исследования и улучшения бизнес-процессов. Спецификой методов process mining является то, что они основываются на анализе истории выполнения процессов, которая представляется в виде логов событий. Проверка соответствия моделей процессов и логов событий является одним из ключевых направлений в области process mining. Алгоритмы проверки соответствия используются для того, чтобы оценить, насколько хорошо данная модель бизнес-процесса, представленная, например, в виде сети Петри, описывает поведение, записанное в логе событий. Проверка соответствия, базирующаяся на использовании так называемых "выравниваний", на данный момент является самым передовым и часто используемым алгоритмом проверки соответствия. В данной работе рассматривается проблема большой вычислительной сложности данного алгоритма. В настоящее время проверка соответствия на основе выравниваний является не слишком эффективной с точки зрения потребления памяти и времени, необходимого для вычислений. Решение этой проблемы имеет большое значение для успешного применения проверки соответствия между реальными моделями бизнес-процессов и логами событий, что весьма проблематично с использованием существующих подходов. MapReduce является популярной моделью параллельных вычислений, которая упрощает реализацию эффективных и масштабируемых распределенных вычислений. В данной работе представлена модифицированная версия алгоритма проверки соответствия на основе выравниваний с применением MapReduce. Так же в работе показано, что проверка соответствия может быть распределена с помощью MapReduce, и что такое распределение может привести к уменьшению времени, требуемого для вычислений. Показано, что алгоритм проверки соответствия модели процесса и лога событий может быть реализован в распределенном виде с помощью MapReduce. Показано, что время вычисления растет линейно с ростом размера логов событий.

Ключевые слова: process mining; conformance checking; MapReduce; Hadoop; big data.

DOI: 10.15514/ISPRAS-2016-28(3)-7

Для цитирования: Шугуров И.С., Мицюк А.А. Применение MapReduce для проверки соответствия моделей процессов и логов событий. Труды ИСП РАН, том 1, вып. 3, 2016 г. стр. 103-122 (на английском). DOI: 10.15514/ISPRAS-2016-28(3)-7.

Список литературы

[1]. Wil M. P. van der Aalst, Process mining: discovery, conformance and enhancement of business processes. Springer, 2011. C. Lattner. LLVM: An Infrastructure for Multi-Stage Optimization. Master's thesis, Computer Science Dept., University of Illinois at Urbana-Champaign, Urbana, IL.

[2]. S. A. Shershakov and V. A. Rubin, "System runs analysis with process mining," Modeling and Analysis of Information Systems, vol. 22, no. 6, pp. 818–833, December 2015.

[3]. S. A. Shershakov, "VTMine framework as applied to process mining modeling," International Journal of Computer and Communication Engineering, vol. 4, no. 3, pp. 166–179, May 2015.

[4]. W. M. van der Aalst, "Process Mining in the Large: A Tutorial," in Business Intelligence. Springer, 2014, pp. 33–76.

[5]. C. Bratosin, N. Sidorova, and W. van der Aalst, "Distributed Genetic Process Mining," in Evolutionary Computation (CEC), 2010 IEEE Congress on, 2010, pp. 1–8.

[6]. S. J. J. Leemans, D. Fahland, and W. M. P. van der Aalst, "Discovering Block-Structured Process Models from Event Logs Containing Infrequent Behaviour," in Business Process Management Workshops, ser. Lecture Notes in Business Information Processing, N. Lohmann, M. Song, and P. Wohed, Eds. Springer International Publishing, 2014, vol. 171, pp. 66–78.

[7]. A. A. Kalenkova, I. A. Lomazova, and W. M. P. van der Aalst, "Process Model Discovery: A Method Based on Transition System Decomposition," in Petri Nets, ser. Lecture Notes in Computer Science, vol. 8489. Springer, 2014, pp. 71–90.

[8]. D. Fahland and W. M. P. van der Aalst, "Model Repair - Aligning Process Models to Reality," Inf. Syst., vol. 47, pp. 220–243, 2015. [Online]. Доступно по ссылке: http://dx.doi.org/10.1016/j.is.2013.12.007.

[9]. I. S. Shugurov and A. A. Mitsyuk, "Iskra: A Tool for Process Model Repair," Proceedings of the Institute for System Programming, vol. 27, no. 3, pp. 237–254, 2015.

[10]. J. Dean and S. Ghemawat, "MapReduce: Simplified Data Processing on Large Clusters," Commun. ACM, vol. 51, no. 1, pp. 107–113, 2008. [Online]. Доступно по ссылке: http://doi.acm.org/10.1145/1327452.1327492.

[11]. W. M. P. van der Aalst, "Distributed Process Discovery and Conformance Checking," in Fundamental Approaches to Software Engineering, ser. Lecture Notes in Computer Science, J. de Lara and A. Zisman, Eds. Springer Berlin Heidelberg, 2012, vol. 7212, pp. 1–25.

[12]. A. Adriansyah, "Aligning Observed and Modeled Behavior," PhD Thesis, Technische Universiteit Eindhoven, Eindhoven, The Netherlands, 2014.

[13]. A. Adriansyah, B. van Dongen, and W. M. van der Aalst, "Conformance Checking using Cost-Based Fitness Analysis," in IEEE International Enterprise Computing Conference (EDOC 2011), C. Chi and P. Johnson, Eds. IEEE Computer Society, 2011, pp. 55–64.

[14]. D. Miner and A. Shook, MapReduce Design Patterns: Building Effective Algorithms and Analytics for Hadoop and Other Systems, 1st ed. O'Reilly Media, Inc., 2012.

[15]. W. M. P. van der Aalst, "Decomposing Petri Nets for Process Mining: A Generic Approach," Distributed and Parallel Databases, vol. 31, no. 4, pp. 471–507, 2013.

[16]. J. Munoz-Gama, "Conformance checking and diagnosis in process mining," PhD Thesis, Universitat Politecnica de Catalunya, 2014.

[17]. W. M. P. van der Aalst, “Decomposing Process Mining Problems Using Passages,” in Application and Theory of Petri Nets, ser. Lecture Notes in Computer Science, S. Haddad and L. Pomello, Eds. Springer Berlin Heidelberg, 2012, vol. 7347, pp. 72–91.

[18]. J. Munoz-Gama, J. Carmona, and W. M. van der Aalst, “Single-Entry Single-Exit Decomposed Conformance Checking,” Information Systems, vol. 46, pp. 102–122, 2014.

[19]. “Apache hadoop,” доступно по ссылке: http://hadoop.apache.org/, 2016-04-01.

[20]. W. M. P. van der Aalst and. B. van Dongen, C. Gunther, A. Rozinat, ¨ E. Verbeek, and T. Weijters, “ProM: The Process Mining Toolkit,” in Business Process Management Demonstration Track (BPMDemos 2009), ser. CEUR Workshop Proceedings, A. Medeiros and B. Weber, Eds., vol. 489. CEUR-WS.org, 2009, pp. 1–4.

[21]. “Prom framework,” доступно по ссылке: http://www.promtools.org/doku.php, 2016-04-01.

[22]. IEEE Task Force on Process Mining, “XES Standard Definition,” www.xes-standard.org, 2013.

[23]. “Apache mahout,” доступно по ссылке: http://mahout.apache.org/,: 2016-04-01.

[24]. “Amazon EMR,” доступно по ссылке: https://aws.amazon.com/ru/elasticmapreduce/, accessed: 2016-04-01.

[25]. W. M. P. van der Aalst, A. H. M. ter Hofstede, B. Kiepuszewski, and A. P. Barros, “Workflow Patterns,” Distrib. Parallel Databases, vol. 14, no. 1, pp. 5–51, Jul. 2003. [Online]. Доступно по ссылке: http: //dx.doi.org/10.1023/A:1022883727209

[26]. I. S. Shugurov and A. A. Mitsyuk, “Generation of a Set of Event Logs with Noise,” in Proceedings of the 8th Spring/Summer Young Researchers Colloquium on Software Engineering (SYRCoSE 2014), 2014, pp. 88–95.

[27]. H. Reguieg, F. Toumani, H. R. Motahari-Nezhad, and B. Benatallah, “Using MapReduce to Scale Events Correlation Discovery for Business Processes Mining,” in Business Process Management. Springer, 2012, pp. 279–284.

[28]. J. Evermann, “Scalable Process Discovery using Map-Reduce,” IEEE Transactions on Services Computing, vol. PP, no. 99, pp. 1–1, 2014.

[29]. J. Evermann and G. Assadipour, “Big Data meets Process Mining: Implementing the Alpha Algorithm with Map-Reduce,” in Proceedings of the 29th Annual ACM Symposium on Applied Computing. ACM, 2014, pp. 1414–1416.

[30]. W. M. P. van der Aalst, A. J. M. M. Weijters, and L. Maruster, “Workflow Mining: Discovering Process Models from Event Logs,” IEEE Transactions on Knowledge and Data Engineering, vol. 16, no. 9, pp. 1128–1142, 2004.

[31]. A. J. M. M. Weijters and J. T. S. Ribeiro, “Flexible Heuristics Miner (FHM),” in Computational Intelligence and Data Mining (CIDM), 2011 IEEE Symposium on, April 2011, pp. 310–317.

[32]. S. Hernandez, S. Zelst, J. Ezpeleta, and W. M. P. van der Aalst, “Handling big (ger) logs: Connecting ProM 6 to Apache Hadoop,” in Proceedings of the BPM2015 Demo Session, ser. CEUR Workshop Proceedings, vol. 1418, 2015, pp. 80–84.

[33]. S. J. J. Leemans, D. Fahland, and W. M. P. van der Aalst, “Discovering Block-Structured Process Models from Incomplete Event Logs,” in Application and Theory of Petri Nets and Concurrency, ser. Lecture Notes in Computer Science, G. Ciardo and E. Kindler, Eds. Springer.

К синтезу адаптивных проверяющих последовательностей для недетерминированных автоматов

А.Д. Ермаков <antonermak@inbox.ru>
Н.В. Евтушенко <yevtushenko@sibmail.com>
Национальный исследовательский Томский государственный университет, 634050, Россия, г. Томск, пр. Ленина, д. 36.

Аннотация. Существует достаточно много публикаций, посвященных построению проверяющей последовательности для полностью определенного детерминированного конечного автомата. Тем не менее, для недетерминированных автоматов таких публикаций достаточно мало; исследователи начали с того, что предложили алгоритм построения проверяющей последовательности для инициального недетерминированного автомата относительно эквивалентности. В данной работе рассматривается построение адаптивной проверяющей последовательности относительно редукции. Проверяемый автомат есть редукция полностью определенного автомата-спецификации, если для каждой входной последовательности выходная реакция проверяемого автомата содержится в множестве выходных реакций спецификации на эту входную последовательность. В первой части данной статьи мы предполагаем, что полностью определенный возможно недетерминированный автомат-спецификация имеет разделяющую последовательность разумной длины, каждое состояние детерминировано достижимо из любого другого состояния, и проверяемый автомат (автомат-реализация) является полностью определенным и детерминированным. Поведение проверяемого автомата неизвестно; мы знаем только, что его число состояний не больше числа состояний автомата-спецификации. При описанных выше условиях проверяемый автомат является редукцией спецификации, если и только если проверяемый автомат изоморфен подавтомату автомата-спецификации. Таким образом, необходимо адаптивно построить проверяющую последовательность, проходящую по каждому переходу проверяемого автомата, и проверить конечное состояние перехода посредством разделяющей последовательности. Во второй части статьи мы предлагаем использовать вместо разделяющей последовательности (адаптивный) различающий тестовый пример и на простом примере иллюстрируем, как такая замена может сократить длину адаптивной проверяющей последовательности. Длина тестового примера обычно короче разделяющей последовательности, и вообще говоря, различающий тестовый пример может существовать для автомата-спецификации, не имеющего разделяющей последовательности. В третьей части статьи мы обсуждаем возможность применения

предлагаемой методики построения проверяющей последовательности для частичных возможно недетерминированных автоматов, выделяя наибольший полностью определенный подавтомат. Если такой подавтомат существует, обладает разделяющей последовательностью или различающим тестовым примером, то его можно использовать для построения адаптивной различающей последовательности для исходного частичного, возможно недетерминированного автомата. Тем не менее, следует отметить, что в последнем случае проверяющая последовательность строится относительно не относительно редукции, а относительно отношения квази редукции.

Ключевые слова: недетерминированный конечный автомат; отношение редукции; модель неисправности; адаптивная проверяющая последовательность**.**

DOI: 10.15514/ISPRAS-2016-28(3)-8

Для цитирования: Ермаков А.Д., Евтушенко Н.В. К синтезу адаптивных проверяющих последовательностей для недетерминированных автоматов. Труды ИСП РАН, том 28, вып. 3, 2016 г. стр. 123-144. DOI: 10.15514/ISPRAS-2016-28(3)-8.

1. Введение

Тестирование на основе конечных автоматов широко используется при построении различных тестовых последовательностей для интерактивных дискретных систем [1]; хорошим примером может служить синтез проверяющих тестов для телекоммуникационных протоколов на основе конечных автоматов [2].

В большинстве публикаций предполагается, что спецификация представлена в виде инициального автомата, а проверяющий тест есть множество входных последовательностей, объединяемых посредством надежного сигнала сброса [3]. Если такой сигнал сброса является слишком дорогим, то вместо множества тестовых последовательностей используются так называемые проверяющие последовательности [4, 5]. Для детерминированных автоматов такие последовательности можно построить, когда автоматная спецификация имеет синхронизирующую последовательность, переводящую автомат из любого состояния в одно и то же состояние, и диагностическую последовательность, способную различить любые два различных состояния автомата [4]. Для недетерминированных автоматов количество публикаций на тему синтеза проверяющих последовательностей значительно меньше, тогда как недетерминированные спецификации возникают в различных компьютерных приложениях (программах, системах) [5]. Одной из причин появления недетерминированных спецификаций является опциональность, сопутствующая, например, RFC описаниям многих телекоммуникационных протоколов [6].

В работе [5] авторы предлагают метод построения проверяющей последовательности для полностью определенного недетерминированного автомата относительно эквивалентности при наличии соответствующих ограничений на конечно автоматную спецификацию и область неисправности. В работе [7] авторы обобщают результаты для построения проверяющей последовательности относительно редукции. Как обычно, проверяющая последовательность называется адаптивной, если следующий входной символ зависит от выходных символов, полученных от проверяемой системы в ответ на приложенные ранее входные символы. Еще один метод для построения адаптивной проверяющей последовательности относительно редукции предлагается в [8].

В данной работе мы ослабляем некоторые ограничения работы [8] и предлагаем использовать при построении адаптивной проверяющей последовательности вместо разделяющей последовательности (адаптивный) тестовый пример (адаптивную различающую последовательность). Длина адаптивного различающего тестового примера может быть меньше, чем длина разделяющей последовательности [9,10]; более того, различающий тестовый пример может существовать и при отсутствии разделяющей последовательности.

В первой части данной статьи мы предполагаем, что полностью определенный возможно недетерминированный автомат-спецификация имеет разделяющую последовательность разумной длины, каждое состояние детерминировано достижимо из любого другого состояния, и проверяемый автомат (автомат-реализация) является полностью определенным и детерминированным. Более того, поведение проверяемого автомата неизвестно; мы знаем только, что его число состояний не больше числа состояний автомата-спецификации. Проверяемый автомат есть редукция спецификации, если для каждой входной последовательности выходная реакция проверяемого автомата содержится в множестве выходных реакций автомата-спецификации на эту входную последовательность. При описанных выше условиях проверяемый автомат является редукцией спецификации, если и только если проверяемый автомат изоморфен подавтомату автомата-спецификации [8]. Таким образом, вместо того, чтобы проверять все входные последовательности, достаточно установить взаимно однозначное соответствие между состояниями и переходами автомата-спецификации и проверяемого автомата [4]. Другими словами, при построении проверяющей последовательности нужно «пройти» по каждому переходу проверяемого автомата и проверить конечное состояние перехода посредством разделяющей последовательности. Такой подход позволяет построить проверяющую последовательность разумной длины, если разделяющая и передаточные последовательности имеют полиномиальную длину относительно числа состояний автомата-спецификации. Мы кратко описываем, как построить адаптивную проверяющую последовательность при таком подходе.

Во второй части статьи мы предлагаем использовать вместо разделяющей последовательности (адаптивный) различающий тестовый пример [11,12] и на простом примере иллюстрируем, как такая замена может сократить длину адаптивной проверяющей последовательности. Мы также отмечаем, что различающий тестовый пример может существовать для автомата-спецификации, не имеющего разделяющей последовательности.

В третьей части статьи мы обсуждаем возможность применения методики построения проверяющей последовательности относительно редукции для частичных возможно недетерминированных автоматов, выделяя наибольший полностью определенный подавтомат. Если данный подавтомат существует, обладает разделяющей последовательностью или различающим тестовым примером и является детерминировано связным, то установление взаимно однозначного соответствия между состояниями проверяемого автомата и автомата-спецификации возможно точно так же как для полностью определенного автомата.

Структура статьи следующая. Раздел 2 содержит основные определения и обозначения. Раздел 3 описывает адаптивный подход к построению проверяющей последовательности при использовании разделяющей последовательности. Раздел 4 описывает изменения в адаптивном подходе при использовании различающего тестового примера вместо разделяющей последовательности. В разделе 5 мы обсуждаем возможности использования предложенного подхода к построению адаптивной проверяющей последовательности для частичных, возможно недетерминированных автоматов.

2. Определения

Конечным автоматом, или просто *автоматом* называется четверка $S = \langle S, I, O, h_S \rangle$, где S - конечное непустое множество состояний, I и O - конечные входной и выходной алфавиты, и $h_S \subseteq S \times I \times O \times S$ есть отношение переходов. Автомат S *недетерминированный*, если существует хотя бы одна пара $(s, i) \in S \times I$, для которой существуют несколько пар $(o, s') \in O \times S$ таких, что $(s, i, o, s') \in h_S$. Автомат S называется *полностью определенным*, если для каждой пары $(s, i) \in S \times I$ существует $(o, s') \in O \times S$ такое, что $(s, i, o, s') \in h_S$; в противном случае автомат S называется *частично определенным* или *частичным.* Автомат S *наблюдаемый*, если для каждой пары переходов (s, i, o, s_1), $(s, i, o, s_2) \in h_S$ имеет место $s_1 = s_2$. Автомат S *инициальный*, если существует выделенное начальное состояние s_1 (обозначение: S/s_1). Таким образом, инициальный автомат есть пятерка $\langle S, I, O, h, s_1 \rangle$. Для автоматов $S = \langle S, I, O, h_S, s_1 \rangle$ и $T = \langle T, I, O, h_T, t_1 \rangle$ автомат T есть подавтомат S, если $T \subseteq S$, $t_1 = s_1$ и $h_T \subseteq h_S$. В дальнейшем, мы рассматриваем наблюдаемые и полностью определенные автоматы, если явно не указано иное.

В качестве примера рассмотрим автомат на рис. 1 с множеством состояний $S = \{1, 2, 3\}$ и начальным состоянием 1. Множество $I = \{i_1, i_2\}$ есть множество входных символов автомата, $O = \{0, 1, 2\}$ есть множество входных символов. Автомат является недетерминированным. Например, из состояния 2 под действием входного символа i_2 есть переходы в состояния 2 и 3.

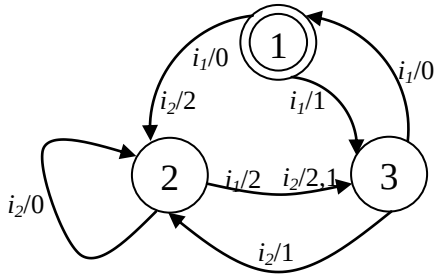


Рис. 1. Автомат S.

Fig. 1. FSM S.

Последовательность входо-выходных пар переходов, начинающаяся в состоянии s автомата $S = \langle S, I, O, h \rangle$, называется *входо-выходной последовательностью* или *трассой* автомата S в состоянии s. Множество всех трасс автомата S в состоянии s, включая пустую последовательность, обозначается $Tr(S/s)$. Обычным образом, отношение переходов автомата распространяется на входные и выходные последовательности. Четверка (s, α, β, s'), в которой s и s' состояния автомата, α и β - входная и выходная последовательности, принадлежит отношению переходов h_S автомата S, если пара последовательностей α и β (обозначение α/β) является трассой в состоянии s, переводящей автомат из состояния s в состояние s'.

Обозначение $successor(s, \alpha)$ используется для обозначения всех состояний, достижимых из состояния s после подачи входной последовательности α, т.е. $successor(s, \alpha) = \{s' : \exists \beta \in O^* [(s, \alpha, \beta, s') \in h_S]\}$.

Множество $out(s, \alpha)$ обозначает множество всех выходных последовательностей (реакций), которые автомат S может выдать в состоянии s после подачи входной последовательности α, т.е., $out(s, \alpha) = \{\beta : \exists s' \in S [(s, \alpha, \beta, s') \in h]\}$. Для автомата на рис. 1 $successor(2, i_2\, i_1) = \{1, 3\}$, а $out(2, i_2\, i_1) = \{02, 20, 10\}$.

Автомат S *детерминировано связный* (*д-связный*), если для каждой пары состояний $s, s' \in S$ существует входная последовательность α такая, что $successor(s, \alpha) = \{s'\}$. В этом случае мы говорим, что s' *детерминировано* (*д-*)*достижимо* из состояния s. Такая *д*-передаточная входная

последовательность обозначается $\alpha_{ss'}$. Инициальный автомат *связный*, если для каждого $s \in S$ существует входная последовательность, которая переводит автомат S из начального состояния в состояние s. Любой ∂-связный автомат является связным по определению.

Пусть $S = \langle S, I, O, h_S, s_1 \rangle$ и $P = \langle P, I, O, h_P, p_1 \rangle$ инициальные автоматы; *пересечением* автоматов $S \cap P$ называется наибольший связный подавтомат автомата $\langle S \times P, I, O, f, s_1 p_1 \rangle$, где $(sp, i, o, s'p') \in f \Leftrightarrow (s, i, o, s') \in h_S \,\&\, (p, i, o, p') \in h_P$. В полностью определенном автомате $S = \langle S, I, O, h_S \rangle$ состояния s_1 и s_2 называются *неразделимыми*, если для каждой входной последовательности $\alpha \in I^*$ имеет место $out(s_1, \alpha) \cap out(s_2, \alpha) \neq \varnothing$, т.е. множества выходных реакций автомата в состояниях s_1 и s_2 на каждую входную последовательность пересекаются; в противном случае, состояния s_1 и s_2 *разделимы*. Для разделимых состояний s_1 и s_2 существует входная последовательность $\alpha \in I^*$ такая, что $out(s_1, \alpha) \cap out(s_2, \alpha) = \varnothing$, т.е. множества выходных реакций в состояниях s_1 и s_2 на входную последовательность α не пересекаются. Такая входная последовательность α называется *разделяющей* последовательностью для состояний s_1 и s_2 (обозначение: $s_1 \not\sim_\alpha s_2$).

В качестве примера рассмотрим состояния 1 и 2 автомата на рис. 1. Множества выходных последовательностей автомата S в состояниях 1 и 2 на входную последовательность $i_2\, i_1$ суть непересекающиеся множества {22} и {02, 20, 10}; таким образом, последовательность $i_2\, i_1$ является разделяющей для этих двух состояний.

Если последовательность α разделяет каждую пару состояний автомата S, то она является *разделяющей* последовательностью для автомата S. Для автомата S на рис. 1 последовательность $i_2\, i_1$ разделяет каждую пару состояний автомата, поскольку множество выходных реакций автомата S на разделяющую последовательность $i_2\, i_1$ в состоянии 3 есть множество {12}. Методы проверки существования разделяющей последовательности и методы её построения можно найти в [13]. К сожалению, разделяющая последовательность существует не для всякого полностью определенного автомата. Кроме того, известно, что длина разделяющей последовательности может быть экспоненциальной относительно числа состояний автомата S. Тем не менее, в [14] отмечается, что согласно экспериментам со случайно сгенерированными автоматами, если разделяющая последовательность для автомата существует, то она довольно короткая.

Пусть S и P полностью определенные автоматы. Состояние p автомата P называется *редукцией* состояния s автомата S (обозначение $p \leq s$), если множество трасс автомата P в состоянии p является подмножеством трасс автомата S в состоянии s; в противном случае, p не является редукцией s. Инициальный автомат P/p_1 является редукцией инициального автомата S/s_1, если $p_1 \leq s_1$, т.е. если множество трасс P/p_1 является подмножеством трасс

S/s_1. Если множества трасс S/s_1 и P/p_1 совпадают, тогда эти автоматы *эквивалентны* [11].

Модель неисправности. При синтезе проверяющих тестов на основе автоматной модели предполагается, что автомат-спецификация описывает эталонное поведение, тогда как область неисправности содержит автоматное описание для каждой возможной реализации спецификации. В данной работе мы предполагаем, что все автоматы инициальные, полностью определенные и наблюдаемые; более того, предполагается, что автомат-реализация (проверяемый автомат) детерминированный и имеет не больше состояний, чем автомат-спецификация. В качестве отношения соответствия (конформности) мы рассматриваем отношение редукции. Другими словами, мы неявно предполагаем, что недетерминизм спецификации вытекает из ее опциональности, где разработчик выбирает лучший вариант в соответствии с некоторыми критериями. В такой системе по-прежнему имеется надежный сигнал сброса, который является довольно дорогим и может быть использован только один раз перед подачей проверяющей последовательности.

Таким образом, мы рассматриваем модель неисправностей $FM = <S/s_1, \leq, \Omega>$, где S/s_1 полностью определенный возможно недетерминированный наблюдаемый инициальный автомат с n состояниями, $n > 1$, Ω – множество всех полностью определенных детерминированных автоматов, определенных на том же входном алфавите, не более чем с n состояниями.

Под *адаптивной стратегией* при тестировании предъявленного (проверяемого) автомата из области неисправности понимается построение входной последовательности, в которой следующий входной символ, подаваемый на проверяемый автомат, вычисляется на основе выходных реакций этого автомата на предыдущие входные символы. Адаптивная стратегия называется *исчерпывающей* относительно модели неисправности FM, если для каждого $P/p_1 \in \Omega$ выходная последовательность на построенную входную последовательность содержится в множестве выходных реакций автомата-спецификации, если и только если P/p_1 есть редукция S/s_1. Соответственно построенная входная последовательность часто называется адаптивной *проверяющей* последовательностью, поскольку гарантировано проверяет только предъявленный к тестированию автомат. Следующие утверждения могут быть полезны при доказательстве того, что адаптивная стратегия является исчерпывающей относительно данной модели неисправностей.

Утверждение 1 [11]. Для полностью определенных наблюдаемых связных автоматов S/s_1 и P/p_1, автомат P/p_1 есть редукция автомата S/s_1, если и только если пересечение $P/p_1 \cap S/s_1$ есть полностью определенный автомат.

Утверждение 2. Пусть полностью определенный наблюдаемый связный автомат S/s_1 обладает разделяющей последовательностью. Если автомат P/p_1 есть редукция S/s_1, то разделяющая последовательность различает каждую пару состояний автомата P/p_1.

Утверждение 3 [15]. Пусть P/p_1 и S/s_1- полностью определенные наблюдаемые связные автоматы, причем автомат S/s_1 обладает разделяющей последовательностью и является ∂-связным. Автомат P/p_1 есть редукция S/s_1, если и только если P/p_1 изоморфен подавтомату S/s_1.

Утверждения 2 и 3 показывают, какой должна быть адаптивная стратегия. Проверяющая последовательность должна пройти по каждому переходу проверяемого автомата, и финальное состояние перехода необходимо проверить посредством разделяющей последовательности. Соответственно, процедура построения проверяющей последовательности разделяется на два этапа. На первом этапе проверяется, что предъявленный к проверке автомат имеет n состояний, и в каждом состоянии проверяемого автомата фиксируются реакция автомата на разделяющую последовательность и соответствующее состояние-преемник. На втором этапе проверяется каждый переход проверяемого автомата.

Пример 1. Рассмотрим автомат-спецификацию на рис. 1. Автомат обладает разделяющей последовательностью $i_2\, i_1$, и в табл. 1 представлены выходные реакции на эту последовательность в каждом состоянии.

Табл. 1. Выходные реакции на последовательность I_2I_1.

Table 1. Output responses to I_2I_1.

Состояния	Выходные реакции для разделяющей последовательности: i_2i_1
1	22
2	02, 20, 10
3	12

Отметим, что автомат на рис. 1 является ∂-связным, т.е. для любой пары различных состояний j и k существует ∂-передаточная последовательность α_{jk}: $\alpha_{12} = i_2$, $\alpha_{23} = i_1$, и $\alpha_{31} = i_1$.

3. Адаптивная стратегия построения проверяющей последовательности

В этом разделе мы кратко повторяем некоторые шаги из [8], которые служат основой при построении адаптивной проверяющей последовательности.

Вход. Автомат $S = (S, I, O, h_S, s_1)$ с n состояниями, разделяющая последовательность δ для автомата S, ∂-передаточные последовательности $\alpha_{ss'}$ для каждой пары различных состояний s и s', полностью определенный детерминированный проверяемый автомат P/p_1 не более чем с n состояниями, структура переходов которого не известна.

Выход. Сообщение 'P/p_1 является редукцией S/s_1' или 'P/p_1 не является редукцией S/s_1' и входная последовательность σ, которая отличает автомат P/p_1 от S/s_1 в последнем случае.

Как было сказано выше, процедура состоит из двух этапов. На первом шаге проверяется реакция на разделяющую последовательность δ в каждом

состоянии проверяемого автомата, т.е. устанавливается взаимно однозначное соответствие между состояниями автоматов P/p_1 и S/s_1, если P/p_1 есть редукция S/s_1 (утверждение 2). Выходные реакции на разделяющую последовательность и соответствующие δ-преемники сохраняются в множестве *Separable*.

Таким образом, множество *Separate* содержит тройки (s, ρ, s'), где s есть текущее состояние автомата S, $\rho = out_P(s, \delta)$ есть выходная реакция автомата P на разделяющую последовательность δ в состоянии, которое соответствует состоянию s и s' есть δ/ρ-преемник состояния s. Процедура построения множества *Separate* достаточно простая: разделяющая последовательность подается на проверяемый автомат, начиная с начального состояния, до тех пор, пока некоторая выходная реакция автомата не будет получена дважды. Если хотя бы на одном входном символе выходная реакция проверяемого автомата не содержится в ожидаемом множестве реакций, то выдается сообщение 'P/p_1 не является редукцией S/s_1' и входная последовательность, которая отличает автомат P/p_1 от S/s_1. Если на все входные символы получена ожидаемая выходная реакция, то по построенному отрезку проверяющей последовательности заполняется множество *Separate*. Пусть финальное состояние рассмотренного отрезка проверяющей последовательности соответствует состоянию s автомата S/s_1. Если в множестве *Separate* отсутствует реакция проверяемого автомата на разделяющую последовательность в некотором состоянии s', то на проверяемый автомат подается передаточная последовательность $\alpha_{ss'}$, и процесс построения передаточной последовательности продолжается с состояния s'. Выполнение первого шага заканчивается, когда выдано сообщение 'P/p_1 не является редукцией S/s_1', или множество *Separate* содержит реакцию проверяемого автомата на разделяющую последовательность в каждом состоянии. В последнем случае нами установлено взаимно однозначное соответствие между состояниями автоматов P/p_1 и S/s_1.

Пример 2. Пусть автомат S на рис. 1 есть автомат-спецификация; в качестве проверяемого автомата рассмотрим автомат P на рис. 2 с начальным состоянием a.

Последовательность $\delta = i_2\, i_1$ является разделяющей последовательностью для автомата S. В начальном состоянии проверяемого автомата мы получаем выходную реакцию 22 на поданную последовательность δ, и таким образом, начальное состояние a проверяемого автомата соответствует состоянию s_1 спецификации S. После повторной подачи последовательности δ будет получена выходная реакция 12, т.е. состояние 3 автомата S, достигнутое после трассы δ/12, соответствует состоянию c проверяемого автомата. Подаем δ еще раз и получаем уже полученную выходную реакцию 12, т.е. мы снова достигаем состояния, которое соответствует состоянию 3. Таким образом, мы можем внести два элемента (1, 22, 3), (3, 12, 3) в множество *Separate* и

отметить, что текущим состояние проверяемого автомата является состояние, соответствующее состоянию 3. После подачи $\alpha_{32} = i_2$ должно быть достигнуто состояние, соответствующее состоянию 2, и после подачи $\delta = i_2\, i_1$ дважды мы заключаем, что проверяемый автомат под действием последовательности δ переходит из состояния *b*, соответствующего состоянию 2, в состояние *a*, соответствующее состоянию 1. Поскольку *P* имеет не более трех состояний, то все его состояния идентифицированы посредством последовательности δ, и δ различает два любые различные состояния проверяемого автомата. Таким образом, *Separate* = {(1, 22, 3), (2, 10, 1), (3, 12, 3)}; в этом множестве в каждой тройке первый элемент обозначает текущее состояние, второй элемент есть выходная реакция в данном состоянии на последовательность δ, и последний элемент показывает δ-преемник текущего состояния. Иными словами, нами установлено взаимно однозначное соответствие между состояниями автоматов *S* и *P*: 1 и *a*, 2 и *b*, 3 и *c*.

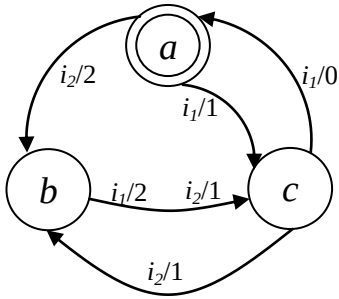


Рис. 2. Проверяемый автомат-реализация P.

Fig. 2. FSM P/p_1 under test.

На следующем этапе строится множество *Transition*. Это множество содержит четверки (*s, i, o, s'*), и его построение заканчивается, как только для каждой пары $(s, i) \in S \times I$ множество *Transition* содержит четверку с такими первыми элементами. Для достижения этой цели в каждом состоянии проверяемого автомата подается каждый входной символ, после которого подается разделяющая последовательность. Если в текущем состоянии проверены переходы по всем входным символам, то используется последовательность уже проверенных переходов из множества *Transition*, чтобы достичь состояния, в котором есть непроверенный переход. Как показано в [8], такая последовательность всегда существует в силу свойств автомата-спецификации. Если хотя бы на одном входном символе выходная реакция проверяемого автомата не содержится в ожидаемом множестве реакций, то выдается сообщение 'P/p_1 не является редукцией S/s_1' и входная последовательность, которая отличает автомат P/p_1 от S/s_1. Если на все входные символы получена ожидаемая выходная реакция, то вторая часть

проверяющей последовательности, построенной для проверки переходов проверяемого автомата, устанавливает взаимно однозначное соответствие между переходами проверяемого автомата и некоторым подавтоматом автомата-спецификации (утверждение 3), т.е. предъявленный автомат является редукцией спецификации.

Пример 3. В табл. 2 представлено множество, полученное на первом этапе построения адаптивной проверяющей последовательности.

Табл. 2. Множество Separate, построенное для автомата P.

Table 2. The set Separate for FSM P.

s	ρ	s'
a(1)	22	c(3)
b(2)	10	a(1)
c(3)	12	c(3)

Перед вторым этапом (проверки переходов) у нас уже есть проверяющая последовательность $\sigma = i_2i_1\ i_2i_1i_2i_1\ i_2\ i_2i_1i_2i_1$, которая заканчивается в состоянии c, соответствующем состоянию 3 автомата S.

На начальном шаге второго этапа множество *Transition* пусто, и существует непроверенный переход из состояния 3 под действием входного символа i_1. После подачи i_1 проверяемый автомат выдает ожидаемый входной символ 1, и после подачи δ мы получаем 12, что соответствует третьей строке табл. 2. Соответственно мы заключаем, что проверяемый автомат перешел в состояние c, соответствующее состоянию 3 автомата S, и заносим четверку (3, i_1, 1, 3) в множество *Transition*. В состоянии 3 есть еще один непроверенный переход под действием входного символа i_2. После подачи i_2 и разделяющей последовательности δ проверяемый автомат выдает ожидаемые выходные реакции 1 и 22, что соответствует третьей строке множества *Separate*, и мы добавляем четверку (3, i_2, 1, 1) в множество *Transition*. Аналогичным образом проверяются остальные переходы автомата P. В результате получается проверяющая последовательность $\sigma = i_2i_1i_2i_1i_2i_1i_2i_2i_1i_2i_1 + i_1i_2i_1\ i_2i_2i_1i_1i_2i_1i_1i_2i_2i_1i_2i_1i_2i_1i_2i_2i_1$, для которой выходная реакция проверяемого автомата содержится в множестве реакций автомата-спецификации на эту последовательность. Соответственно, мы можем заключить, что предъявленный для проверки автомат на рис. 2 есть редукция автомата-спецификации на рис. 1.

4. Использование (адаптивных) различающих тестовых примеров вместо разделяющей последовательности

Для адаптивной стратегии, описанной в разделе 3, существуют достаточно жесткие ограничения на автомат-спецификацию. Такой автомат должен обладать различающей последовательностью и быть d-связным. Известно [11], что не каждый автомат обладает этими свойствами; более того, длина таких

последовательностей (если существуют) может оказаться экспоненциальной относительно числа состояний автомата.

Поскольку мы используем адаптивную стратегию для построения проверяющей последовательности, то имеет смысл заменить разделяющую последовательность так называемым тестовым примером, который представляет адаптивный эксперимент с проверяемым автоматом [13]. Во-первых, известно, что различающий тестовый пример может существовать и при отсутствии разделяющей последовательности, и, во-вторых, длина такого тестового примера во многих случаях существенно меньше [9].

Для входного и выходного алфавитов I и O *тестовый пример* есть связный наблюдаемый инициальный автомат $P = (P, I, O, h_P, p_0)$, граф переходов которого ациклический и в каждом не тупиковым состоянии определены переходы только по одному входному символу со всеми возможными выходными символами. По определению, если $|I| > 1$, то тестовый пример является частичным автоматом. Длина тестового примера определяется как длина самого длинного пути из начального в тупиковое состояние. Вообще говоря, длиной тестового примера является длина самой длинной входной последовательности, которая подается на автомат в процессе адаптивного эксперимента (иногда ее называют высотой адаптивного эксперимента). Как обычно, при тестировании мы бы хотели использовать тестовые примеры минимальной длины. Если известно, что автомат-спецификация является полностью определенным наблюдаемым автоматом без слияний, т.е. для каждых входного символа i и выходного символа o непустые преемники двух различных состояний по входо-выходной паре io не совпадают, то длина различающего тестового примера (если таковой существует) является полиномиальной относительно числа состояний автомата, более точно имеет порядок $O(n^3)$ [16]. Класс автоматов без слияний достаточно большой, по крайней мере, он содержит достаточно много детерминированных автоматов, которые используются в различных приложениях [17].

Пусть S есть полностью определенный и наблюдаемый автомат с входным и выходным алфавитами I и O. Тестовый пример, определенный относительно этих алфавитов, называется *различающим*, если для каждой трассы, переводящей тестовый пример из начального в тупиковое состояние, справедливо, что данная трасса является трассой автомата S только в одном состоянии. Трасса, переводящая тестовый пример в тупиковое состояние, называется *полной* трассой тестового примера. Множество полных трасс тестового примера TC обозначается как $Complete(TC)$.

Пример 4. В качестве примера рассмотрим автомат-спецификацию на рис. 3 из [12] с начальным состоянием 1. Непосредственной проверкой можно убедиться, что автомат не имеет разделяющей последовательности; однако состояния автомата попарно $д$-достижимы и существует различающий тестовый пример (рис. 4). Тупиковые состояния тестового примера помечены соответствующими начальными состояниями. Поскольку проверяющая

последовательность строится адаптивно, то можно использовать различающий пример вместо разделяющей последовательности. Единственное различие с множеством *Separate* из табл. 2 будет в том, что вместо единственной различающей последовательности в проверяемом автомате мы будем фиксировать соответствующие идентификаторы состояний.

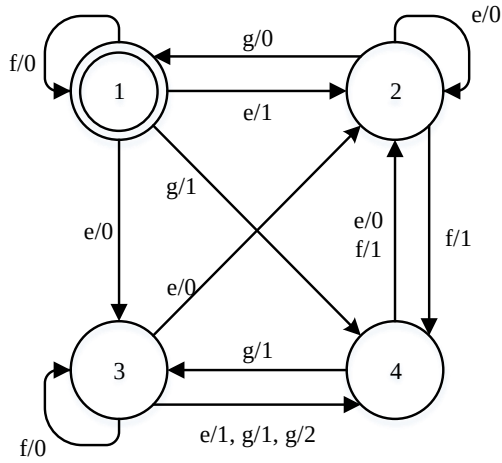


Рис. 3. Автомат-спецификация S.

Fig. 3. Specification FSM S.

Отметим также, что автомат на рис. 3 является *д*-связным; для каждой пары *j* и *k* различных состояний существует *д*-передаточная последовательность α_{jk}: $\alpha_{12} = ge$, $\alpha_{13} = gg$, $\alpha_{14} = g$, $\alpha_{21} = g$, $\alpha_{23} = fg$, $\alpha_{24} = f$, $\alpha_{31} = gfg$, $\alpha_{32} = gf$, $\alpha_{34} = g$, $\alpha_{41} = eg$, $\alpha_{42} = e$, $\alpha_{43} = g$.

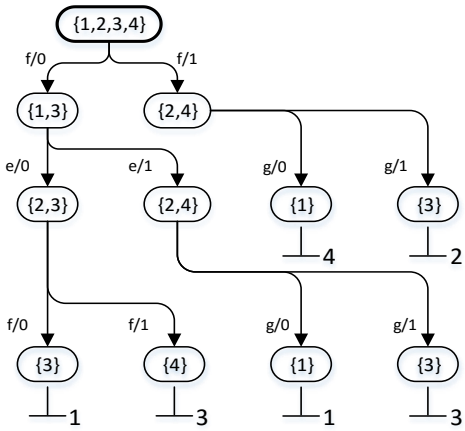


Рис. 4. Различающий тестовый пример T для автомата-спецификации S на рис. 3.

Fig. 4. A distinguishing test case T for the FSM S in Fig. 3.

Для полностью определенного наблюдаемого автомата S и состояния s входная последовательность α есть *идентификатор состояния* s, если α есть разделяющая последовательность для любой пары состояний (s, s'), $s' \neq s$. Последовательности *fef* и *feg* суть идентификаторы состояний 1 and 3 автомата S на рис. 3; для состояний 2 и 4 последовательность *fg* является идентификатором. Пусть автомат-спецификация S/s_1 имеет различающий тестовый пример TC, и P/p_1 есть детерминированный полностью определенный автомат с тем же числом состояний. Автомат P/p_1 называется TC-совместимым с S/s_1, если существует взаимно однозначное соответствие F: $S \rightarrow P$, такое что для каждого состояния $s \in S$ пересечение $Tr(S/s_1) \cap Tr(P/p_1) \cap Complete(TC)$ не пусто, если и только если $p = F(s)$.

Теорема 4. Пусть S/s_1 – полностью определенный наблюдаемый автомат-спецификация, для которого существует различающий тестовый пример TC, и P/p_1 полностью определенный детерминированный автомат, который является TC-совместимым с S/s_1. Для каждого состояния p автомата P/p_1 различающий тестовый пример TC содержит полную трассу α/β, которая является трассой в состоянии p; более того, α есть идентификатор состояния p в P/p_1.

Действительно, если существует взаимно однозначное соответствие между состояниями S/s_1 и P/p_1 согласно различающему тестовому примеру TC, то для любых двух состояний s и s', $s' \neq s$, существует начальный отрезок некоторой полной трассы в тестовом примере TC такой что выходные реакции на этот отрезок в состояниях $p = F(s)$ и $p' = F(s')$ различны. Поскольку TC является различающим тестовым примером для S/s_1, и P/p_1 есть полностью

определенный детерминированный автомата, это означает что входная проекция трассы α/β является идентификатором состояния *p*.

Согласно теореме 4, адаптивную проверяющую последовательность можно построить с использованием различающего тестового примера вместо разделяющей последовательности. Единственное различие состоит в том, что множество *Separate* вместо реакции на различающую последовательность содержит соответствующий идентификатор состояния, выходную реакцию и следующее состояние.

Пример 5. Пусть проверяемым автоматом является автомат P/p_1 на рис. 5 с начальным состоянием *A*, Непосредственной проверкой можно убедиться, что P/p_1 изоморфен подавтомату автомата S/s_1 (рис. 3), т.е. P/p_1 есть редукция автомата S/s_1. Поскольку при тестировании автомат P/p_1 является неизвестным, мы используем адаптивную стратегию для проверки, является ли P/p_1 редукцией S/s_1 (рис. 3). Начиная с начального состояния *A*, мы подаем различающий тестовый пример, подача которого в нашем случае заканчивается подачей входной последовательности *feg*, и эта последовательность является идентификатором состояния 1 автомата *S* (согласно различающему тестовому примеру на рис. 4). Проверяемый автомат выдает реакцию 010, и мы заключаем, что начальное состояние *A* автомата *P* соответствует состоянию 1 автомата *S*.

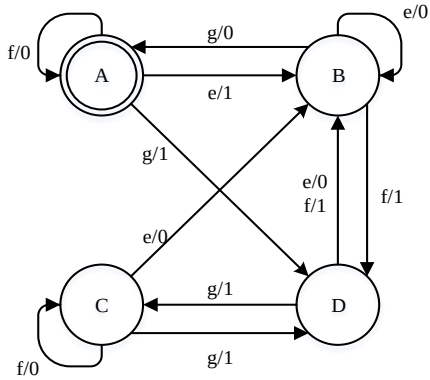


Рис. 5. Автомат P/p_1, для которого строится адаптивная проверяющая последовательность

Fig. 5. FSM P/p_1 under test

Для того чтобы определить, в какое состояние переходит проверяемый автомат под действием входной последовательности *feg*, мы подаем различающий тестовый пример еще раз. В нашем случае это означает, что на проверяемый автомат еще раз подается последовательность *feg*; в качестве выходной реакции мы получаем 010 и заключаем, что *feg* переводит

проверяемый автомат в состояние *A*, которое соответствует состоянию 1 автомата *S*.

Далее мы подаем *д*-передаточную последовательность $\alpha_{12} = ge$, проверяемый автомат выдает 10 и переходит в новое состояние *B*, которое соответствует состоянию 2 автомата *S*. Для того чтобы убедиться в этом, мы подаем идентификатор *fg* состояния 2 из тестового примера на рис. 4 и получаем выходную реакцию 11, т.е. состояние *B* автомата *P* соответствует состоянию 2 автомата *S*. Проверяемый автомат переходит в состояние *C* , которое должно соответствовать состоянию 3 автомата *S*. Мы подаем входную последовательность *fef*, получаем выходную реакцию 001, и проверяемый автомат переходит в новое состояние *D*, которое должно соответствовать состоянию 4 автомата *S*. Чтобы убедиться в этом, мы подаем идентификатор состояния *fg*, получаем выходную реакцию 10, и проверяемый автомат переходит в начальное состояние *A*. Подача входной последовательности *feg*, которая является идентификатором состояния *A*, завершает процедуру, результат которой приведен в табл. 3. Поскольку проверяемый автомат имеет не более четырех состояний, эта таблица содержит идентификатор для каждого состояния, ожидаемую выходную реакцию и следующее ожидаемое состояние.

Табл. 3. Множество Separate для автомата на рис. 3, соответствующее тестовому примеру на рис. 4.

Table. 3. The set Separate for the FSM in Fig. 3 according to the test case in Fig. 4

Текущее состояние	Идентификатор состояния	Выходная реакция	Следующее состояние
A (1)	*feg*	010	A (1)
B (2)	*fg*	11	C (3)
C (3)	*fef*	001	D (4)
D (4)	*fg*	10	A (1)

Множество *Transition* строится точно так же как при использовании разделяющей последовательности, т.е. после подачи каждого входного символа в каждом состоянии следующее состояние верифицируется посредством подачи соответствующего идентификатора. Если в текущем состоянии все входные символы проверены, то, используя уже построенную часть множества *Transition*, автомат переводится в состояние, в котором есть непроверенные входные символы. Для автомата-спецификации на рис. 3 и проверяемого автомата на рис. 5 мы получили адаптивную проверяющую последовательность длины 68, и поскольку были получены только ожидаемые выходные реакции, мы сделали вывод, что проверяемый автомат является редукцией спецификации.

Таким образом, справедливо следующее утверждение.

Теорема 5. Пусть автомат-спецификация S/s_1 является полностью определенным наблюдаемым автоматом, который обладает различающим тестовым примером и является *д*-связным. Тогда адаптивная стратегия,

направленная на построение множеств *Separate* и *Transition*, является исчерпывающей относительно модели неисправности $<S/s_1, \leq, \Omega>$, т.е. предъявленный автомат P/p_1 из области неисправности выдает только ожидаемые выходные реакции на построенную адаптивную проверяющую последовательность, если и только если P/p_1 есть редукция S/s_1.

5. Построение адаптивной проверяющей последовательности для частичных наблюдаемых автоматов

Предположим, что автомат-спецификация является частичным, возможно недетерминированным автоматом. В данной работе мы не обсуждаем построение адаптивной проверяющей последовательности в общем случае, тем не менее, для достаточно широкого класса автоматов такую последовательность можно построить на основе описанных выше результатов.

Пусть в автомате-спецификации S/s_1 существует полностью определенный наблюдаемый *д*-связный подавтомат S'/s_1, который обладает различающим тестовым примером. Согласно утверждению 2, полностью определенный автомат P'/p_1 будет редукцией автомата S'/s_1, если и только если P'/p_1 изоморфен подавтомату S'/s_1. Поскольку подавтомат S' обладает всеми необходимыми свойствами, то согласно теореме 5, для него можно построить исчерпывающую адаптивную стратегию.

Таким образом, множество *Separate* строится на основе подавтомата S'/s_1. После того как установлено взаимно однозначное соответствие между состояниями автомата-спецификации и проверяемого автомата, множество *Transition* строится так же как в разделе 3. Единственное отличие состоит в том, что в каждом состоянии проверяются только входные символы, по которым определены переходы в соответствующем состоянии.

Следует отметить, что если автомат-спецификация является частичным, то отношение редукции в модели неисправности заменяется на отношение квази-редукции [11]. В этом случае условие принадлежности выходной реакции проверяемого автомата множеству реакций спецификации должно выполняться только для входных последовательностей, на которых определено поведение спецификации. С другой стороны, такой подход не требует, чтобы спецификация была, вообще говоря, наблюдаемым автоматом; однако для разработки исчерпывающей адаптивной стратегии для частичных возможно ненаблюдаемых автоматов необходимы дальнейшие исследования.

6. Заключение

В данной статье предлагается адаптивная стратегия построения проверяющей последовательности для случая, когда автомат-спецификация и проверяемый автомат являются полностью определенными инициальными автоматами. При этом автомат-реализация является детерминированным автоматом, число

состояний которого не превышает число состояний автомата-спецификации, и отношением конформности является отношение редукции. Подобно детерминированным автоматам проверяющая последовательность строится при наличии определенных свойств у автомата-спецификации. Автомат-спецификация должен обладать различающим тестовым примером (должен существовать адаптивный различающий эксперимент) и быть детерминировано связным, т.е. каждое состояние должно быть детерминировано достижимо из любого другого состояния. Различающий тестовый пример может существовать и в том случае, когда в автомате-спецификации отсутствует разделяющая последовательность, и длина такого примера обычно значительно меньше длины разделяющей последовательности (если таковая существует). Более того, имеет смысл вместо *д*-передаточных последовательностей использовать адаптивные передаточные последовательности [18], что также расширит возможности использования адаптивной стратегии при построении проверяющей последовательности.

Acknowledgement. Данная работа выполнена при частичной поддержке РФФИ грантом No. 15-58-46013 СТ_а.

Acknowledgement. This work is partly supported by RFBR grant No. 15-58-46013 CT_a.

Список литературы

[1]. Kohavi Z. Switching and Finite Automata Theory, McGraw-Hill, New York, 1978.

[2]. Chow T.S.. “Testing software Design Modelled by Finite State Machines”, In IEEE Trans. Software Eng. Vol. 4 (3), 1978, pp. 178-187.

[3]. Dorofeeva R., El-Fakih K., Maag S., Cavalli A., Yevtushenko N. “FSM-based conformance testing methods: A survey annotated with experimental evaluation”, In Information & Software Technology, Vol. 52 (12), 2010, pp. 1286-1297.

[4]. Hennie F.C. “Fault-Detecting Experiments for Sequential Circuits”, In Proc. Fifth Ann. Symp. Switching Circuit Theory and Logical Design, 1964, pp. 95-110.

[5]. Petrenko A., Simão A., Yevtushenko N. “Generating Checking Sequences for Nondeterministic Finite State Machines”, In Proceedings of the ICST, 2012, pp. 310-319.

[6]. Жигулин М.В., Коломеец А.В., Кушик Н.Г., Шабалдин А.В. “Тестирование программной реализации протокола IRC на основе модели расширенного”, Известия Томского политехнического университета, Т. 318 (5), 2011, сс. 81-84.

[7]. Petrenko A., Simão A. “Generalizing the DS-Methods for Testing Non-Deterministic FSMs”, In Comput. J. 58(7), 2015, pp. 1656-1672.

[8]. Ермаков А.Д.“Синтез проверяющих последовательностей для недетерминированных автоматов относительно редукции”, Труды ИСП РАН, Т. 26(6), 2014, сс. 111-124.

[9]. Alur R., Courcoubetis C., Yannakakis M.. “Distinguishing tests for nondeterministic and probabilistic machines”, In Proceedings of the twenty-seventh annual ACM symposium on Theory of computing, 1995, pp. 363-372.

[10]. Kushik N., El-Fakih K., Yevtushenko N. ”Adaptive Homing and Distinguishing Experiments for Nondeterministic Finite State Machines”, In Lecture Notes in Computer Science, Vol. 8254, 2013, pp. 33-48.
[11]. Petrenko A., Yevtushenko N.. “Conformance Tests as Checking Experiments for Partial Nondeterministic FSM”, In Lecture Notes in Computer Science, Vol. 3997, 2005, pp. 118-133.
[12]. Petrenko A., Yevtushenko N.. “Adaptive Testing of Deterministic Implementations Specified by Nondeterministic FSMs”, In Lecture Notes in Computer Science, Vol. 7019, 2011, pp. 162-178.
[13]. Кушик Н.Г. Методы синтеза установочных и различающих экспериментов с недетерминированными автоматами. Диссертация на соискание ученой степени кандидата физико-математических наук, Томский Государственный университет, 2013.
[14]. Shabaldina N., El-Fakih K., Yevtushenko N. “Testing Nondeterministic Finite State Machines with Respect to the Separability Relation”, In Proceedings of Intern. Conf. on Testing Systems and Software (ICTSS/FATYES), 2007, pp. 305-318.
[15]. Ветрова М.В. Разработка алгоритмов синтеза и тестирования конечно-автоматных компенсаторов. Диссертация на соискание ученой степени технических наук, Томский Государственный университет, 2004.
[16]. Yevtushenko N., Kushik N. “Decreasing the length of Adaptive Distinguishing Experiments for Nondeterministic Merging-free Finite State Machines”, In Proceedings of IEEE East-West Design & Test Symposium, pp.338 – 341.
[17]. Güniçen C., Inan K., Türker U.C., Yenigün H. “The relation between preset distinguishing sequences and synchronizing sequences”, In Formal Aspects of Computing, Vol. 26 (6), 2014, pp. 1153–1167.
[18]. Kushik N., Yevtushenko N., Yenigun H. “Reducing the complexity of checking the existence and derivation of adaptive synchronizing experiments for nondeterministic FSMs”, In Proceedings of International Workshop on Domain Specific Model-based Approaches to Verification and Validation (AMARETTO’2016), 2016, pp. 83-90.

Deriving adaptive checking sequence for nondeterministic Finite State Machines

A.D. Ermakov <antonermak@inbox.ru>
N.V. Yevtushenko <yevtushenko@sibmail.com>
National Research Tomsk State University,
634050, Russia, Tomsk, Lenin Ave., 36.

Abstract. The derivation of checking sequences for Finite State Machines (FSMs) has a long history. There are many papers devoted to deriving a checking sequence that can distinguish a complete deterministic specification FSM from any non-equivalent FSM with the same number of states. To the best of our knowledge, for nondeterministic FSMs, the topic appeared only recently; the authors started with preset checking sequences for FSMs where the initial state is still known but the reset is very expensive. In this paper, a technique is proposed for deriving an adaptive checking sequence for a complete nondeterministic finite state machine with respect to the reduction relation. The main contribution of the paper is the use of a (adaptive) distinguishing test case instead of a separating sequence. Such a test case

is usually shorter that a separating sequence (when it exists) and can exist when there is no separating sequence. We also discuss the possibilities of using adaptive transfer sequences instead of deterministic transfer sequences that also allows to extend the set of FSMs for which the strategy can be used and reduce the length of a checking sequence. The application of a proposed strategy to partial possibly nondeterministic FSMs is briefly discussed.

Keywords: nondeterministic Finite State Machines (FSM); reduction relation; fault model; test derivation; adaptive checking sequences.

DOI: 10.15514/ISPRAS-2016-28(3)-8

For citation: A.D. Ermakov, N.V. Yevtushenko. Deriving adaptive checking sequence for nondeterministic Finite State Machines. *Trudy ISP RAN /Proc. ISP RAS*, 2016, vol. 28, issue 3, pp. 123-144 (in Russian). DOI: 10.15514/ISPRAS-2016-28(3)-8

References

[1]. Kohavi Z. Switching and Finite Automata Theory, McGraw-Hill, New York, 1978.
[2]. Chow T.S.. "Testing software Design Modelled by Finite State Machines", In IEEE Trans. Software Eng. Vol. 4 (3), 1978, pp. 178-187.
[3]. Dorofeeva R., El-Fakih K., Maag S., Cavalli A., Yevtushenko N. "FSM-based conformance testing methods: A survey annotated with experimental evaluation", In Information & Software Technology, Vol. 52 (12), 2010, pp. 1286-1297.
[4]. Hennie F.C. "Fault-Detecting Experiments for Sequential Circuits", In Proc. Fifth Ann. Symp. Switching Circuit Theory and Logical Design, 1964, pp. 95-110.
[5]. Petrenko A., Simão A., Yevtushenko N. "Generating Checking Sequences for Nondeterministic Finite State Machines", In Proceedings of the ICST, 2012, pp. 310-319.
[6]. Zhigulin M., Kolomeez A., Kushik N., Shabaldin A... "EFSM based testing a software implementation of IRC protocol", In Izvestia Tomskogo polytechnicheskogo instituta [Bulletin of the Tomsk Polytechnic University], 318 (5), 2011, pp. 81-84 (in Russian).
[7]. Petrenko A., Simão A.. "Generalizing the DS-Methods for Testing Non-Deterministic FSMs", In Comput. J. 58(7), 2015, pp. 1656-1672.
[8]. Ermakov A.. "Deriving checking sequences for nondeterministic FSMs", In Proceedings of the Institute for System Programming of RAS, Vol. 26, 2014, pp. 111-124 (in Russian).
[9]. Alur R., Courcoubetis C., Yannakakis M.. "Distinguishing tests for nondeterministic and probabilistic machines", In Proceedings of the twenty-seventh annual ACM symposium on Theory of computing, 1995, pp. 363-372.
[10]. Kushik N., El-Fakih K., Yevtushenko N. "Adaptive Homing and Distinguishing Experiments for Nondeterministic Finite State Machines", In Lecture Notes in Computer Science, Vol. 8254, 2013, pp. 33-48.
[11]. Petrenko A., Yevtushenko N.. "Conformance Tests as Checking Experiments for Partial Nondeterministic FSM", In Lecture Notes in Computer Science, Vol. 3997, 2005, pp. 118-133.
[12]. Petrenko A., Yevtushenko N.. "Adaptive Testing of Deterministic Implementations Specified by Nondeterministic FSMs", In Lecture Notes in Computer Science, Vol. 7019, 2011, pp. 162-178.

[13]. Kushik N.. Methods for deriving homing and distinguishing experiments for nondeterministic FSMs. PhD thesis, Tomsk State University, 2013 (in Russian).
[14]. Shabaldina N., El-Fakih K., Yevtushenko N. “Testing Nondeterministic Finite State Machines with Respect to the Separability Relation”, In Proceedings of Intern. Conf. on Testing Systems and Software (ICTSS/FATYES), 2007, pp. 305-318.
[15]. Vetrova M. FSM based methods for compensator design and testing. PhD thesis, Tomsk State University, 2004 (in Russian).
[16]. Yevtushenko N., Kushik N. Decreasing the length of Adaptive Distinguishing Experiments for Nondeterministic Merging-free Finite State Machines / // Proceedings of IEEE East-West Design & Test Symposium, pp.338 – 341.
[17]. Güniçen C., Inan K., Türker U.C., Yenigün H. “The relation between preset distinguishing sequences and synchronizing sequences”, In Formal Aspects of Computing, Vol. 26 (6), 2014, pp. 1153–1167.
[18]. Kushik N., Yevtushenko N., Yenigun H. “Reducing the complexity of checking the existence and derivation of adaptive synchronizing experiments for nondeterministic FSMs”, In Proceedings of International Workshop on Domain Specific Model-based Approaches to Verification and Validation (AMARETTO’2016), 2016, pp. 83-90.

Дробинцев П.Д., Котляров В.П., Никифоров И.В., Воинов Н.В., И.А. Селин. Преобразование абстрактных поведенческих сценариев в сценарии применимые для тестирования. *Труды ИСП РАН*, том 28, вып. 3, 2016, с. 145-160.

Conversion of abstract behavioral scenarios into scenarios applicable for testing

Pavel Drobintsev < drob@ics2.ecd.spbstu.ru>
Vsevolod Kotlyarov <vpk@ics2.ecd.spbstu.ru>
Igor Nikiforov < i.nikiforov@ics2.ecd.spbstu.ru>
Nikita Voinov <voinov@ics2.ecd.spbstu.ru>
Ivan Selin < ivanselin93@gmail.com>
Peter the Great Saint-Petersburg Polytechnic University,
29 Polytechnicheskaya str, St. Petersburg, 195251, Russian Federation

Abstract. In this article, an approach of detailing verified test scenarios for developed software system without losing the model's semantics is proposed. Existing problem of generating test cases for real software systems is solved by using multi-level paradigm to obtain the real system signals, transactions and states. Because of this, the process is divided into several steps. Initial abstract traces (test cases) with symbolic values are generated from the verified behavioral model of software product. On the next step, called concretization, these values in test scenarios are replaced with concrete ones. Resulting concrete traces are then used as input for the next step, data structures conversion. This step is needed because concrete traces do not contain all the information for communicating with developed software and presented in another way with different data structures. After concrete test scenarios are detailed, they can be used for generation of executable test cases for informational and control systems. In this paper, a software tool is suggested for detailing test scenarios. It consists of several modules: a Lowering editor that allows user to create rules of detailing a signal, a Signals editor used to define complex data structures inside the signal and a Templates editor that eases work with similar signals. Process of translating abstract data structures into detailed data structures used in system implementation is presented with examples.

Keywords: model approach; model verification; test mapping

DOI: 10.15514/ISPRAS-2016-28(3)-9

For citation: P. Drobintsev, V. Kotlyarov, I. Nikiforov, N. Voinov, I. Selin. Conversion of abstract behavioral scenarios into scenarios applicable for testing. Trudy ISP RAN / Proc. ISP RAS, vol. 28, issue 3, 2016, pp. 145-160. DOI: 10.15514/ISPRAS-2016-28(3)-9.

1. Introduction

One of the most perspective approaches to modern software product creation is usage of model oriented technologies both for software development and testing. Such technologies are called MDA (Model Driven Architecture) [1,2], MDD (Model Driven Development) [2] and MDSD (Model Driven Software Development) [3]. All of them are mainly aimed to design and generation of application target code based on a formal model.

The article is devoted to specifics of model oriented approaches usage in design and generation of large industrial software applications. These applications are characterized by multilevel representation related to detailing application functionality to the level where correct code is directly generated.

The idea of model oriented approach is in creating of multilevel model of application during design process. This model is iteratively specified and detailed to the level when executable code can be generated. On the design stage formal model specification allows using verification together with other methods of static analysis with goal to guaranty correctness of the model on early stages of application development.

More than 80% [4] of model-oriented approaches are using graphical notations, which allows simplifying of work with formal notations for developers. Requirements for knowledge of testers and customer representatives is reduced by this way and process of models developing are also simplified.

2. Levels of behavioral models development

One of high level languages for system formal model specification is Use Case Maps (UCM) [5, 6]. It provides visible and easy understandable graphical notation. Further abstract models will be specified in UCM language to demonstrate proposed approach in details. Also considered is VRS/TAT technology chain [7], which uses formal UCM models for behavioral scenarios generation.

Traditional steps of formal abstract model development in UCM language are the following:

1. Specifying main interacting agents (components) and their properties, attributes set by agent and global variables.
2. Introducing main system behaviors to the model and developing diagrams of agent's interaction control flow.
3. Developing internal behaviors for each agent and specifying data flow in the system.

Undoubted benefit of UCM language is possibility to create detailed structured behavioral diagrams. Structuring is specified both by Stub structural elements and reused diagrams (Maps), which are modeling function calls or macro substitution. Unfortunately, standard UCM language deals with primitive and abstract data structures, which are not enough to check implementation of a real system. This

drawback is compensated by using metadata mechanism [6]. But metadata does not allow detailing data flow to more detailed levels. That's why for creating detailed behaviors it is proposed to use vertical levels of abstractions during behavioral models development which are: structured system model in UCM language, behavioral scenarios with symbolic values and variables, concrete behavioral scenarios are behavioral scenarios with detailed data structures.

Another benefit of UCM usage is possibility to execute model verification process. UCM diagrams are used as input for VRS/TAT toolset which provides checks for specifications correctness. These checks can detect issues with unreachable states in the model, uninitialized variables in metadata, counterexamples for definite path in UCM, etc. After all checks are completed the user gets a verdict with a list of all findings and a set of counterexamples which show those paths in UCM model which lead to issue situations. If a finding is considered to be an error, the model is corrected and verification process is launched again. As a result after all fixes a correct formal model is obtained which can be used for further generation of test scenarios.

After formal model of a system has been specified in UCM language, behavioral scenarios generation is performed. Note that behavioral generator is based not on concrete values assigned to global variables and agents attributes, but on symbolic ones which reduces significantly the number of behavioral scenarios covering the model. However symbolic test scenarios cannot be used for applications testing as executing behavioral scenarios on the real system requires concrete values for variables. So the problem of different level of abstraction between model and real system still exists. In VRS/TAT technology concretization step [8] is used to convert symbolic test scenarios. On this step ranges of possible values for variables and attributes are calculated based on symbolic formula and symbolic values are substituted with concrete ones. But concretization of abstract model's behavioral scenarios is not enough for their execution, because on this stage scenarios still use abstract data structures which differ from data structures in real system. As a result conversion of concretized behavioral scenarios of abstract UCM level into scenarios of real system level was integrated into technology chain for behavioral scenarios generation.

2. Concretization

In behavioral scenarios data structures are mainly used in signals parameters. There are two types of signals in UCM model: incoming to an agent and outgoing from an agent. Incoming signals are specified with the keyword "in" and can be sent either by an agent or from outside the system specifying with the keyword "found". Outgoing signals are specified with the keyword "out" and can be sent either to an agent or to outside the system specifying with the keyword "lost".

An example of outgoing signal can be seen on Fig. 1. The element "send_Fwd_Rel_Req_V2_papu" contains metadata with the signal

"Forward_Relocation_Request_V2" and UCM-level parameter "no_dns". Outgoing signals can only be used inside of "do" section as a reaction of the system on some event.

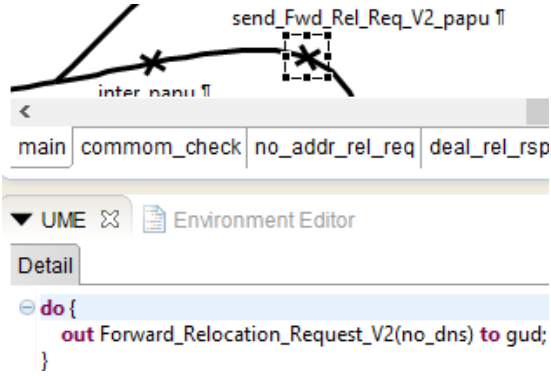


Fig. 1. Description of "Forward_Relocation_Request_V2" signal in metadata corresponding UCM element

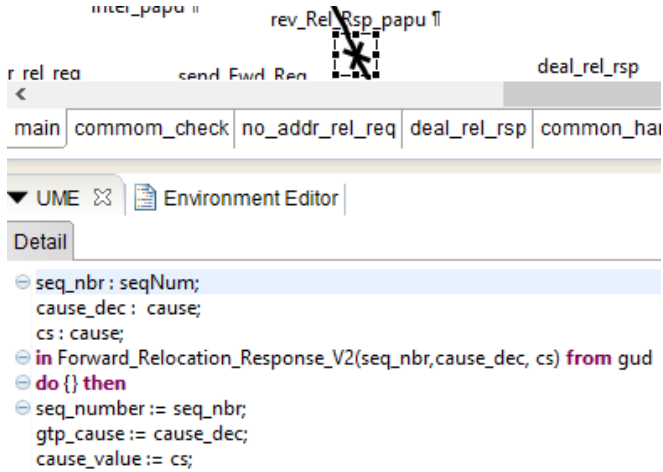


Fig. 2. Description of the " Forward_Relocation_Response_V2" signal in metadata of the "rev_Rel_RSP_papu" UCM element

If the signal Forward_Relocation_Response_V2 is received, then new values taken from signal parameters are assigned to variables.

Consider an example of converting signal structure of UCM level into detailed structures of real system for the signal "gtp_forward_relocation_req_s". Based on high level UCM model symbolic behavioral scenarios are generated containing data structures described in metadata of UCM elements. Fig.3 contains symbolic test

scenario where the agent "GTP#gtp" receives the signal "gtp_forward_relocation_req_s" from agent "GMG#gmg". In symbolic scenarios actual names of UCM model agents specified in metadata are used.

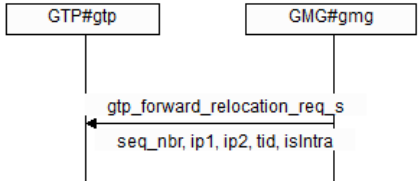


Fig. 3. Symbolic test scenario with the signal "gtp_forward_relocation_req_s"

Symbolic behavioral scenario is input data for concretization module, which substitutes symbolic parameters with concrete values. In current example the parameters "sqe_nbr", "ip1", "ip2", "tid" and "isIntra" are substituted with values "invalid", "valid", "exist", "valid" and "0". Fig.4 contains concrete behavioral scenario.

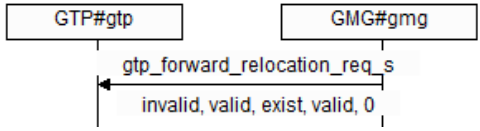


Fig. 4. Concrete test scenario with the signal "gtp_forward_relocation_req_s"

4. Data structures conversion

After concretization, scenarios still have to be processed because their structure does not match with one's of system under test (SUT). The most straightforward approach is to manually review all generated scenarios and edit all used signals so that their structure will reflect SUT interfaces. Obviously, it will require too much time and may be a bottleneck of the whole process. Therefore, there is a need for automation.

The common way is making a wrapper that transforms signals to desired form using one of popular programming languages (C++, Java, etc.). However, it could lead to making new mistakes and loss of correctness of test scenarios. The main reason for this is ability to implement incorrect structures on scenarios level. In addition, other language-specific errors are possible. Cutting down the ability to produce incorrect code will reduce the number of mistakes while still maintaining good level of automation.

4.1 Approach

To be able to satisfy these needs a two-step approach called "Lowering" was suggested. The name comes from descending on lower levels of abstraction. In

general, lowering can be described as creating processing rules for each signal called "lowering rules" and application of these rules to the concrete scenarios.

As said above, there are some restrictions on possible operations to save the correctness of test scenarios, such as:

- It is prohibited to separate constants into several independent parts (e.g. separating value 1536 in 15 and 36 is not possible)
- It is prohibited to separate fields of variables values
- Only structures similar to SUT interfaces can be created
- Only constant template values and values that were obtained during concretization step are allowed

Limitation was made by creating a special language that is used to define lowering rules. Despite having all these limitations, user can define complex signal and protocol structure dependent on UCM signal parameters in accordance with language grammar. On Fig. 5, you can see the grammar in Backus–Naur Form.

```
LoweringSpec ::= UCMSignal "->"
LoweringRule | LoweringSpec UCMSignal "->"
LoweringRule
LoweringRule ::=  LoweringCondition |
LoweringRule LoweringCondition
LoweringCondition ::= <condition STRING>
ConditionContent
ConditionContent ::= LoweredElement |
LoweredElement ConditionContent
LowredElement ::= LoweredDo | LoweredSignal
| LoweredAction
LoweredDo ::= <code STRING>
LoweringSignal ::= <signal name STRING>
SignalContent
SignalContent ::= ValueNotation Instance
Via
ValueNotation ::= <empty> | <value STRING>
| "(." ValueNotation ".)" | ValueNotation
"," ValueNotation
Instance ::= <empty> | "TAT" | "SUT"
Via := <empty> | <port STRING>
UCMSignal ::= Name UCMParam
Name ::= <name STRING>
UCMParam ::= <empty> | <param name STRING>
| UCMParam "," UCMParam
```

Fig. 5. Lowering rules language grammar

4.2 User perspective

For selected UCM-level signal user can define lowering rules. As you can see on Fig. 6, rule consists of trigger condition and content. Content can be either one detailed signal, several signals or actions performed on the variables.

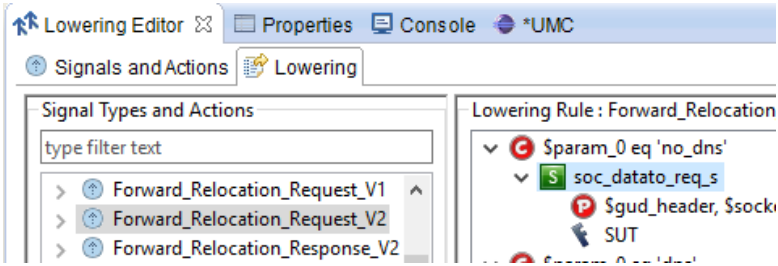


Fig. 6. Lowering editor with signal "Forward_Relocation_Request_V2" being selected

After specifying the condition and choosing the type of content, user can edit it in the right part of the editor. This part dynamically changes depending on what is selected in the middle of the editor.

For example, some signal was selected. Signals editor will appear in the right part of Lowering Editor (Fig. 7).

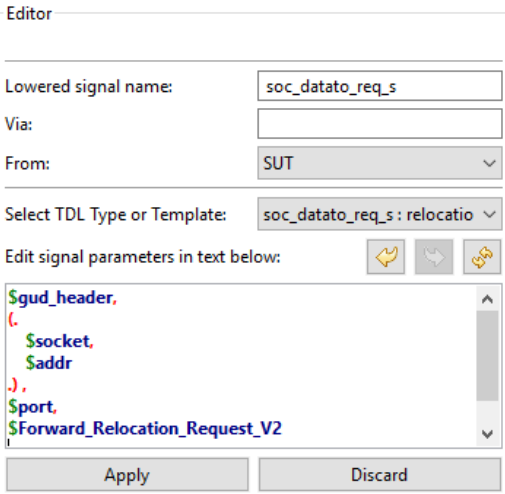


Fig. 7. Signals editor

User selects the needed SUT interface in the drop-down list named "Select TDL Type or Template". Then user names the signal and puts concrete values in the fields of detailed signal. Often similar conversion rules are required for different signals. Templates can be used to simplify this approach. A developer can define a template of detailed signal, specify either formula or concrete values as a parameter

of detailed signal and then apply this template for all required signals. For each case of template usage a developer can specify missed values in the template, change the template itself or modify its structure without violating specified limitations. Templates mechanism simplifies significantly the process of conversion rules creation.

Consider the process of templates usage. Templates are created in separate editor (Templates Editor). In Fig.8 the template "template_0" is shown which contains detailed data structures inside and the dummy values which shall be changed to concrete values when template is used.

Note that template can be created only from SUT interfaces description or another template.

When a template of data structure is ready, it can be used for creation of conversion rules. Fig.9 represents usage of the template "template_0" with substituted concrete values of signal parameters instead of the dummy value "value_temp", which then will appear in behavioral MSC scenario.

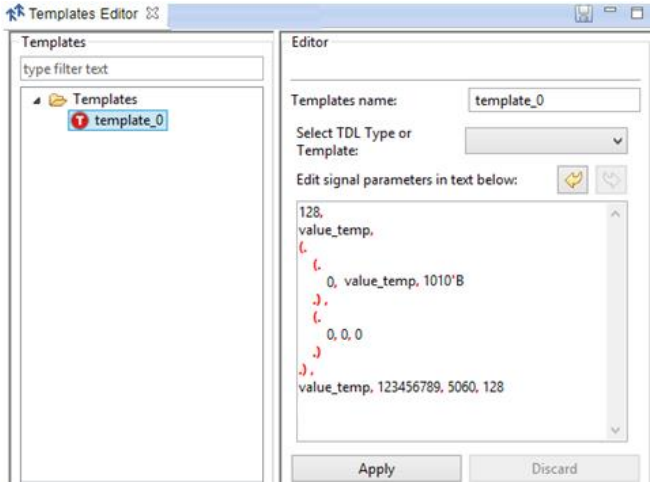


Fig. 8. Templates editor

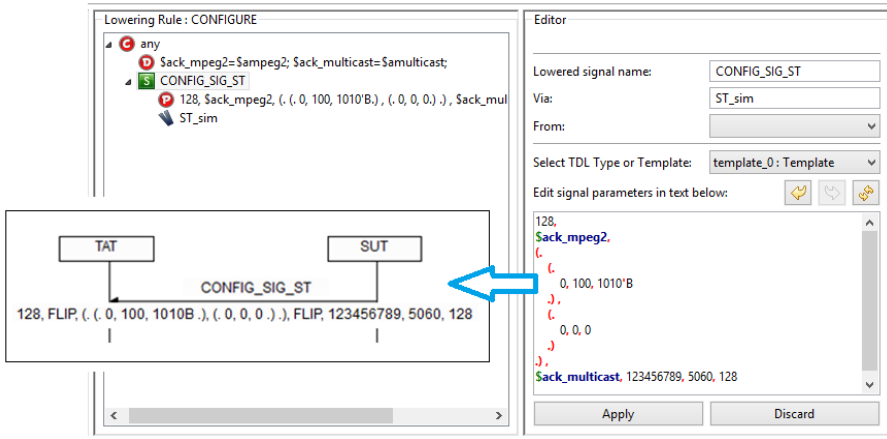


Fig. 9. Template used in signals editor

In both signal and template editors user can use variables – some values that are too big to remember of retype every time. On the Fig. 7 all the values are taken from variables. Variables can be selected in the middle of the lowering editor. There are different types of variables with different editors and checks. For example, the contents of variable "$gud_header" used in "soc_datato_req_s" detailed signal are shown on Fig. 10.

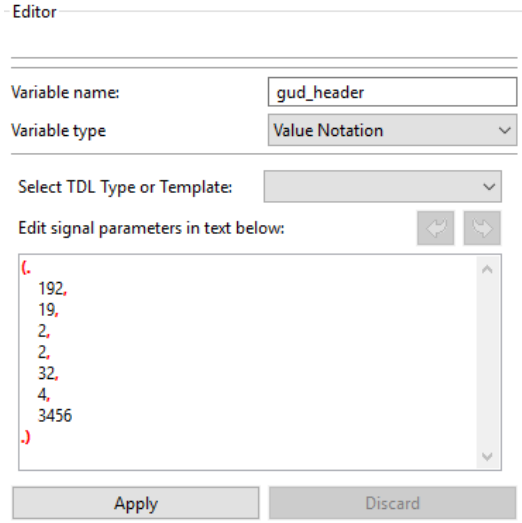


Fig. 10. Contents of the variable "$gud_header"

Variables can contain very complex structures and therefore greatly reduce expenses on creating detailed signals.

Overall process of selecting UCM-level signal, creating lowering rules and editing the resulting signal repeats for all UCM-level signals in the project.

4.3 Scenarios processing

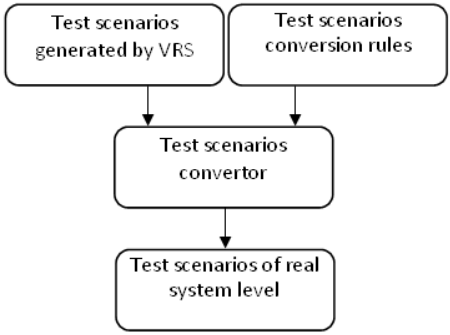


Fig. 11. Test scenarios conversion scheme

Implemented module of behavioral scenarios conversion takes as an input the concrete behavioral scenarios and specified rules of conversion and the output is behavioral scenarios of the real system level, which can be used for testing. Overall scheme of conversion is shown in Fig.11.

Detailing stage is based on the grammar of data structures conversion rules described in Fig. 5 and conversion algorithm. The specific feature of test automatic scenarios detailing to the level of real system is allow to storing of proved properties of the system obtained in process of abstract model verification.

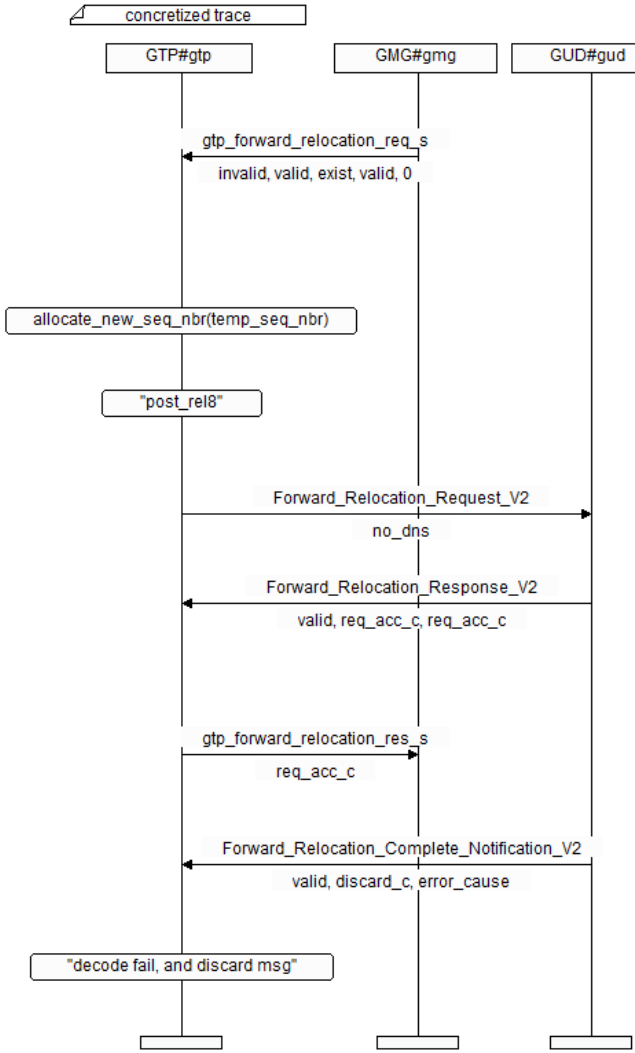


Fig. 12. Concrete scenario to be lowered

Based on the specified conversion rule each abstract signal in concrete behavioral scenario is processed. Signal parameters are matched to rule conditions and if the signal satisfies them, then it is converted into detailed form. Fig.12 shows concrete scenario, which will be processed.

In this scenario you can see 3 agents: "GTP#gtp", "GMG#gmg" and "GUD#gud". For example, we want to test an agent "GTP#gtp". On following trace it will be described as SUT.

Other agents (or whichever we choose in the settings of the trace preprocessing) are marked as TAT and joined together.

After data structures conversion, concrete signals are replaced with detailed signals specified in lowering rules. Once simple signal structure unfolds in very complex nested data while still maintaining its correctness. You can see the results on Fig. 13.

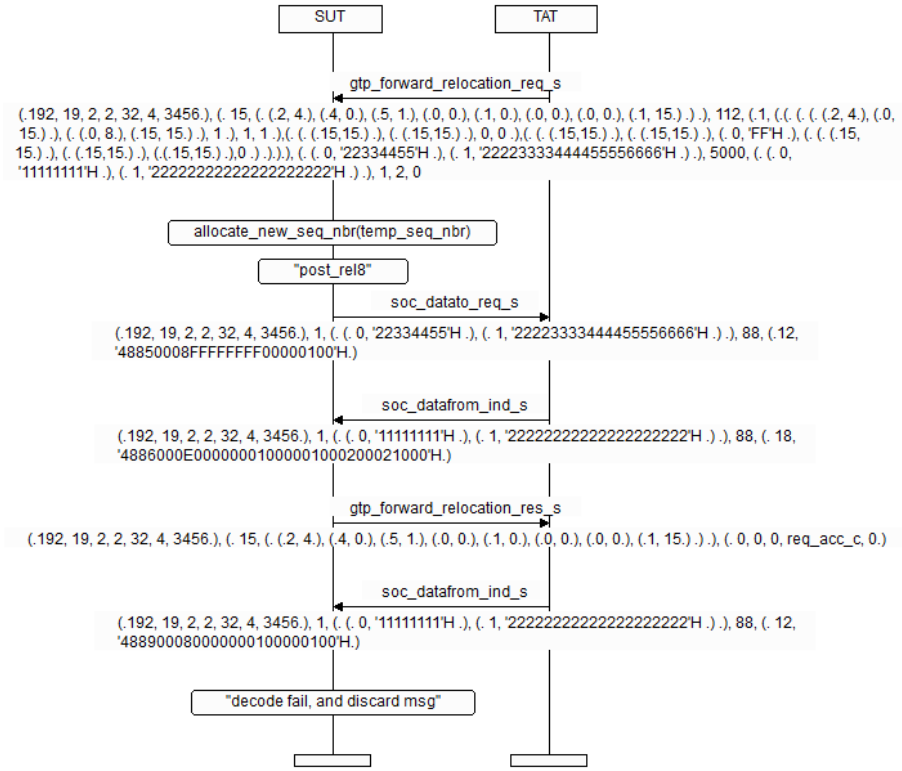


Fig. 13. Lowered trace with detailed signals

5. Conclusion

Proposed approach to behavioral scenarios generation based on formal models differs from existing approaches in using the process of automatic conversion of behavioral scenarios with abstract data structures into behavioral scenarios with detailed data structures used in real applications. Proposed language and overall

scheme of this process allow automating of creation a set of covering behavioral scenarios.

In the scope of this work, the analyzer/editor for conversion rules of signals from abstract UCM model level into signals of real system level was developed and called Lowering Editor. It supports the following functionality: automatic binding between conversion rule and signal of UCM level, conversion rules correctness checking, templates usage, highlighting the syntax of conversion rules applying conditions specification, variables usage, libraries and external scripts (includes) usage, splitting UCM signal or action into several signals of real system in according to communication protocol, copy/paste/remove operations, import and export from/to storage file. Availability of described in the article features is able to make process of automatic conversion powerful and flexible for a different types of telecommunication applications.

Adding Lowering Editor into technology process of telecommunication software applications test automation allowed to exclude effort-consuming manual work in the cycle of test suite automated generation for industrial telecommunication applications, increase productivity of test generation in 25% and spread the properties proved on abstract models into generated code of executable test sets. Excluding of manual work allow to reduce human factor in testing process and guaranty quality of generated test suite based on verification results.

References

[1]. Model Driven Architecture- MDA (2007). Available at: http://www.omg.org/mda

[2]. Oscar Pastor, Sergio España, José Ignacio Panach, Nathalie Aquino. Model-Driven Development. Informatik Spektrum, vol. 31, no. 5, pp. 394-407 (2008)

[3]. Sami Beydeda , Matthias Book, Volker Gruhn. Model Driven Software Development.: Springer-Verlag Berlin Heidelberg, 464 p. (2005)

[4]. Robert V. Binder, Anne Kramer, Bruno Legeard. 2014 Model-based Testing User Survey: Results, 2014. Available at: http://model-based-testing.info/wordpress/wp-content/uploads/2014_MBT_User_Survey_Results.pdf

[5]. Buhr R. J. A., Casselman R. S. Use Case Maps for Object-Oriented Systems. Prentice Hall. 302 p. (1995)

[6]. A.A. Letichevsky, J.V. Kapitonova, V.P. Kotlyarov, A.A. Letichevsky Jr., N.S.Nikitchenko, V.A. Volkov, and T. Weigert. Insertion modeling in distributed system design. Problemy programuvannja [Problems of programming] (4), pp. 13–39 (2008).

[7]. I.Anureev, S.Baranov, D.Beloglazov, E.Bodin, P.Drobintsev, A.Kolchin, V. Kotlyarov, A. Letichevsky, A. Letichevsky Jr., V.Nepomniaschy, I.Nikiforov, S. Potienko, L.Pryima, B.Tyutin. Tools for supporting integrated technology of analysis and verification of specifications for telecommunication applications. Trudy SPIIRAN [SPIIRAS Proceedings], 2013, issue 26. pp. 349-383 (in Russian).

[8]. A. Kolchin, A. Letichevsky, V. Peschanenko, P. Drobintsev, V. Kotlyarov. Approach to creating concretized test scenarios within test automation technology for industrial software projects. Automatic Control and Computer Sciences, vol. 47, no. 7, pp. 433–442 (2013).

Преобразование абстрактных поведенческих сценариев в сценарии применимые для тестирования

П.Д. Дробинцев <drob@ics2.ecd.spbstu.ru>
В.П. Котляров <vpk@ics2.ecd.spbstu.ru >
И.В. Никифоров < i.nikiforov@ics2.ecd.spbstu.ru >
Н.В. Воинов <voinov@ics2.ecd.spbstu.ru >
И.А. Селин < ivanselin93@gmail.com>
Санкт-Петербургский политехнический университет Петра Великого, 195251, Россия, г. Санкт-Петербург, ул. Политехническая, дом 29

Аннотация. В данной статье рассмотрен подход детализации верифицированных тестовых сценариев для разрабатываемой программной системы без изменения семантики набора, то есть с сохранением корректности. Существующая проблема генерации тестов реальных приложений на основе верифицированных абстрактных сценариев, сгенерированных по поведенческой модели, решается на основе детализации абстрактных сценариев до уровня конкретных состояний, транзакций, протоколов и сигналов. Поскольку характерной особенностью рассматриваемых абстрактных моделей является символьное представление поведенческих сценариев, то их детализация происходит в два этапа. На первом этапе – этапе конкретизации, символьные параметры сигналов получают конкретные значения, образуя тем самым конкретные поведенческие сценарии. На втором этапе – этапе собственно детализации, конкретные абстрактные сценарии необходимо представлять в виде структур данных, формы представления и значения которых содержат всю необходимую информацию для обмена с реальными приложениями. Полученные таким образом детальные сценарии предназначены для генерации исполнимых тестовых наборов для информационных и управляющих систем. В работе предложен инструментарий детализации тестовых сценариев, позволяющий не только описать реальные сигналы, но и детализировать протоколы обмена сигналами. В его состав входит Lowering editor, позволяющий описывать правила преобразования сигналов в соответствии с приведенной разработанной грамматикой правил преобразований, Signals editor, используемый для удобного описания сложных структур сигналов и Templates editor, позволяющий однократно описывать типовые структуры. Приведён пример процесса преобразования от абстрактных структур данных к детализированным, использующимся при тестировании целевого кода.

Ключевые слова: model approach; model verification; test mapping

DOI: 10.15514/ISPRAS-2016-28(3)-9

Для цитирования: П.Д. Дробинцев, В.П. Котляров, И.В. Никифоров, Н.В. Воинов, И.А. Селин. Преобразование абстрактных поведенческих сценариев в сценарии применимые для тестирования. Труды ИСП РАН, том 28, вып. 3, 2016 г. стр. 145-160 (на английском). DOI 10.15514/ISPRAS-2016-28(3)-9

Список литературы

[9]. Model Driven Architecture – MDA (2007), доступно по ссылке: http://www.omg.org/mda

[10]. Oscar Pastor, Sergio España, José Ignacio Panach, Nathalie Aquino. Model-Driven Development. Informatik Spektrum, Volume 31, Number 5, pp. 394-407 (2008)

[11]. Sami Beydeda , Matthias Book, Volker Gruhn. Model Driven Software Development.: Springer-Verlag Berlin Heidelberg, 464 p. (2005)

[12]. Robert V. Binder, Anne Kramer, Bruno Legeard. 2014 Model-based Testing User Survey: Results, 2014, доступно по ссылке: http://model-based-testing.info/wordpress/wp-content/uploads/2014_MBT_User_Survey_Results.pdf

[13]. Buhr R. J. A., Casselman R. S.: Use Case Maps for Object-Oriented Systems. Prentice Hall. 302 p. (1995)

[14]. A.A. Letichevsky, J.V. Kapitonova , V.P. Kotlyarov, A.A. Letichevsky Jr., N.S.Nikitchenko, V.A. Volkov, and T.Weigert. Insertion modeling in distributed system design. Проблеми програмування, pp. 13–39 (2008).

[15]. Ануреев И.С., Баранов С.Н., Белоглазов Д.М., Дробинцев П.Д., Колчин А.В., Котляров В.П., Летичевский А.А., Летичевский А.А. мл., Непомнящий В.А., Никифоров И.В., Потиенко С.В., Прийма Л.В., Тютин Б.В., Бодин Е.М. Средства поддержки интегрированной технологии для анализа и верификации спецификаций телекоммуникационных приложений. Труды СПИИРАН. 2013, вып. 26, стр. 349-383.

[16]. A. Kolchin, A. Letichevsky, V. Peschanenko, P. Drobintsev, V. Kotlyarov. Approach to creating concretized test scenarios within test automation technology for industrial software projects. Automatic Control and Computer Sciences, vol. 47, no. 7, pp. 433–442 (2013)

Approaches to Stand-alone Verification of Multicore Microprocessor Caches

Mikhail Petrochenkov <petroch_m@mcst.ru>
Irina Stotland <stotl_i@mcst.ru>
Ruslan Mushtakov <mushtakov_r@mcst.ru>
MCST, 1 Nagatinskaya st., Moscow, 117105, Russia

Abstract. The paper presents an overview of approaches used in verifying correctness of multicore microprocessors caches. Common properties of memory subsystem devices and those specific to caches are described. We describe the method to support memory consistency in a system using cache coherence protocol. The approaches for designing a test system, generating valid stimuli and checking the correctness of the device under verification (DUV) are introduced. Adjustments to the approach for supporting generation of out-of-order test stimuli are provided. Methods of the test system development on different abstraction levels are presented. We provide basic approach to device behavior checking — implementing a functional reference model, reactions of this model could be compared to device reactions, miscompare denotes an error. Methods for verification of functionally nondeterministic devices are described: the «gray box» method based on elimination of nondeterministic behavior using internal interfaces of the implementation and the novel approach based on the dynamic refinement of the behavioral model using device reactions. We also provide a way to augment a stimulus generator with assertions to further increase error detection capabilities of the test system. Additionally, we describe how the test systems for devices, that support out of order execution, could be designed. We present the approach to simplify checking of nondeterministic devices with out-of-order execution of requests using a reference order of instructions. In conclusion, we provide the case study of using these approaches to verify caches of microprocessors with “Elbrus” architecture and “SPARC-V9” architecture.

Keywords: multicore microprocessors; cache memory; out-of-order execution; test system; nondeterministic behavior; model-based verification; stand-alone verification; “SPARC-V9”; “Elbrus-8C”.

DOI:10.15514/ISPRAS-2016-28(3)-10

For citation: Petrochenkov M., Stotland I., Mushtakov R. Approaches to Stand-alone Verification of Multicore Multiprocessor Cores. *Trudy ISP RAN/Proc.ISP RAS*, vol. 28, issue 3, 2016, pp. 161-172. DOI: 10.15514/ISPRAS-2016-28(3)-10

1. Introduction

The key feature of modern microprocessor architecture is multicoreness — combining several computational cores on a single system on a chip (SOC). To reduce time needed to access RAM (Random Access Memory), device can incorporate several levels of cache hierarchy. Access to smaller caches could be executed faster than access to larger caches of the next level of the hierarchy. Caches can keep data for a single computational core or serve as data storage for several of them at the same time. A memory subsystem of a multicore microprocessor must maintain coherence of the memory. Task of maintaining correct state of memory is usually solved by implementing cache coherence protocol that defines a set of data states and actions on transitions between states in a cache [1]. To optimize the design and the implementation of coherency protocol, caches can include a local directory — the device which keeps information on states of data in different components of the memory subsystem. Sufficient complexity of protocols and their implementations in multilevel memory subsystems can lead to hard to find errors.

To ensure the robustness of a microprocessor, one must thoroughly verify its memory subsystem. The importance of the *functional verification* — the checking of correspondence between specifications of designs and their implementations — is obvious for many reasons. This activity could be found out to take more than 70% out of the total design development time. Two main approaches to functional verification of microprocessors are formal verification and simulation-based methods [2]. Formal methods are exhaustive and based on analyzing static formal model. Models are large and formal verification techniques face the “combinatorial explosion” issue. Simulation-based methods are not exhaustive, but they are much more flexible and thereby employed at different stages. We can verify not only the static model of system, but also implementation. The object of simulation-based verification is RTL (Register Transfer Level) model of device.

One of the approaches to microprocessor verification is execution of test programs on the microprocessor model and on the reference implementation of its instruction set, and comparison between them. Such approach is called *system verification*. It should be noticed that caches are often invisible from the point of view of a programmer. That is why designing programs capable for sufficient verification of a microprocessor caches is a complex task.

One way to shorten the design of microprocessors is the application of unit-based verification. It is assumed that system is divided into a set of components and the general functionality of the components does not change [3]. Such a way of verification is called *stand-alone verification*. This paper addresses the problem of stand-alone verification of microprocessor caches of different levels.

The rest of the paper is organized as follows. Section 2 suggests an approach to the problem. Section 3 presents the test stimui generation methods. Section 4 reviews the existing techniques for designing test oracles. Section 5 describes a case study

on using the suggested approach in an industrial setting. Section 6 concludes the paper.

2. Common View on Stand-alone Verification of Microprocessor Caches

The object of stand-alone verification is model of the device under verification (DUV) implemented in hardware description language (usually, Verilog or VHDL). It defines the behavior of the device on a register transfer level. The device specification defines a set of stimuli and reactions based on the state of the device. To check the correctness of the device it is included in a test system — a program that generates test stimuli, checks validity of reactions and determines verification quality. Based on its functions test system can be divided into separate modules — stimulus generator, correctness checking module (test oracle) and coverage collector. Methods of estimation of verification quality are similar to that of other devices: information on functional code coverage is used to identify unimplemented test scenarios and refine stimulus generation by adding new test scenarios and improving existing stimulus generator. This approach is called coverage driven constrained random verification. Besides this, there are some approaches to microprocessor caches verification. In paper [4] authors propose using decomposition and abstraction for standalone verification. In our previous projects, we have used the decomposition methods for L2-cache verification: L2-cache was divided into several submodules for which reference cycle-accurate bit-to-bit models and test systems were implemented [3]. This approach allowed to find bugs in submodules, but did not give the chance to check the cache in general. We also can use a SystemC reference model as presented in [5] but it is employing if SystemC models are used in other stages of an ASIC design flow. In paper [6] the approach to test oracle development for nondeterministic models is presented. However, this approach refers only test oracle developing of in-order cache execution and has no recommendation for caches with out-of-order execution. In such a way, the main goal of this work lies in developing some new techniques of standalone verification of microprocessor caches with different ordering of stimulus execution.

Cache behavior exhibits a set of properties that should be considered while designing a test system for verification of the device:

- Transactions (or requests) in the microprocessor system can be separated into three groups: *primary requests* — requests from subscribers (other caches, cores, etc.) to perform an operation with the memory (load/store), *secondary requests* — responses of the test system to some reaction of the cache, and *reactions* — output transactions from the cache
- A device implements a part of cache coherence protocol
- A device works independently with different cache lines — areas of memory of fixed size

- Requests that work with the same cache line are serialized. It means that requests complete in the same order as they are received
- Device implements data eviction mechanism and protocol to determine victim line (usually some variant of least recent used algorithm (LRU)).

Using these properties of the device under testing while designing a test system could lead to the simplified structure using separate stimulus generators — the primary requests generator and the secondary requests generator. We also can use the fact that requests are serialized for checking the correctness of caches with out-of-order execution.

3. Test Stimuli Generation

3.1 The common approach

Test stimuli are usually generated at more abstract level than register transfers and interface signals. Based on the logical and functional similarity, groups of device ports are combined into interfaces. Interfaces are used to transfer transaction level packets [7]. To transform packets between different representations on signal and transaction level, serializer and deserializer modules are implemented[8].

Test system should generate stimuli similar to that in a real system. Should be noted that primary requests in real microprocessor are consequences of some memory access operation (loading, storing data, eviction, prefetch, atomic swap, etc). Secondary requests are answers for reaction packets from the device. It is usually convenient to use only a sequence of primary requests as a test sequence, and generate secondary requests automatically in corresponding modules. Properties of secondary requests could be changed based on secondary request generation modules configuration.

In the test system interfaces are combined into groups that represent working with some devices. A test system should simulate the state of these devices to generate correct responses from it.

3.2 Generation of Primary Requests for Caches with Out-of-order Execution

Properties of the devices that support out-of-order execution should be considered while designing a stimulus generator:

- Order of primary request can be different from the order of memory accesses in initial program
- Primary request could be divided into several messages accepted at different times. Messages for one primary request are identified by common value of tag field
- Request canceling mechanism is present.

To support out-of-order execution of memory access requests in a cache common approach was augmented. The module responsible for transferring of primary requests was replaced with high-level module that includes components working with interfaces of primary request parts. The order of the request for the module is identical to that of the test program, and reordering of request parts is executed based on module settings.

4. Correctness Checking

Let us consider the existing approaches to reaction checking. Richard Ho suggested two main methods: self-checking tests and co-simulation [9]. Co-simulation is a method for reaction checking in which an independent reference model is used along with the target design model [4]. The two models are co-simulated using the same stimuli and their reactions are verified.

A reference model is implemented either in general purpose programming language (C, C++) or in specialized hardware verification language (SystemVerilog, “e”, Vera). If test stimuli are the same, difference in model and device reactions means an error somewhere in the system[8]. Reference models could be cycle-accurate or untimed functional. To implement the cycle-accurate model, behavior of the device must be specified on a register transfer level. Behavior of caches usually defined on a higher level of abstraction, because cache is not an essential part of a computational pipeline of a microprocessor. A cache is not a subject of strict temporal requirements. Besides this, the development of cycle-accurate model is labor-intensive when the design specification is changing and no stable through the verification phase. To simplify the development of reference models TLM (Transaction Level Modeling) is often used [4]. To verify caches we also propose to implement functional models working on transaction level.

4.1 Checking of nondeterministic caches

If one wants to develop functional model of cache, its specification must have property of transaction level indeterminism. That is, identical transaction level traces of stimuli (a set of RTL traces is mapped into this single transaction level trace) must cause identical transactional reaction trace. It should be noted that caches often include a set of components (eviction arbiter, primary request arbiter serving different requesters), that do not hold that property. That is, different RTL traces that are mapped into the single transaction level trace could lead to different reaction traces. There are several methods to check the behavior of nondeterministic devices.

4.1.1 “Gray box” checking

One of the ways to solve aforementioned problem is to replace usual “black box” method of device verification. That is, we should not consider only external interfaces of the device while analysing its behavior. To determine which variant of

behavior was happened in the cache one could use “hints” from the implementation. To use this approach, a set of internal interfaces and signals is defined and its behavior is specified. This interfaces must be chosen in a way that information on their state could be used to eliminate nondeterminism. In general, for caches such signals are the results of primary request arbitration and the interfaces of finite automata of the cache eviction mechanism. Additionally, that information can be used in a request generator and for the estimation of verification quality. This method is usually easy to implement. Drawbacks of this methods are additional requirements for specification and reliance on interfaces that could also exhibit erroneous behaviour.

4.1.2 Dynamic refinement of transaction level model

Another approach is to create additional instances of model for each variant of behavior in case of nondeterministic choice in the device [6]. Each reaction is checked against every spawned device model. If reaction is impossible for one variation of behavior, then it is removed from set. If set of possible states after some reaction becomes empty, the system must return an error. In general, this approach may cause exponential growth of number of states with each consecutive choice. However, for caches it could be implemented efficiently, because of several properties of caches: serialization of requests and cache line independence. Information on which nondeterministic choice was made in the device (for use in a request generator or for verification quality estimation) could also be extracted from reactions. The strong point of the approach compared to “gray box” method is elimination of reliance on implementation details of the device. Drawback is additional complexity of implementation.

4.1.3 Assertions

A test system generator imitates an environment of DUV. It also should be noticed that interaction between the device and its environment must adhere to some protocol. Based on that protocol, we can include functional requirements of protocols as an assertions in the generator. Then, violation of an assertion signals an error. Usage of assertions is an effective method of detection of a broad class of errors. In addition, to assertions that are common for all memory subsystem devices, several cache-specific assertions could be included. They represent invariants of cache coherence protocol. To check this invariants, coherence of states of a single cache line is analyzed in all parts of test system after each change.

4.2 Checking caches with out-of-order execution

Caches that support out-of-order request execution exhibit properties of limited nondeterminism. That is the memory access request are received in the device in

multiple parts from several interfaces, with different unspecified timing characteristics. On the other hand, there is the "reference" order of memory access operations presented in original test program. If out-of-order execution introduces error to the canonical order, device must be cleaned and erroneous transactions must be restarted. Results of operations that completed successfully are deterministic. Based on these properties of the device, we propose to implement models of two types:

- "Ignoring the cancelled transactions" mode
- Strict checking mode

In the first mode the result of checking is delayed until the moment of the request full completion. If completion was unsuccessful, checks are not made. In the strict mode we use the approaches which is similar to the dynamic refinement of model. Set of possible device states is maintained, and it is augmented with each stimulus and reaction. The number of possible states is limited by the number of simultaneously executed out-of-order requests. Shortcomings of the first mode are delays between erroneous transaction and the execution of actual checking and reduction of the set of errors that could be detected (for example, unnecessary cancel of request will not be detected). On the other hand, implementing that mode is much simpler task, so verification could be started sooner.

5. Case Study

The approaches described above were used for stand-alone verification of L2-cache[3] and the L3-cache[6] of the microprocessor with "Elbrus" architecture and L1Data-cache (L1dc) of the microprocessor with "SPARC-V9" architecture. The test systems for stand-alone verification of this caches were developed using Universal Verification Methodology (UVM) [10].

5.1 Checking the "SPARC-V9" L1Data-cache with out-of-order execution

L1dc supports out-of-order execution of memory access operations. The test system structure for L1DC is presented in fig. 1.

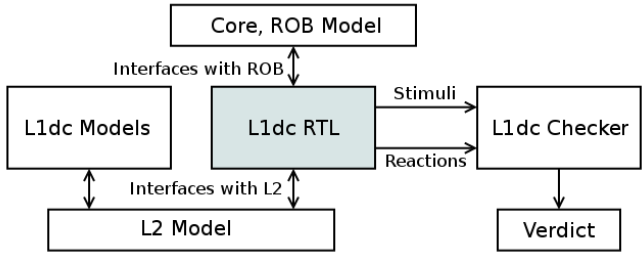


Fig. 1. The principal structure of the test sytem forL1Data-cache of the"SPARC-V9" microprocessor.

The test sequences for L1dc are the memory access assembly instructions. They are sent to computational core and the reordering buffer (Core, ROB Model). In this module, the instructions are split into multiple messages (containing either operation type, address or data). These messages are reordered and sent to DUV. Additionally, this module keeps information about initial order of instructions received, to send completion messages in correct order.

5.2 Checking the "Elbrus" L3-cache with nondeterministic behavior

The test stimulus generator was developed to verify the L3-cache of the "Elbrus" microprocessor[6]. It is based on simplified model of microprocessor core with the L2-cache and the model of system commutator that simulates work in multiprocessor environment. If multiple cores request access to a single cache line, then the order of their execution is unspecified and defined by the device microarchitecture. Internal structure of a cache is also a subject of change, due to changes to requirements of physical design. To verify the device the approach based on dynamic refinement of behavioral model was chosen. To supplement that approach, a set of assertions were implemented in stimulus generator to check validity of the system state. The approach allowed using the same test system with minimal alteration for the next iteration of the "Elbrus" microprocessor.

6. Conclusion

The approaches described in this paper allows avoiding some shortcomings. It could help to avoid excessive subdivision of the verified unit on small subdevices and developing cycle-accurate models of them (as we done in our previous projects) on the one hand and the development and maintaining of complex cycle-accurate reference models of caches on the other. The approaches were used for stand-alone verification of caches of microprocessors developed by MCST. Stand-alone verification allowed finding several errors in different caches. The intermediate results of application introduced approaches in the multicore microprocessor caches verification if presented in table 1. We already had verified the L3-cache of the "Elbrus" microprocessor using another approach and we could find new 7 errors more with help of developed tests system based on nondeterministic caches checking approach.

	Verified caches		
	L2-cache "Elbrus"	***L3-cache "Elbrus"***	***L1 data cache "SPARC-V9"***
Number of bugs	4	7	12

The test systems are developed as a UVM-environment. They were implemented to be flexible enough to set both the pseudorandom and directed test sequences. Using

of aforementioned approaches while developing test systems helped find some new errors and simplify the test system development. Approaches could be used to verify other caches of different multicore microprocessors regardless of its architectures.

Our future research is connected with improving the error diagnostics and localization of found bugs.

References

[1]. Sorin D.J., Hill M.D., Wood D.A. A Primer on Memory Consistency and Cache Coherence. Morgan and Claypool, 2011, 195 p.

[2]. Bergeron J. Writing testbenches: functional verification of HDL models. Boston: Kluwer Academic Publishers, 2003.

[3]. Stotland I, Meshkov A., Kutsevol V. Standalone functional verification of multicore microprocessor memory subsystem units based on application of memory subsystem models. Proc. of IEEE East-West Design & Test Symposium (EWDTS 2015), Batumi, Georgia, September 26-29, 2015, pp. 326-330.

[4]. Kamkin A., Chupilko M. A TLM-based approach to functional verification of hardware components at different abstraction levels. Proc. of the 12th Latin-American Test Workshop (LATW), 2011, pp. 1-6.

[5]. Tessier T., Lin H., Ringoen D., Hickey E., Anderson S. Designing, verifying and building an advanced L2 cache sub-system using SystemC. Proc. of Design and Verification Conference (DV-CON), 2012, pp.1-8.

[6]. Kamkin A., Petrochenkov M. A Model-Based Approach to Design Test Oracles for Memory Subsystems of Multicore Multiprocessors. Trudy ISP RAN / Proc. ISP RAS, vol. 27, issue 3, pp. 149-157.

[7]. TLM-2.0.1. TLM Transaction-Level Modeling Library. (online publication). Available at: http://www.accellera.org/downloads/standards/systemc (accessed 20.05.2016).

[8]. Stotland I., Lagutin A. Using stand alone behavioural models to verify microprocessor components. Voprosy radioelektroniki, seriya EVT [Issues of radio electronics], 2014, issue 3, pp. 17-27.

[9]. Ho R. Validation tools for complex digital designs. PhD thesis, Standford University, 1996.

[10]. Standard Universal Verification Methodology (online publication). Available at: http://accellera.org/downloads/standards/uvm (accessed 20.05.2016).

Подходы к автономной верификации кэш-памятей многоядерных микропроцессоров

[1] *М.В. Петроченков <petroch_m@mcst.ru>*
[1] *И.А. Стотланд <stotl_i@mcst.ru>*
[1] *Р.Е. Муштаков <mushtakov_r@mcst.ru>*

[1] *АО «МЦСТЭ», 117105, Россия, г. Москва, ул. Нагатинская, д.1, стр.1*

Аннотация. В статье приведен обзор методов, применяемых при проверке корректности поведения кэш-памятей многоядерных микропроцессоров. Описаны общие свойства устройств подсистемы памяти микропроцессора, а также свойства, специфичные для кэш-памятей, и метод поддержки согласованности состояния памяти в системе на основании протокола когерентности. Представлены подходы к проектированию тестовой системы, генерации корректных тестовых воздействий и проверке правильности поведения тестируемого устройства. Предложены модификации общего подхода к генерации тестовых воздействий для устройств с внеочередным исполнением инструкций. Приведены способы разработки тестовых систем на различных уровнях абстракции. В статье описан основной способ проверки поведения устройства на уровне транзакций — разработка эталонной поведенческой модели для последующего сравнения реакций устройства с эталонными; расхождения в реакциях сигнализируют об ошибке. Выделены критерии применимости данного подхода. Описаны методы верификации устройств, поведение которых функционально не детерминировано на уровне транзакций: метод «серого ящика», базирующийся на анализе внутренних интерфейсов устройства, для устранения возникающей неопределенности в поведении устройства. Кроме того, приведен новый метод, основанный на динамическом уточнении поведенческой модели на основе реакции устройства. Также рассмотрены преимущества использования утверждений утверждения в генераторе тестовых воздействий в качестве дополнительных методов обнаружения ошибок. В работе приведен метод, позволяющий упростить проверку поведения устройств с внеочередным исполнением инструкций, основанный формировании эталонной очереди их выполнения. В заключение представлены результаты применения предложенных подходов к верификации кэш-памятей многоядерных микропроцессоров архитектуры «Эльбрус» и «SPARC-V9».

Ключевые слова: многоядерный микропроцессор; кэш-память; внеочередное исполнение; тестовая система; недетерминированное поведение; верификация на основе эталонных моделей; автономная верификация; «SPARC-V9»; микропроцессор «Эльбрус».

DOI:10.15514/ISPRAS-2016-28(3)-10

Для цитирования: Петроченков М.В., Стотланд И.А., Муштаков Р.Е. Подходы к автономной верификации кэш-памятей многоядерных микропроцессоров. Труды ИСП РАН, том 28, вып. 3, 2016 г., стр. 161-172 (на английском). DOI: 10.15514/ISPRAS-2016-28(3)-10.

Список литературы

[1]. Sorin D.J., Hill M.D., Wood D.A. A Primer on Memory Consistency and Cache Coherence. Morgan and Claypool, 2011. 195 p.

[2]. Bergeron J. Writing testbenches: functional verification of HDL models. Boston: Kluwer Academic Publishers, 2003.

[3]. Stotland I, Meshkov A., Kutsevol V. Standalone functional verification of multicore microprocessor memory subsystem units based on application of memory subsystem models. Proc. of IEEE East-West Design & Test Symposium (EWDTS 2015), Batumi, Georgia, September 26-29, 2015, pp.326-330.

[4]. Kamkin A., Chupilko M. A TLM-based approach to functional verification of hardware components at different abstraction levels. Proc. of ehe 12th Latin-American Test Workshop (LATW), 2011, pp. 1-6.

[5]. Tessier T., Lin H., Ringoen D., Hickey E., Anderson S. Designing, verifying and building an advanced L2 cache sub-system using SystemC. Proc. of Design and Verification Conference (DV-CON), 2012, pp.1-8.

[6]. Kamkin A., Petrochenkov M. A Model-Based Approach to Design Test Oracles for Memory Subsystems of Multicore Multiprocessors. Trudy ISP RAN, vol. 27, 3, p 149-157.

[7]. TLM-2.0.1. TLM Transaction-Level Modeling Library. (online). Доступно по ссылке: http://www.accellera.org/downloads/standards/systemc (дата обращения 20.05.2016).

[8]. Стотланд И.А., Лагутин А.А. Применение эталонных событийных моделей для автономной верификации модулей микропроцессоров. // Вопросы радиоэлектроники, сер. ЭВТ, 2014, вып. 3, стр. 17-27.

[9]. Ho R. Validation tools for complex digital designs.PhD thesis, Standford University, 1996.

[10]. Standard Universal Verification Methodology (online). Доступно по ссылке: http://accellera.org/downloads/standards/uvm (дата обращения 20.05.2016).

Инструменты математического сервиса MathPartner для выполнения параллельных вычислений на кластере

Е.А. Ильченко <ilchenkoea@gmail.com>
Тамбовский государственный университет имени Г.Р. Державина,
392 000, г.Тамбов, ул. Интернациональная, 33

Аннотация. Во многих прикладных областях необходимо выполнять символьно-численные расчеты с данными большого объема. Примерами таких областей являются робототехника, распознавание речи, распознавание графической информации, автоматизация производства и другие. Системы символьных вычислений, их так же называют системами компьютерной алгебры, активно развиваются с конца восьмидесятых годов. Хорошо известными системами являются Mathematica, Maple, Reduce и многие другие. Почти все эти системы не были ориентированы изначально ни на масштабные математические объекты, ни на многопроцессорные кластеры. Система Form является единственным исключением. Эта система была изначально задумана для оперирования объектами, превышающими по размеру оперативную память. Такие объекты размещаются на жестком диске. В статье дается описание алгоритмов для тех инструментов системы компьютерной алгебры MathPartner, которые предназначены для взаимодействия с вычислительным кластером. Приводится описание алгоритма работы сокетного сервера, являющегося связующим звеном между MathPartner и некоторой супер ЭВМ, который обеспечивает исполнение параллельных программ на кластере. Подробно объясняется механизм, который позволяет абстрагироваться от конкретной супер ЭВМ и установленной на нее PBS, работая исключительно с веб-интерфейсом MathPartner. Кроме запуска готовых программ, описываемый сокетный сервер дает возможность запускать пользовательские программы, отправляемые на вычислительный кластер в виде zip-архива через веб-интерфейс. В статье даются примеры использования уже реализованных параллельных алгоритмов, которые входят в состав веб сервиса MathPartner. Некоторые из параллельных программ MathPartner реализованы с помощью парадигмы «DDP» (dynamic decentralized parallelization) – подхода, позволяющего написать эффективную параллельную программу для работы с неоднородными данными, такими как разреженные матрицы. В статье показаны примеры использования DDP-программ, интегрированных в MathPartner.

Ключевые слова: параллельный алгоритм; облачная математика; MathPartner; веб-интерфейс; сокетный сервер.

DOI: 10.15514/ISPRAS-2016-28(3)-11

Для цитирования: Ильченко Е.А. Инструменты математического сервиса MathPartner для выполнения параллельных вычислений на кластере. Труды ИСП РАН, том 28, вып. 3, 2016 г., стр. 173-188. DOI: 10.15514/ISPRAS-2016-28(3)-11

1. Введение

При разработке системы компьютерной алгебры MathPartner ставилась задача обеспечения вычислений с масштабными символьно-численными математическими объектами и проведения вычислений на многопроцессорном вычислительном кластере. Сегодня этот облачный математический сервис свободно доступен на платформе проекта «Университетский кластер» на сайте http://mathpar.cloud.unihub.ru.

Сервис MathPartner содержит библиотеку символьно-численных алгоритмов, написанную на Java, а также пакеты параллельных программ, которые выполняются на вычислительном кластере [5]-[7]. Кроме того, MathPartner предоставляет возможность загружать и исполнять на кластере пользовательские программы, которые были предварительно загружены на него с помощью web-интерфейса.

В данном сообщении описываются алгоритмы взаимодействия web-сервиса MathPartner и вычислительного кластера, который управляется системой PBS.

2. Взаимодействие MathPartner с PBS

Portable Batch System (PBS) – система управления распределенными вычислениями. Основная функция PBS - запуск задач в вычислительной среде. Эта система имеет консольный интерфейс. Как правило, для взаимодействия с ней достаточно двух команд - установка программы в очередь для последующего выполнения и проверка ее состояния, чтобы выяснить, когда программа закончила работу. Команда **`qsub`** используется для установки программы в очередь на выполнение, где **`config`** – это имя файла с настройками запуска. К настраиваемым параметрам относится количество требуемых процессоров, число требуемых узлов, путь к файлам для сохранения потока вывода и потока ошибок и путь к запускаемой программе.

Для взаимодействия с PBS в MathPartner был разработан комплекс программных средств. Управляющий узел кластера с пакетом PBS и сервер, на котором установлен веб-сервис MathPartner, – это разные устройства. Они связываются по сети Интернет. Программа, которая обеспечивает их связь, устанавливается на управляющий узел кластера. С веб-сервисом MathPartner она взаимодействует через сокетное соединение.

Для взаимодействия с PBS Java-программа использует класс **`Runtime`**, в частности его метод **`exec(String[] command)`**. Это программный аналог обычной отправки команды в терминал. Программа «PBSbridge» выполняет следующие функции:

- Запуск уже готовых программ, которые входят в библиотеку MathPartner. Для этого необходимо, чтобы на управляющем узле кластера располагалась копия Java-классов MathPartner, находящихся на веб-сервере.
- Обеспечение возможности запуска пользовательских параллельных программ, написанных на Java, с использованием MPI. При этом должно осуществляться копирование скомпилированных Java-классов на управляющий узел кластера и последующий их запуск. Для этого создан специальный интерфейс, который позволяет производить копирование файлов, запуска программы на выполнение, отслеживание состояния запущенной программы и отображение результатов выполнения.
- Разграничение пользователей друг от друга и запрет доступа к файловой системе кластера. Выполнение задач от разных пользователей осуществляется в разных потоках.

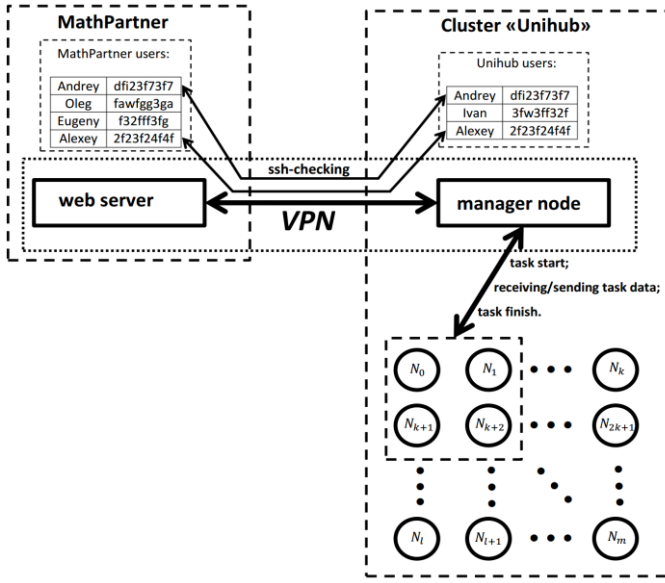


Рис. 1. Общая схема взаимодействия web-сервиса MathPartner и кластера Unihub.

Fig. 1. The general scheme of interaction of MathPartner and Unihub.

Поскольку необходимо разграничить всех пользователей, необходимо обеспечить их регистрацию в системе. Для этого в MathPartner добавлена процедура аутентификации пользователя. Для хранения данных используется внешняя СУБД. После того, как пользователь вошел в систему, он может использовать команды MathPartner для работы с PBSbridge. Вся система работает по принципу запрос-ответ. Пользователь вводит в интерфейс MathPartner команду, после чего соответствующий ей запрос отправляется по сокетному соединению программе PBSbridge. Выполняются действия, соответствующие этому запросу, и результат возвращается по тому же соединению обратно.

Опишем общую схему работы программы PBSbridge. Класс, содержащий **`main`**-метод для запуска, имеет имя **`Server`**. На рис.2 представлен алгоритм метода **`main`**, с которого начинается выполнение программы.

В начале метода **`main`** закрываются потоки ввода-вывода, чтобы предупредить возможность существования чужих открытых потоков. После этого создаются необходимые директории и файл для записи лог-файла. При этом все настройки берутся из констант, которые определены в классе **`AlgorithmsConfig`**. Метод **`writeLog`** служит для записи лог-файла с отчетом. После этого запускается бесконечный цикл, в котором происходит прослушивание порта, указанного в настройках сокета. Если происходит соединение, то для обработки запросов, приходящих по этому соединению, создается новый поток выполнения.

```
//Закрытие потоков ввода-вывода
System.in.close();
System.out.close();
//создание файла, в который будет записываться отчет работы программы
logFile=new FileWriter(AlgorithmsConfig.CNF_DATA_PATH+"/log.txt", false);
//создание серверного сокета
ServerSocket server = new ServerSocket(AlgorithmsConfig.CNF_SERV_PORT);
//создание структуры для хранения состояний задач
taskStates=new TreeMap<Integer,TreeMap<Integer,Integer>>();
writeLog("Server successfully started");
while true do
 |  new Server(server.accept());
end
```

*Рис. 2. Алгоритм **`main`**-метода класса **`Server`**.*

*Fig. 2. **`Main`**-method algorithm of class **`Server`**.*

Файлы пользователей хранятся на сервере. Для каждого пользователя создается своя папка, в качестве имени берется идентификатор пользователя (**`id`**) из базы данных. Программа PBSbridge не имеет доступа к базе, поэтому идентификатор пользователя **`id`** приходит вместе со всеми запросами на сервер.

Обеспечивается возможность осуществления одного из двух режимов. Можно либо запускать свои собственные задачи, предварительно загруженные на сервер, либо запускать задачи, являющиеся частью MathPartner.

Для каждой задачи создается своя папка, которая находится внутри папки данного пользователя. Например, они могут располагаться вот так: **`../userX/taskY`**. Перед запуском задачи в папке будут находиться 2 файла: файл с настройками запуска параллельной программы и файл с входными данными для запускаемой программы.

Содержимое файла с настройками запуска описывается спецификацией, которая используется PBS системой. В файле с входными данными хранится массив входных данных в виде сериализованных объектов. После того как программа завершит работу, в этой папке появятся еще 3 файла, которые являются результатами вычислений. Это будет массив сериализованных объектов, содержащих результаты вычислений, файл, содержащий стандартный поток вывода, и файл, содержащий сообщения об ошибках. Содержимое каждого из этих файлов может быть получено с помощью соответствующих команд интерфейса MathPartner.

Поскольку для каждой задачи создается несколько файлов, необходимо каким-то образом позаботиться об их удалении. Это можно сделать следующим образом: для каждой задачи будем запоминать время последнего обращения к ее файлам. Если разница между текущим временем на сервере и пометкой какой-либо задачи больше, чем, например, двое суток, то эти файлы можно считать устаревшими, и они подлежат удалению.

Этот механизм реализован в виде еще одного потока, который раз в сутки перебирает все папки, проверяя даты последнего обращения к ним. Также необходимо удалять пользовательские загруженные файлы. Для них можно использовать механизм, описанный выше, но требуется увеличить продолжительность хранения этих файлов, например, до 30 дней.

Взаимодействие с программой PBSbridge осуществляется с помощью запросов, которые могут идти либо с веб-части, либо от того узла кластера, с которого началось выполнение на кластере текущей задачи.

Необходимо иметь механизм получения входных данных на ведущем узле кластера, которые передаются с управляющего узла, и механизм для отправки результата вычислений на управляющий узел. Для этого служат соответствующие запросы к программе PBSbridge. Формат запросов следующий: сначала идет целое число, являющееся идентификатором запроса, а потом данные. Программа PBSbridge считывает это число, являющееся идентификатором, и, в зависимости от его значения, принимает решение о получении дополнительных данных. Количество этих данных для каждого запроса может быть различно.

Обработка всех запросов происходит в цикле, работа которого завершится только тогда, когда сокетное соединение будет разорвано. На рис. 3 приведен алгоритм такого цикла, где показана реализация обработки запроса

QS_ADD_TASK, который служит для запуска программ. Он универсален, потому что используется для запуска как готовых MathPartner-алгоритмов, так и пользовательских загруженных программ. Настройки запуска передаются в виде сериализованного объекта класса **TaskConfig**. В случае успешного запуска пользователю возвращается номер, присвоенный текущей задаче при запуске, он будет использоваться для отслеживания состояния задачи и для получения ее результатов.

Непосредственное создание файла с настройками запуска для PBS осуществляется классом **Launcher**. Он создает обычный текстовый файл, учитывая все настройки, которые были указаны в запросе от web-части, устанавливает права на исполнение для этого файла и отправляет соответствующую команду для запуска. На рис. 4 приведен алгоритм такого метода.

Параллельная программа может иметь 4 состояния: она может находиться в очереди на выполнение, в процессе выполнения, вычисления программы может быть успешно завершены, работа программы может быть завершена с ошибками.

Далее рассмотрим механизм отслеживания состояния задачи. После того как завершится выполнение метода **CreateAndRunPBSfile**, задаче присваивается состояние «в очереди». Для запуска задачи на счет должны быть получены входные данные с управляющего узла кластера. Это происходит по запросу **QS_GET_DATA_FOR_CALC**. Задаче присваивается состояние «в процессе выполнения».

```
//инициализация потоков ввода-вывода:
InputStream is = socket.getInputStream();
ObjectInputStream inp=new ObjectInputStream(is);
OutputStream os = socket.getOutputStream();
ObjectOutputStream oos=new ObjectOutputStream(os);
//создание класса, с помощью которого мы будем запускать задачи:
Launcher launcher =new Launcher();
while true do
    //получение идентификатора входящего запроса:
    Integer qType=(Integer)inp.readObject();
    Integer userID;
    //выполнение действий для соответствующего запроса
    switch qType do
        case AlgorithmsConfig.QS_ADD_TASK
            //получение id пользователя
            userID=(Integer)inp.readObject();
            //получение данных и настроек для алгоритма
            Object []data=(Object[])inp.readObject();
            TaskConfig taskConf=(TaskConfig)inp.readObject();
            //номер задачи, который присваивается данной задаче:
            Integer taskNumb=addTaskInMap(userID);
            //запуск задачи
            Integer launchResult=launcher.launch(userID, taskNumb, data, taskConf);
            //если по каким-то причинам задачу не удалось запустить, удаляем записи о ней:
            if launchResult!=AlgorithmsConfig.RES_SUCCESS then
                removeTaskFromMap(userID, taskNumb);
            end
            //возвращение результата запуска задачи и присвоенного ей номера:
            oos.writeObject(launchResult);
            oos.writeObject(taskNumb);
        end
        case AlgorithmsConfig.QS_GET_DATA_FOR_CALC
            //далее действия для обработки этого запроса
            ...
        end
        //далее идет обработка прочих запросов
        ...
    endsw
end
```

Рис. 3. Алгоритм обработки поступающих запросов.

Fig. 3. Algorithm for processing incoming requests.

Возможны 2 варианта: либо задача успешно завершается и результаты пересылаются в PBSbridge с помощью запроса **QS_RECV_RESULT_FOR_TASK_CLUSTER**, либо задача аварийно завершает свою работу и в этом случае она не возвращает результат.

```
//файл, в который мы запишем настройки для PBS:
FileWriter runFile;
String run_file_path = folderPath+"/run";
File dir = new File(run_file_path);
runFile = new FileWriter(dir);
//запись строки настроек в файл запуска:
runFile.append("Здесь строка запуска, учитывающая все необходимые настройки");
//команда для изменение прав доступа к запускающему файлу:
String []chMod="chmod", "777", run_file_path;
Process chmod = Runtime.getRuntime().exec(chMod);
//команда запуска команды qsub:
String []command="/opt/pbs/bin/qsub", run_file_path;
Process qsub = Runtime.getRuntime().exec(command);
```

Рис. 4. Алгоритм создания файла с настройками для PBS и его последующего исполнения системой запуска.

Fig. 4. File creation algorithm with settings for PBS and its execution of the launch system.

Если задача аварийно завершит работу, то результат вычислений не будет отсылаться на сервер, и ее состояние будет по-прежнему «в процессе выполнения». Но в файл, содержащий поток ошибок, будет записан протокол появления исключительной ситуации (Exception), которая привела к остановке выполнения программы. Если задача завершила работу успешно, то файл с сообщением об ошибках будет пустым.

Поэтому при каждом запросе с веб-части о состоянии задачи производится проверка этого файла, и в случае, если он не пуст, то пользователю возвращается соответствующее сообщение, а задаче присваивается статус «аварийное завершение».

3. Параллельные алгоритмы в MathPartner

MathPartner содержит параллельные программы, которые должны выполняться на кластере с использованием web-интерфейса. Часть алгоритмов реализована с использованием парадигмы **DDP** – децентрализованного динамического управления параллельным вычислительным процессом. Эта схема управления позволяет эффективно использовать узлы кластера, даже когда входные данные для алгоритма имеют неоднородную структуру (разреженные матрицы). В табл. 1 представлен перечень имеющихся на данный момент параллельных программ в составе MathPartner.

Для любых параллельных алгоритмов можно производить настройку запуска на кластере. Описание этих настроек приведено в таблице 2. Все переменные здесь могут принимать только целые положительные значения.

На рис. 6 показан пример выполнения оператора **\matMultPar1x8**, который выполняет параллельное блочно-рекурсивное умножение двух матриц. Сначала выполняются настройки всех необходимых переменных для запуска,

затем инициализируются матрицы-сомножители *A* и *B*, после этого запускается задача умножения матриц на кластере. Сообщение «`Task ID is 1`» означает, что задача успешно помещена в очередь выполнения на кластере и ее идентификатор равняется единице. Информацию о состоянии задачи можно получить с помощью оператора **`\getStatus`**, передав ему в качестве аргумента полученный **`Task ID`**. Сообщение «`Task is finished`» означает, что задача успешно завершила свою работу. Результат выполнения программы можно получить с помощью команды **`\getCalcResult(taskID)`**.

В том случае, когда входные данные имеют большой объем, в MathPartner предусмотрен файловый импорт и экспорт математических объектов (матрицы, векторы, функции, полиномы). На рис. 7 показан пример матричного умножения, когда входные данные вводятся из файлов пользователя, с последующим сохранением результата в файле. Файлы *`matrixA.txt`* и *`matrixB.txt`* содержат текст «`[[1,2],[3,4]]`» и «`[[4,5],[6,7]]`», соответственно, файл *`matrixC.txt`* содержит результат «`[[16,19],[36,43]]`». Перед тем, как выполнять операциями с файлами их необходимо сначала загрузить на веб-сервер, как показано на рис. 5.

Табл. 1. Параллельные программы в составе MathPartner.

Название алгоритма	Оператор mathpar	Реализовано с помощью **DDP**
Матрицы и полиномы		
Матричное умножение [13]	**`\matMultPar1x8(A,B)`**	да
Вычисление присоединенной матрицы	**`\adjointPar(A)`**	нет
Умножение полиномов	**`\polMultPar(A,B)`**	да
Тропические вычисления		
Решение однородного уравнения Беллмана $Ax=x$ [16]	**`\BellmanEquationPar(A)`**	да
Решение однородного уравнения Беллмана $Ax+b=x$ [16]	**`\BellmanEquationPar(A,b)`**	да
Решение однородного неравенства Беллмана $Ax \leq x$ [16]	**`\BellmanInequalityPar(A)`**	да
Решение однородного неравенства Беллмана $Ax+b \leq x$ [16]	**`\BellmanInequalityPar(A,b)`**	да

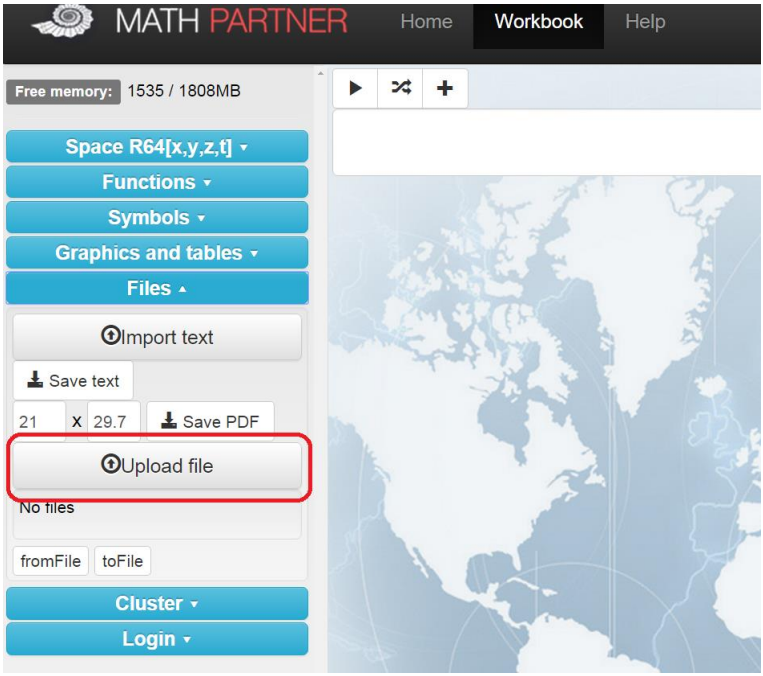


Рис. 5. Загрузка пользовательского файла на сайт.

Fig. 5. Downloading the user file to the site.

Табл. 2. Описание переменных для настроек запуска параллельных программ.

Имя переменной	Назначение
\TOTALNODES	Количество узлов кластера, на которых будет запущена программа
\PROCPERNODES	Количество программ, которые будут запущены на одном узле кластера (количество MPI-процессов для одного узла)
\CLUSTERTIME	Максимальное время работы программы, превысив которое она будет аварийно завершена
\MAXCLUSTERMEMORY	Количество памяти, доступное для JVM для одного MPI-процесса

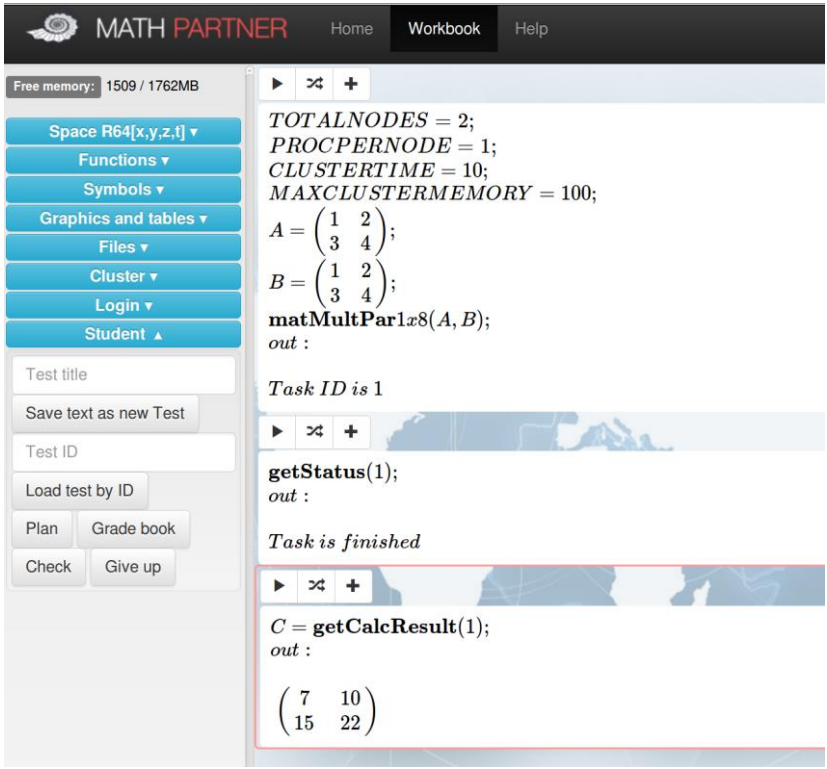


Рис. 6. Пример запуска параллельного матричного умножения с настройками параметров запуска.

Fig. 6. Example of running a parallel matrix multiplication with setting startup options.

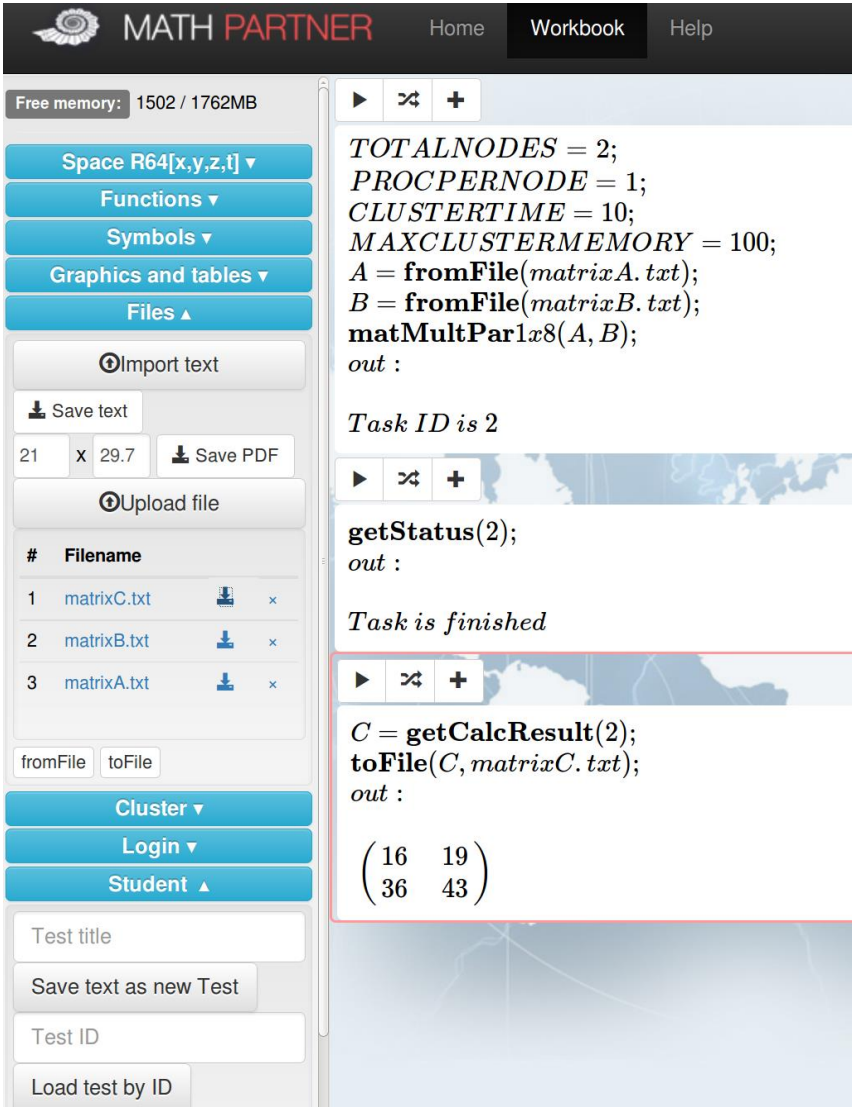


Рис. 7. Пример запуска параллельного матричного умножения с получением исходных данных из файлов.

Fig. 7. Example of running a parallel matrix multiplication where data is taken from the files.

4. Запуск параллельных программ, разработанных пользователем в проекте MathPartner

MathPartner предоставляет удобный интерфейс для запуска на кластере новых параллельных программ, разработанных пользователем в проекте MathPartner на языке Java. Процесс запуска такой параллельной программы состоит из следующих шагов:

- Необходимо создать zip-архив пакета скомпилированных java-классов.
- Выполнить вход в MathPartner.
- Загрузить архив на сайт MathPartner (рис. 5).
- Загрузить архив с программой на кластер с помощью команды **`\uploadToCluster(fileName.zip)`**.
- Поставить программу в очередь на исполнение с помощью команды **`\runUploadedClass(fileName.zip, PackageWithClass.ClassWithMainMethod, param1, param2, ...).`**

После выполнения описанных шагов пользователю будет сообщен идентификатор задачи, с помощью которого будет осуществляться дальнейшая работа. Команда **`\getStatus(taskID)`** позволяет узнать состояние задачи (в очереди, в процессе запуска, завершена, завершена аварийно). Для отображения содержимого потоков вывода и ошибок используются команды **`\getOut(programId)`** и **`\getErr(programId)`**.

5. Заключение

Были изложены алгоритмы взаимодействия веб-сервиса MathPartner с вычислительным кластером, на котором установлена система PBS.

Эти алгоритмы обеспечивают решение двух задач. Одна из них – это выполнение параллельных программ, входящих в состав MathPartner. Другая – это загрузка и выполнение на кластере параллельных программ пользователя, которые разработаны им в проекте MathPartner.

Дальнейшее развитие пакета параллельных программ в проекте MathPartner может в всецело опираться на приведенные в данной работе алгоритмы.

Приведен список параллельных прогамм, входящих в текущую версию системы MathPartner.

Описанные алгоритмы можно применить и для других веб-сервисов, когда требуется обеспечить взаимодействие с кластером, на котором установлена система BPS.

Список литературы

[1]. Strassen V. Gaussian Elimination is not optimal. Numerische Mathematik. 13, 1969, pp. 354-356.

[2]. Малашонок Г.И. Матричные методы вычислений в коммутативных кольцах. Тамбов: Изд-во Тамбовского университета, 2002. 213 с.

[3]. Малашонок Г.И. О вычислении ядра оператора действующего в модуле. Вестник Тамбовского университета. Сер. Естественные и технические науки, 2008, том 13, вып. 1, стр. 129-131.

[4]. Gennadi Malaschonok and Evgeni Ilchenko. Decentralized control of parallel computing. International conference Polynomial Computer Algebra. St.Petersburg, PDMI RAS, 2012, pp. 57-58.

[5]. Бетин А.А. Эксперименты с параллельным алгоритмом вычисления присоединённой матрицы и параллельным умножением файловых матриц. Вестник Тамбовского университета. Сер. Естественные и технические науки, Тамбов, 2010, том 15, вып. 1, стр. 341-345.

[6]. Бетин А.А. Эксперименты с параллельным алгоритмом вычисления присоединённой матрицы. Вестник Тамбовского университета. Сер. Естественные и технические науки, 2010, том 15, вып. 6, стр. 1748-1754.

[7]. Малашонок Г.И. Компьютерная математика для вычислительной сети. Вестник Тамбовского университета. Сер. Естественные и технические науки, 2010, том 15, вып. 1, стр. 322-327.

[8]. Малашонок Г.И. Управление параллельным вычислительным процессом. Вестник Тамбовского университета. Сер. Естественные и технические науки, 2009, том 14, вып. 1, стр. 269-274.

[9]. Малашонок Г.И., Валеев Ю.Д. Организация параллельных вычислений в рекурсивных символьно-численных алгоритмах. Труды конференции ПаВТ'2008 (Санкт-Петербург). Челябинск: Изд-во ЮУрГУ, 2008, стр. 153-165.

[10]. Г.И. Малашонок, Ю.Д. Валеев. Рекурсивное распараллеливание символьно-численных алгоритмов. Вестник Тамбовского университета. Сер. Естественные и технические науки, 2006, том 11, вып. 4, стр. 536-549.

[11]. Г.И. Малашонок, Ю.Д. Валеев. О некоторых подходах к построению параллельных программ. Вестник Тамбовского университета. Сер. Естественные и технические науки, 2005, том 10, вып. 1, стр. 154-156.

[12]. Malaschonok G.I. Effective Matrix Methods in Commutative Domains. Formal Power Series and Algebraic Combinatorics. Berlin: Springer, 2000, pp. 506-517.

[13]. Е.А. Ильченко. Об эффективном методе распараллеливания блочных рекурсивных алгоритмов. Вестник Тамбовского университета. Сер. Естественные и технические науки, 2015, том 20, вып. 5, стр. 1173-1186.

[14]. О.Н. Переславцева. Параллельный алгоритм вычисления характеристического полинома и его временная сложность. Вестник Тамбовского университета. Сер. Естественные и технические науки, 2014, том 19, вып. 2, стр. 530-538.

[15]. Д.С. Ивашов. Параллельный алгоритм разложения многочленов на множители с различными наборами переменных. Вестник Тамбовского университета. Сер. Естественные и технические науки, 2014, том 19, вып. 2, стр. 558-565.

[16]. С.А. Киреев, Г.И. Малашонок. Тропические вычисления в веб-сервисе MathPartner. Вестник Тамбовского университета. Сер. Естественные и технические науки, 2014, том 19, вып. 2, стр. 539-550.

Tools of mathematical service MathPartner for parallel computations on a cluster

E.A. Ilchenko <ilchenkoea@gmail.com>
Tambov State University, Internatsionalnaya, 33, RU-392000, Tambov, Russia

Abstract. In many application areas it is necessary to perform symbolic-numerical calculations with a large volume of data. Examples of such areas are robotics, speech recognition, recognition of graphical information, automation and others. Symbolic computation systems, they also called computer algebra system, actively developed since the late eighties. Well-known systems are Mathematica, Maple, Reduce, and many others. Almost all of these systems were not originally focused any large-scale mathematical objects or on multiprocessor clusters. System FORM is a unique exception. It was conceived as a system which can operate with objects exceeding RAM. Such objects are placed on the hard drive. We give a description of such algorithms of MathPartner web services, which are designed to interact with a computing cluster. We give an algorithm to work a socket server, which is the link between MathPartner and super computers, and which provides the execution of parallel programs on a cluster. We explain in detail the mechanism which abstracts the specific features of super computers and the installed PBS package. The user can run on the cluster or program of MathPartner package, or their own programs. To run its own programs, they are able to send the compiled classes to the computing cluster in a zip-archive through the MathPartner web interface. We show examples of using parallel algorithms included in MathPartner package. Some of MathPartner parallel programs implemented with the paradigm of **DDP** (dynamic decentralized parallelization). DDP is designed as a framework that allows to write efficient parallel program for working with nonhomogeneous data such as sparse matrix. We demonstrate examples of using DDP-programs that are integrated into MathPartner.

Keywords: Parallel algorithm; Cloud mathematics; MathPartner; Web interface; the socket server.

DOI: 10.15514/ISPRAS-2016-28(3)-11

For citation: Ilchenko E.A. Tools of mathematical service MathPartner for parallel computations on a cluster. *Trudy ISP RAN / Proc. ISP RAS*, vol. 28, issue 3, 2016. pp. 173-188 (in Russian). DOI: 10.15514/ISPRAS-2016-28(3)-11.

References

[1]. Strassen V. Gaussian Elimination is not optimal. Numerische Mathematik. 13, 1969, pp. 354-356.

[2]. Malaschonok G. Matrix calculation methods in commutative domains. Tambov: Izd-vo Tambovskogo universiteta [Tambov University publishing], 2002, 213 p. (in Russian).

[3]. Malaschonok G. On computation of kernel of operator acting in a module. Vestnik Tambovskogo universiteta. Ser. Estestvennye i tekhnicheskie nauki [Tambov University Reports. Series: Natural and Technical Sciences], 2008, vol. 13, issue 1, pp. 129-131 (in Russian).

[4]. Gennadi Malaschonok and Evgeni Ilchenko. Decentralized control of parallel computing. International conference Polynomial Computer Algebra. St.Petersburg, PDMI RAS, 2012, pp. 57-58.

[5]. Betin A. Experiments with a parallel algorithm for calculation of adjoint matrix and with a parallel algorithm for multiplication of file matrices. Vestnik Tambovskogo universiteta. Ser. Estestvennye i tekhnicheskie nauki [Tambov University Reports. Series: Natural and Technical Sciences, 2010, vol. 15, issue 1, pp. 341-345 (in Russian).

[6]. Betin A. Experiments with a parallel algorithm for calculation of adjoint matrix. Vestnik Tambovskogo universiteta. Ser. Estestvennye i tekhnicheskie nauki [Tambov University Reports. Series: Natural and Technical Sciences], 2010, vol. 15, issue 6, pp. 1748-1754 (in Russian).

[7]. Malaschonok G. Computer mathematics for computational network. Vestnik Tambovskogo universiteta. Ser. Estestvennye i tekhnicheskie nauki [Tambov University Reports. Series: Natural and Technical Sciences], 2010, vol. 15, issue 1, pp. 322-327 (in Russian).

[8]. Malaschonok G. Managing of the parallel calculative process. Vestnik Tambovskogo universiteta. Ser. Estestvennye i tekhnicheskie nauki [Tambov University Reports. Series: Natural and Technical Sciences], 2009, vol. 14, issue 1, pp. 269-274 (in Russian).

[9]. Malaschonok G., Valeev U. Organization of parallel computations in recursive symbol-numerical algorithms. Trudy konferencii PaVT'2008 [Proceedings of conference PCT'2008] (St. Petersburg). Chelyabinsk: Publishing house SUSU, 2008, pp. 153-165 (in Russian).

[10]. Malaschonok G., Valeev U. Recursive disparallelizing of symbol-numerical algorithms. Vestnik Tambovskogo universiteta. Ser. Estestvennye i tekhnicheskie nauki [Tambov University Reports. Series: Natural and Technical Sciences, 2006, vol. 11, issue. 4, pp. 536-549 (in Russian).

[11]. Malaschonok G., Valeev U. On some approaches to the construction of the parallel program. Vestnik Tambovskogo universiteta. Ser. Estestvennye i tekhnicheskie nauki [Tambov University Reports. Series: Natural and Technical Sciences], 2005, vol. 10, issue 1, pp. 154-156 (in Russian).

[12]. Malaschonok G.I. Effective Matrix Methods in Commutative Domains. Formal Power Series and Algebraic Combinatorics. Berlin: Springer, 2000, pp. 506-517.

[13]. Ilchenko E. About effective methods of parallelizing block recursive algorithms. Vestnik Tambovskogo universiteta. Ser. Estestvennye i tekhnicheskie nauki [Tambov University Reports. Series: Natural and Technical Sciences], 2015, . 20, issue. 5, pp. 1173-1186 (in Russian).

[14]. Pereslavceva O. Parallel algorithm for computing the characteristic polynomials of polynomial matrices and algorithm’s computational time. Vestnik Tambovskogo universiteta. Ser. Estestvennye i tekhnicheskie nauki [Tambov University Reports. Series: Natural and Technical Sciences], 2014, vol. 19, issue 2, pp. 530-538 (in Russian).

[15]. Ivaschov D. Parallel algorithms factorizationof polinomialswith different sets of variables. Vestnik Tambovskogo universiteta. Ser. Estestvennye i tekhnicheskie nauki [Tambov University Reports. Series: Natural and Technical Sciences], 2014, vol. 19, issue 2, pp. 558-565 (in Russian).

[16]. Kireev S., Malaschonok G. Tropical calculations in web service MathPartner. Vestnik Tambovskogo universiteta. Ser. Estestvennye i tekhnicheskie nauki [Tambov University Reports. Series: Natural and Technical Sciences], 2014, vol. 19, issue 2, pp. 539-550 (in Russian).

Верификация и анализ вариабельных операционных систем[1]

[1,2,3] *В.В. Кулямин <kuliamin@ispras.ru>*
[1,4] *Е.М. Лаврищева <lavr@ispras.ru>*
[1] *В.С. Мутилин <mutilin@ispras.ru>*
[1,2,3] *А.К. Петренко <petrenko@ispras.ru>*

[1] *Институт системного программирования РАН,*
109004, Россия, г. Москва, ул. А. Солженицына, д. 25
[2] *Московский государственный университет имени М.В. Ломоносова,*
119991, Россия, Москва, Ленинские горы, д. 1
[3] *НИУ Высшая школа экономики,*
101000, Россия, Москва, ул. Мясницкая, д. 20
[4] *Московский физико-технический институт (гос. университет),*
141700, Россия, Московская обл., г. Долгопрудный, Институтский пер., д. 9

Аннотация. В данной работе рассматриваются проблемы верификации и анализа сложных операционных систем с учетом их вариабельности, или наличия большого количества разнообразных конфигураций. Исследуются методы, позволяющие преодолеть эти проблемы, проводится их обзор и классификация. Выделены классы методов, использующих для анализа инструменты, не учитывающие вариабельность, и выборки вариантов системы и методов, использующих специализированные инструменты, учитывающие вариабельность. Как наиболее перспективные с точки зрения масштабируемости, выделены техники анализа, использующие выборки вариантов системы, обеспечивающие покрытие ее кода и комбинаций значений конфигурационных параметров, а также специализированные, учитывающие вариабельность кода техники анализа с итеративным уточнением модели поведения системы на основе контрпримеров.

Ключевые слова: операционная система; семейство систем; модель вариабельности; верификация; статический анализ; проверка моделей; проверка типов; покрытие кода; покрывающий набор; итеративное уточнение модели на основе контрпримеров.

DOI: 10.15514/ISPRAS-2016-28(3)-12

[1] Работа поддержана грантом Российского фонда фундаментальных исследований № 16-01-00352.

Для цитирования: Кулямин В.В., Лаврищева Е.М., Мутилин В.С., Петренко А.К. Верификация и анализ вариабельных операционных систем. Труды ИСП РАН, том 28, вып. 3, 2016 г., стр. 189-208. DOI: 10.15514/ISPRAS-2016-28(3)-12

1. Введение

В современном мире программное обеспечение (ПО) используется для решения все более ответственных и сложных задач, что обуславливает постоянный рост сложности самих программ. При этом процессы разработки, анализа и сопровождения ПО также усложняются, и требуются специальные меры для снижения темпов роста их стоимости и трудозатрат. Одним из методов снижения затрат на создание и сопровождение сложных систем, решающих большое количество разнообразных задач, является создание *вариабельных систем* (или *семейств систем*, software product families, software product lines) [1-4]. Экономия здесь достигается на том, что разработчики пытаются сразу создавать многократно используемые элементы нескольких систем с близким набором функций, предназначенных для различных групп пользователей, сокращая таким образом затраты на создание всех этих систем в совокупности. На сегодняшний день предложены многочисленные техники разработки вариабельных систем различных типов, включая операционные системы (ОС).

При разработке вариабельных систем важнейшую роль играют *модель вариабельности* (variability model) и *механизм обеспечения вариабельности* (variation mechanism). Модель вариабельности задает пространство возможных вариантов данного семейства систем. Обычно она определяется набором характеристик (features) или конфигурационных параметров, множествами их возможных значений и ограничениями на возможные комбинации этих значений, каждый вариант системы при этом соответствует некоторому набору значений всех характеристик. Механизм вариабельности обеспечивает возможность построения всех возможных вариантов системы из ограниченного набора создаваемых и сопровождаемых артефактов.

Механизмы обеспечения вариабельности достаточно разнородны, но могут быть разделены на три группы [5]: *интеграционные*, основанные на отдельной разработке элементов для различных вариантов и дальнейшей их интеграции в разных комбинациях, *генеративные*, использующие параметризацию для генерации различных вариантов одного элемента из общей основы, и *смешанные*, определенным образом объединяющие обе эти техники. В области операционных систем (здесь и далее под операционной системой мы понимаем ядро и базовые библиотеки ОС, предоставляющие приложениям интерфейсы для работы с вычислительными ресурсами и аппаратным обеспечением) в качестве механизма вариабельности смешанного типа широко используется механизм условной компиляции языков C/C++ (на основе макросов #ifdef, #elif, #else). Он позволяет на этапе сборки составлять код, объединяющий различные элементы, задаваемые набором значений

характеристик, являющихся в этом случае параметрами условной компиляции (определяемыми с помощью макросов #define и #undef, а также параметров запуска препроцессора).

Сложность моделей вариабельности современных операционных систем очень высока, например, ядро Linux версии 2.6.32 имеет 6319 характеристик, более 10000 ограничений, которые могут задействовать до 22-х отдельных характеристик, при этом большинство характеристик зависит как минимум от 4-х других, а максимальная глубина дерева зависимостей равна 8 [6]. Такая сложность приводит к большому количеству ошибок, связанных, прежде всего, с трудностями учета всех факторов, которые должен принимать во внимание разработчик отдельного элемента кода. Соответственно, для выявления и преодоления этих ошибок необходимо использовать специализированные техники анализа и верификации. Сложности анализа, характерные для систем с таким механизмом вариабельности, возникают из-за огромного размера пространства допустимых вариантов (что делает совершенно нереалистичным проверку их всех) и одновременной невозможности проверки отдельных фрагментов, из которых собирается код системы. Поскольку используется условная компиляция, каждый фрагмент не обязан являться отдельным компонентом с определенным поведением, которое можно было бы проанализировать отдельно от остального кода, обычно такие фрагменты являются лишь вставками в общий код, и могут быть проверены лишь в определенных комбинациях друг с другом. Необходимость решения этих проблем накладывает на инструменты и методы анализа, применяемые для сложных вариабельных операционных систем и системного ПО вообще, особые требования. Эти требования специфичны именно для анализа и верификации — методы, применяемые для создания таких систем, сами по себе не облегчают их анализ [7,8]. Основная цель данного исследования — определение и развитие масштабируемых методов анализа вариабельных операционных систем, позволяющих справиться с описанными проблемами.

Далее мы кратко рассматриваем основные элементы моделей вариабельности и используемые в области системного ПО языки для их описания, проводим обзор методов анализа и верификации сложного вариабельного системного ПО и выделяем из описанных в литературе методов наиболее перспективные для дальнейшего развития.

2. Модели вариабельности и языки их описания

Для продуктивной работы по созданию или сопровождению определенной вариабельной системы (семейства систем) необходимо понимание ее модели вариабельности, описывающей все многообразие вариантов системы (систем, входящих в данное семейство). Модели вариабельности операционных систем по существу не отличаются от моделей вариабельности, используемых для систем других типов. Обычно в рамках такой модели один вариант системы

соответствует некоторому набору значений определенных характеристик или конфигурационных параметров, поэтому модель вариабельности также часто называется моделью характеристик [9] или моделью конфигураций. Сама модель при этом описывается множеством используемых характеристик, множествами возможных значений для каждой характеристики и набором ограничений на допустимые комбинации значений характеристик. Часть характеристик соответствует видимым пользователю функциям системы, а другая часть - различным альтернативным способам реализации этих функций, а также решениям, принятым при разработке системы и не имеющим непосредственного влияния на ее восприятие пользователями (только косвенное). Для операционных систем примерами характеристик могут быть: поддержка работы с сетью, поддержка многих потоков, поддержка отладки системных процессов и пр.

Большое количество характеристик обычно имеют булевские значения (характеристика может быть включена или выключена в данном варианте), реже встречаются числовые значения или строки, чаще всего одна характеристика может иметь лишь небольшое конечное множество значений. Довольно часто встречаются ограничения, предписывающие использовать значения одной характеристики только при включенной другой или нескольких других (первая характеристика зависит от второй, первая имеет смысл только при включенной второй, например, возможность удаленного доступа к системному журналу может иметься только при включенной поддержке работы с сетью и включенной поддержке ведения журнала). Чаще всего из нескольких выбирается одна, наиболее значимая зависимость, что позволяет оформлять подобного рода ограничения в виде структуры дерева (иерархии) зависимостей на характеристиках. В этом дереве зависимые характеристики считаются дочерними по отношению к той, от которой они зависят, корнем дерева считается некоторая абстрактная характеристика (соответствующая самому рассматриваемому семейству систем), промежуточными узлами чаще всего становятся только булевские характеристики (которые можно включить/выключить), а характеристики с другими множествами значений могут быть только листьями этого дерева. Изредка небулевские характеристики тоже могут быть промежуточными узлами дерева, но при этом во множестве их значений должно быть выделенное значение, соответствующее «выключению» этой характеристики, остальные значения при этом означают разные варианты ее «включения». Довольно часто встречаются также ограничения вида «включение/исключение»: включение одной характеристики должно всегда сопровождаться включением другой и/или выключением нескольких других характеристик.

Известно достаточно много языков для описания моделей характеристик [10,11], предоставляющих наглядные средства для описания указанных выше широко встречающихся видов ограничений и зависимостей.

В области операционных систем (и системного ПО вообще) для этих целей чаще всего [12] используются языки Kconfig [13,14], применяемый для описания возможных конфигураций ядра Linux с 2002 г., и CDL (Component Definition Language) [15], используемый в рамках открытой операционной системы реального времени eCos [16] для встроенных систем. Оба языка поддерживают все основные элементы метода FODA (Feature-Oriented Domain Analysis) [9], описанные выше. Однако поддержка сложных ограничений, связывающих характеристики, не являющиеся детьми одного родителя в дереве зависимостей, в инструментах, работающих с Kconfig, несколько хуже, что часто приводит к ошибкам при описании и обработке сложных конфигураций [6]. Детальное сопоставление возможностей обоих языков и обзор их использования на практике приведены в [12], на Рис. 1 показан пример описания небольшой модели на обоих языках, взятый из [12].

```
k-1  menuconfig MISC_FILESYSTEMS
k-2    bool "Miscellaneous filesystems"
k-3
k-4   if MISC_FILESYSTEMS
k-5
k-6    config JFFS2_FS
k-7     tristate "Journalling Flash File System" if MTD
k-8     select CRC32 if MTD
k-9
k-10
k-11
k-12
k-13   config JFFS2_FS_DEBUG
k-14    int "JFFS2 Debug level (0=quiet, 2=noisy)"
k-15    depends on JFFS2_FS
k-16    default 0
k-17    range 0 2
k-18    --- help ---
k-19      Debug verbosity of ...
k-20
k-21
k-22   config JFFS2_FS_WRITEBUFFER
k-23    bool
k-24    depends on JFFS2_FS
k-25    default HAS_IOMEM
k-26
k-27
k-28   config JFFS2_COMPRESS
k-29    bool "Advanced compression options for JFFS2"
k-30    depends on JFFS2_FS
k-31
k-32   config JFFS2_ZLIB
k-33    bool "Compress w/zlib..." if JFFS2_COMPRESS
k-34    depends on JFFS2_FS
k-35    select ZLIB_INFLATE
k-36    default y
k-37
k-38   choice
k-39    prompt "Default compression" if JFFS2_COMPRESS
k-40    default JFFS2_CMODE_PRIORITY
k-41    depends on JFFS2_FS
k-42    config JFFS2_CMODE_NONE
k-43     bool "no compression"
k-44    config JFFS2_CMODE_PRIORITY
k-45     bool "priority"
k-46    config JFFS2_CMODE_SIZE
k-47     bool "size (EXPERIMENTAL)"
k-48   endchoice
k-49  endif
```

```
c-1  cdl_component MISC_FILESYSTEMS {
c-2   display    "Miscellaneous filesystems"
c-3   flavor     none
c-4  }
c-5
c-6  cdl_package CYGPKG_FS_JFFS2 {
c-7    display     "Journalling Flash File System"
c-8    requires    CYGPKG_CRC
c-9    implements  CYGINT_IO_FILEIO
c-10   parent      MISC_FILESYSTEMS
c-11   active_if   MTD
c-12
c-13   cdl_option CYGOPT_FS_JFFS2_DEBUG {
c-14    display          "Debug level"
c-15    flavor           data
c-16    default_value    0
c-17    legal_values     0 to 2
c-18    define           CONFIG_JFFS2_FS_DEBUG
c-19    description      "Debug verbosity of...."
c-20   }
c-21
c-22   cdl_option CYGOPT_FS_JFFS2_NAND {
c-23    flavor           bool
c-24    define           CONFIG_JFFS2_FS_WRITEBUFFER
c-25    calculated       HAS_IOMEM
c-26   }
c-27
c-28   cdl_component CYGOPT_FS_JFFS2_COMPRESS {
c-29    display          "Compress data"
c-30    default_value    1
c-31
c-32      cdl_option CYGOPT_FS_JFFS2_COMPRESS_ZLIB {
c-33       display          "Compress data using zlib"
c-34       requires         CYGPKG_COMPRESS_ZLIB
c-35       default_value    1
c-36      }
c-37
c-38      cdl_option CYGOPT_FS_JFFS2_COMPRESS_CMODE {
c-39       display          "Set the default compression mode"
c-40       flavor           data
c-41       default_value    { "PRIORITY" }
c-42       legal_values     { "NONE" "PRIORITY" "SIZE" }
c-43     }
c-44    }
c-45 }
c-46
c-47
c-48
c-49
```

Рис. 1. Описание одной модели на Kconfig (слева) и CDL (справа).

Fig. 1. Description of a model in Kconfig (left) and CDL (right).

2.1. Методы анализа моделей вариабельности

Языки описания моделей вариабельности предоставляют широкий набор возможностей по заданию структуры характеристик и ограничений на их значения. Ограничения чаще всего описываются в виде иерархии и

зависимостей, представленных как логические выражения. Такие ограничения на практике могут быть довольно сложными, что приводит к разного рода ошибкам в моделях вариабельности, например, к противоречиям и неразрешимым зависимостям, из-за которых некоторые характеристики невозможно включить. Для преодоления этих трудностей предназначен автоматический анализ моделей вариабельности, который помогает выявить противоречия и несогласованности в самой модели или проверить допустимости заданной конкретной конфигурации (т.е., что конфигурация удовлетворяет всем ограничениям модели).

Методы анализа моделей вариабельности являются темой активных исследований. Наиболее развитые средства анализа имеются для языков моделирования вариабельности, разрабатываемых в рамках исследовательских проектов [12,17]. Их можно разделить на четыре основных группы [10].

- Анализ на основе пропозициональной логики.
  В рамках этого подхода ограничения модели характеристик транслируются в представляющие их логические формулы, которые затем анализируются с помощью решателей, инструментов для автоматического доказательства различных видов (на основе SAT, BDD и пр.) или инструментов для работы с формальными языками типа Alloy, B или Z. Такие методы довольно широко распространены, их обзор можно найти в [18].
- Анализ на основе онтологий.
  Этот подход использует трансляцию модели вариабельности в модель онтологии. Например, в [19] производится трансляция в OWL DL (Ontology Web Language Description Logic), разрешимое подмножество языка OWL, обладающее достаточной выразительной мощностью. После трансляции становится возможным использовать автоматизированные инструменты анализа онтологий [20], такие как RACER [21].
- Анализ на основе программирования в ограничениях.
  В этом подходе ограничения модели вариабельности транслируются в описание задачи CSP (Constraint Satisfaction Problem), которая затем анализируется с помощью существующих инструментов программирования с ограничениями (constraint programming). См., например, работы [22,23].
- Анализ на основе проверки моделей.
  Некоторые исследователи используют преобразование ограничений модели характеристик в задачи проверки моделей (model checking), которые затем решаются соответствующими инструментами [24,25].
- Применение специализированных алгоритмов.
  Некоторые исследователи предлагают узко специализированные

алгоритмы для решения конкретных задач анализа моделей характеристик, например, для оценки числа допустимых вариантов. Описание таких подходов можно найти в [10].

В инструментах, работающих с практически важными языками Kconfig и CDL, возможности анализа моделей весьма ограничены. Для Kconfig поиск несогласованностей и недопустимости конфигураций выполняются автоматически только для моделей с ограничениями, не включающими характеристики, не имеющие общего родителя. При использовании более общих ограничений их противоречия и соответствие им конфигураций автоматически не выявляются. Пользователи должны вручную отслеживать соблюдение такого рода ограничений.

В CDL имеется поддержка анализа непротиворечивости конфигурации, основанная на пропозициональной логике. При модификации конфигурации инструмент анализа определяет противоречия и подсказывает пользователю способы их разрешения.

В работах [10,12] показано, что для обоих языков Kconfig и CDL нет поддержки более развитых подходов анализа моделей вариабельности, доступных для исследовательских языков, почти неиспользуемых в промышленных проектах. Это говорит о наличии возможностей для существенного развития применяемых на практике языков описания конфигураций системного ПО.

3. Методы верификации и анализа вариабельных ОС

Как уже было сказано, основные проблемы анализа и верификации сложного вариабельного системного ПО связаны с невозможностью получить значимую информацию, анализируя комбинируемые в рамках вариантов фрагменты кода по отдельности, из-за отсутствия у них свойств, переносимых на поведение самих вариантов ПО, и с неспособностью проверить каждый из вариантов ПО в силу их огромного количества. Решать эти проблемы можно одним из двух способов.

- Использовать без модификаций те же инструменты и методы анализа, что и для анализа согласованного кода (одного варианта системы). При этом подвергнуть анализу можно только отдельные варианты, но не все (всех слишком много), соответственно, возникает задача выбора небольшого представительного подмножества из всего пространства допустимых вариантов. Эта задача аналогична задаче выбора небольшого, но представительного множества ситуаций, которые будут использоваться в тестах, из всего гигантского множества ситуаций, возможных при работе тестируемой системы. Вторая задача обычно решается при помощи выбора так называемого *критерия полноты* тестирования, или *критерия покрытия*. Так же и при решении первой логично сформулировать некоторый *критерий покрытия пространства вариантов*, достижение которого (т.е.,

проведение анализа для набора вариантов, удовлетворяющего этому критерию) по некоторым причинам можно считать достаточным для выявления всех существенных свойств и ошибок, характерных для всего набора возможных вариантов. Идеи, лежащие в основе выбора такого критерия, могут быть различными, но требования к нему такие же, как и к критерию полноты тестирования. Он должен давать, с одной стороны, возможность выявить на удовлетворяющем ему наборе вариантов все существенные особенности поведения систем из семейства и ошибки, и, с другой, возможность выбрать достаточно маленькое множество вариантов, удовлетворяющих этому критерию, чтобы их полный анализ был практически осуществим в рамках ограничений проекта на трудоемкость и стоимость.

Такие методы можно назвать *анализом выборки вариантов* (sample-based analysis). В обзоре [26] подобные методы названы нацеленными на продукт (product-based).

- Другой способ – модификация методов и инструментов анализа с целью поддержки ими работы с несколькими вариантами одновременно, т.е., внесение в их работу таких изменений, которые позволяют анализировать свойства сразу нескольких вариантов проверяемой системы, используя для экономии усилий близость большинства вариантов друг к другу. В этом случае по-прежнему нельзя надеяться на возможность проверки сразу всех вариантов системы - их слишком много, чтобы можно было учесть все возможные варианты поведения в рамках одного анализа. Однако, можно выделять для каждого выполнения анализа более крупные куски кода, которые уже могут быть компонентами (или группами компонентов) с четко выделенным интерфейсом и определенным поведением, анализируя которое можно адекватно выявлять свойства системы. Поэтому при этом подходе также возникает потребность в критерии покрытия, но не в пространстве допустимых вариантов, а на множестве компонентов системы с их точками вариации - будем называть его далее *критерием покрытия вариаций*. Этот критерий также должен определенным образом гарантировать выявление всех существенных свойств и ошибок вариантов системы и обеспечить возможность выбора лишь небольшого количества групп, каждая из которых в итоге будет подвергнута анализу. Заметим, что особенности используемых инструментов и поведения различных вариантов системы могут потребовать, чтобы эти группы пересекались как по компонентам, так и по точкам вариации.

  Подобные методы иногда называют *учитывающим вариабельность анализом* (variability-aware analysis) [27]. В [26] эти методы названы нацеленными на семейство (family-based).

Обзор описанных в литературе методов анализа вариабельных операционных систем показывает, что оба указанных подхода используются в исследованиях и на практике, однако более активно развиваются методы первого типа, использующие критерии покрытия пространства вариантов. Техники второго типа возникли относительно недавно и пока не достигли масштабируемости, необходимой для поддержки анализа промышленной ОС, такой как Linux.

В обзоре [26] выделен еще один вид методов анализа семейств систем – анализ, нацеленный на характеристики (feature-based), проводимый таким образом, чтобы выявить свойства всех вариантов, обладающих заданным значением выделенной характеристики, безотносительно остальных. Однако используемый для системного ПО механизм вариабельности (условная компиляция) крайне усложняет проведение подобного анализа при наличии многих характеристик, поскольку специфичные только для заданной характеристики свойства очень трудно выделить на фоне сложного поведения, определяемого большим числом характеристик и их взаимосвязями. Поэтому такая разновидность анализа сложного системного ПО практически не встречается и обсуждаться в данной работе не будет.

Еще один обзор инструментов анализа вариабельных систем можно найти в [28]. Некоторую информацию из этого обзора мы используем в дальнейшем, но он, по большей части, рассматривает инструменты анализа вне той системы понятий, которая требуется нам.

3.1. Методы анализа на основе выборки вариантов

Известные методы анализа выборки вариантов делятся на следующие группы [27].

- Использование одной «наиболее представительной» конфигурации. В этом случае пытаются выбрать одного представителя проверяемого семейства систем, обладающего как можно большим количеством характерных свойств или включающего как можно больше кода. Чаще всего для выбора такого варианта необходимо привлечение экспертов. В сообществе разработчиков ядра Linux в этом качестве принято использовать конфигурацию allyesconfig, в которой большинство характеристик включено [29].

  Обычно при достаточно высокой сложности модели вариабельности анализ только одного варианта неспособен дать достаточно полную информацию о возможных свойствах всего семейства. Этому часто мешают многочисленные взаимоисключающие характеристики и более сложные ограничения на возможные комбинации их значений. Поэтому в сложных случаях выбор «наиболее представительной» конфигурации может быть достаточно хорошим началом для построения набора вариантов для анализа, но всегда должен дополняться другими вариантами.

- Использование случайных конфигураций.
  Для анализа может использоваться (псевдо-)случайно построенный набор допустимых конфигураций. Для Linux есть инструмент их построения, randconfig, часто используемый при необходимости получить случайные конфигурации [30]. Иногда построение конфигураций, удовлетворяющих всем ограничениям, нетривиально и требует применения сложных эволюционных алгоритмов [31].
  Случайный выбор набора вариантов не гарантирует представительности их поведения для всего анализируемого семейства или достижения определенного критерия покрытия. Тем не менее, он может использоваться как затравка набора вариантов для дальнейшего пополнения на основе выбранного критерия покрытия.
- Использование критериев покрытия кода.
  Критерии покрытия вариантов, использующие покрытие кода, учитывают, насколько в выбранном наборе вариантов представлены различные фрагменты кода, которые могут быть включены или исключены при построении допустимого варианта системы (также могут учитываться различные возможные комбинации таких фрагментов). Эти критерии отличаются от критериев покрытия кода, используемых при тестировании, — первые нацелены на выбор некоторого набора вариантов, содержащих какие-то комбинации фрагментов кода, а вторые на выполнение определенных частей кода. Наиболее широко используемым критерием такого рода является требование того, чтобы каждый (актуальный) фрагмент кода из репозитория программного семейства входил хотя бы в один из вариантов, отобранных для анализа. Несмотря на то, что это не самый сильный критерий — при его использовании могут быть не выявлены ошибки, проявляющиеся только при определенных комбинациях варьируемых фрагментов, — даже его достижение в сложных случаях сопряжено со значительными затратами. Теоретически задача построения минимального такого набора вариантов сводится к некоторой NP-полной задаче [27], поэтому на практике лучше использовать алгоритмы, строящие неоптимальные наборы [32], но даже они не всегда дают удовлетворительный результат. В работе [33] для выбора набора конфигураций ядра Linux, покрывающих как можно больше кода, использовался специализированный инструмент VAMPYR, который смог получить максимальное покрытие кода 91% для архитектуры mips (для архитектур x86 и arm удалось получить всего 88% и 84%). Использование одной конфигурации allyesconfig дало для этой архитектуры покрытие 55% кода (соответственно, 79% и 60%), что показывает, также, насколько на практике одна, даже «самая представительная» конфигурация может мало затрагивать возможные варианты поведения систем семейства. Для повышения

этих значений могут понадобиться гораздо более сложные подходы, способные, например, обеспечить включение кода, выделенного несколькими директивами #if/#ifdef.

- Использование критериев покрытия комбинаций значений характеристик.

  Один из способов задействовать закрываемый условными директивами код – использовать соответствующую комбинацию значений характеристик. Это соображение, а также более общая идея, что, реализуя различные комбинации значений характеристик, можно добиться проявления всех возможных вариантов поведения, служат обоснованием выбора набора вариантов для анализа на основе образуемого ими покрытия различных допустимых комбинаций значений характеристик из модели вариабельности. Математической основой таких методов служат алгоритмы построения *покрывающих наборов* (covering arrays) [34-36], которые часто используются в конфигурационном тестировании [37,38]. Покрывающий набор глубины *t* определяет матрицу значений, в которой столбцы соответствуют характеристикам (значениями в столбце могут быть только допустимые значения соответствующей характеристики), строки – выбираемым вариантам, и каждая комбинация *t* значений в любых разных *t* столбцах обязательно встречается в одной из строк. Простейшим случаем является покрывающий набор глубины 2 (или попарный, pairwise), позволяющий покрыть в рамках набора вариантов, задаваемого его строками, все сочетания пар значений характеристик [39]. Задача построения минимального покрывающего набора NP-полна, но существуют эффективные алгоритмы построения наборов, лишь немного больших, чем минимальные [35-38]. Для сложных моделей вариабельности большей проблемой является удовлетворение всех налагаемых моделью ограничений, поэтому техники создания покрывающих наборов должны дополняться достаточно эффективными методами построения или выбора удовлетворяющих ограничениям конфигураций [40,41].

Как видно, исходя из нацеленности на анализ как можно более широкого набора вариантов поведения, наибольшие перспективы для дальнейшего развития среди методов анализа на основе выборки вариантов имеют методы, основанные на покрытии кода и комбинаций значений характеристик. Вполне возможно создание гибридных техник, совмещающих использование комбинаторной генерации и отслеживание достигаемого покрытия кода.

3.1. Методы анализа, учитывающие вариабельность

Методы анализа и верификации, учитывающие вариабельность проверяемой системы, пытаются проводить проверку сразу многих (или даже всех

возможных) вариантов одновременно, сокращая суммарные затраты на нее за счет большого объема общего кода во всех вариантах.

Учет вариабельности требует при реализации инструментов анализа использования модифицированных деревьев абстрактного синтаксиса, размеченных условиями использования тех или иных узлов, представляющими собой обычно пропозициональные формулы над равенствами характеристик и их возможных значений [27,42]. Опубликованные методы такого рода обычно относятся к методам проверки типов (type safety checking, well-formedness checking) [43-46] или методам проверки моделей (model checking) [47-51], есть также несколько работ, использующих дедуктивную верификацию (theorem proving) [52] и символический мониторинг [53].

Методы анализа, учитывающие вариабельность, обладают двумя существенными недостатками (по сравнению с основанными на выборке вариантов): их применение нуждается в аккуратном использовании полной информации о модели вариабельности, которая часто представлена не только в коде, а и в конфигурационных файлах, и в настройках инструментов сборки, а также требует значительной доработки и усложнения инструментов анализа, чтобы те могли учитывать все используемые характеристики и ограничения.

Наиболее перспективно с точки зрения масштабируемости из этих методов выглядят техники проверки типов, дающие возможность эффективно проверять простейшие свойства корректности, и техники проверки моделей с их итеративным уточнением, [51] требующие минимального вмешательства человека при анализе большого по объему кода со сложным поведением. Остальные методы становятся чрезмерно сложными при учете реалистичных моделей вариабельности.

4. Заключение

В статье рассмотрены задачи верификации и анализа современных промышленных операционных систем с учетом их вариабельности, или наличия большого числа возможных конфигураций. Основные проблемы, стоящие в этой области, — невозможность в разумные сроки провести анализ всех допустимых вариантов системы и невозможность выявить значимые свойства при анализе отдельных фрагментов кода, из которых собираются эти варианты.

Методы анализа, способные справиться с этими проблемами, делятся на две группы: анализ некоторой выборки вариантов из всех возможных и анализ кода с учетом его вариативности. На основе проведенного обзора таких методов, учитывая большой объем кода и сложность современного системного ПО, а также нацеленность на проведение как можно более полного анализа поведения всех конфигураций системы, мы выбрали несколько методов, наиболее перспективных для дальнейшего развития. Ими являются хорошо масштабируемые техники, использующие выборку

вариантов на базе покрытия кода и/или на базе покрытия различных комбинаций значений конфигурационных параметров, а также техники анализа кода, использующие итеративное уточнение моделей на основе контрпримеров.

Список литературы

[1]. Jacobson I., Griss M., Jonsson P. Software Reuse, Architecture, Process and Organization for Business Success. Addison-Wesley, 1997. ISBN-13: 978-0201924763.

[2]. Bosch J. Design and Use of Software Architectures: Adopting and Evolving a Product Line Approach. Pearson Education, 2000. ISBN-13: 978-0201674941.

[3]. Clements P., Northrop L. Software Product Lines: Practices and Patterns. SEI Series in Software Engineering, Addison-Wesley, 2001. ISBN-13: 978-0201703320.

[4]. Pohl K., Böckle G., van der Linden F. J. Software Product Line Engineering: Foundations, Principles and Techniques. Springer-Verlag, 2005. DOI: 10.1007/3-540-28901-1.

[5]. Bachmann F., Clements P. Variability in software product lines. CMU/SEI Technical Report CMU/SEI-2005-TR-012, 2005.

[6]. Lotufo R., She S., Berger T., Czarnecki K., Wąsowski A. Evolution of the Linux kernel variability model. Proc. of SPLC'10, LNCS 6287:136-150, Springer, 2010. DOI: 10.1007/978-3-642-15579-6_10.

[7]. Лаврищева Е. М., Коваль Г.И., Слабоспицькая О.О., Колесник А.Л. Особенности процессов управления при создании семейств программных систем. Проблемы программирования, (3):40-49, 2009.

[8]. Лаврищева Е.М., Слабоспицькая О.А., Коваль Г.И., Колесник А.А. Теоретические аспекты управления вариабельностью в семействах программных систем. Вестник КНУ, серия физ.–мат. наук, (1):151-158, 2011.

[9]. Kang K., Cohen S., Hess J., Novak W., Peterson S. Feature-oriented domain analysis (FODA) feasibility study. CMU/SEI Technical Report CMU/SEI-90-TR-21, 1990.

[10]. Benavides D., Segura S., Ruiz-Cortés A. Automated analysis of feature models 20 years later: a literature review. Information Systems, 35(6):615–636, 2010. DOI: 10.1016/j.is.2010.01.001.

[11]. Chen L., Babar M.A. A systematic review of evaluation of variability management approaches in software product lines. Information and Software Technology, 53(4):344–362, 2011. DOI: 10.1016/j.infsof.2010.12.006.

[12]. Berger T., She S., Lotufo R., Wąsowski A., Czarnecki K. A study of variability models and languages in the systems software domain. IEEE Transactions on Software Engineering, 39(12):1611-1640, 2013. DOI: 10.1109/TSE.2013.34.

[13]. Zippel R. et al. Kconfig language. https://www.kernel.org/doc/Documentation/kbuild/kconfig-language.txt.

[14]. She S., Berger T. Formal semantics of the Kconfig language. Technical note, University of Waterloo, 2010.

[15]. Veer B., Dallaway J. The eCos component writer's guide, 2000.

[16]. eCos home page. http://ecos.sourceware.org.

[17]. Berger T., Rublack R., Nair D., Atlee J.M., Becker M., Czarnecki K., Wąsowski A. A survey of variability modeling in industrial practice. Proc. of the 7-th Intl. Workshop on Variability Modelling of Software-intensive Systems (VaMoS'2013), article No. 7, ACM 2013. DOI: 10.1145/2430502.2430513.

[18]. Batory D. Feature models, grammars, and propositional formulas. Proc. of the 9-th Intl. Conf. on Software Product Lines (SPLC'05), LNCS 3714, pp. 7-20, 2005. DOI: 10.1007/11554844_3.

[19]. Wang H., Li Y., Sun J., Zhang H., Pan J. A semantic web approach to feature modeling and verification. Proc. of Workshop on Semantic Web Enabled Software Engineering (SWESE'05), p. 44, 2005.

[20]. OWL DL List of reasoners. http://owl.cs.manchester.ac.uk/tools/list-of-reasoners/

[21]. Haarslev V., Hidde K., Möller R., Wessel M. The RacerPro knowledge representation and reasoning system. Semantic Web, 3(3):267-277, 2012.

[22]. Benavides D., Segura S., Trinidad P., Ruiz-Cortés A. Using Java CSP solvers in the automated analyses of feature models. Generative and Transformational Techniques in Software Engineering, LNCS 4143:399-408. Springer, 2006. DOI: 10.1007/11877028_16.

[23]. White J., Doughtery B., Schmidt D., Benavides D. Automated reasoning for multi-step software product-line configuration problems. Proc. of the 13-th Sofware Product Line Conference, pp. 11-20, 2009.

[24]. Zhang W., Mei H., Zhao H. Feature-driven requirement dependency analysis and high-level software design. Requirements Engineering, 11(3):205-220, 2006. DOI: 10.1007/s00766-006-0033-x.

[25]. Hemakumar A. Finding Contradictions in Feature Models. Proc. of 12-th Intl. Conf. on Software Product Lines (SPLC'2008), v. 2, pp. 183-190, 2008.

[26]. Thüm T., Apel S., Kästner C., Kuhlemann M., Schaefer I., Saake G. A classification and survey of analysis strategies for software product lines. ACM Computing Surveys, 47(1), article No. 6, 2014. DOI: 10.1145/2580950.

[27]. Liebig J., von Rhein A., Kästner C., Apel S., Dörre J., Lengauer C. Scalable analysis of variable software. Proceedings of the 2013 9-th Joint Meeting on Foundations of Software Engineering, pp. 81-91. ACM, 2013. DOI: 10.1145/2491411.2491437.

[28]. Meinicke J., Thüm T., Schröter R., Benduhn F., Saake G. An overview on analysis tools for software product lines. Proc. of the 18-th Intl. Software Product Line Conf.: Companion Vol. for Workshops, Demonstrations and Tools – Vol. 2, pp. 94-101. ACM, 2014. DOI: 10.1145/2647908.2655972.

[29]. Dietrich C., Tartler R., Schröder-Preikshat W., Lohmann D. Understanding Linux feature distribution. Proceedings of the 2012 Workshop on Modularity in Systems Software, pp. 15-20. ACM, 2012. DOI: 10.1145/2162024.2162030.

[30]. Melo J., Flesborg E., Brabrand C., Wąsowski A. A quantitative analysis of variability warnings in Linux. Proceedings of the 10-th International Workshop on Variability Modelling of Software-intensive Systems, pp. 3-8. ACM, 2016. DOI: 10.1145/2866614.2866615.

[31]. Sayyad A. S., Ingram J., Menzies T., Ammar H. Scalable product line configuration: a straw to break the camel's back. Proc. of IEEE/ACM 28-th International Conference on Automated Software Engineering (ASE 2013), pp. 465-474. IEEE, 2013. DOI: 10.1109/ASE.2013.6693104.

[32]. Tartler R., Lohmann D., Dietrich C., Egger C., Sincero J. Configuration coverage in the analysis of large-scale system software. SIGOPS Oper. Syst. Rev., 45(3):10-14, 2012. DOI: 10.1145/2039239.2039242.

[33]. Tartler R., Dietrich C., Sincero J., Schröder-Preikschat W., Lohmann D. Static analysis of variability in system software: the 90000 #ifdefs Issue. Proc. of USENIX Annual Technical Conference (USENIX ATC 14), pp. 421-432, 2014.

[34]. Sloane N.J.A. Covering arrays and intersecting codes. Journal of combinatorial designs, 1(1):51-63, 1993. DOI: 10.1002/jcd.3180010106.
[35]. Hartman A., Raskin L. Problems and algorithms for covering arrays. Discrete Mathematics, 284(1):149-156, 2004. DOI: 10.1016/j.disc.2003.11.029.
[36]. Кулямин В.В., Петухов А.А. Обзор методов построения покрывающих наборов. Программирование 37(3):3-41, 2011. DOI: 10.1134/S0361768811030029.
[37]. Cohen M.B., Gibbons P.B., Mugridge W.B., Colbourn C.J. Constructing test suites for interaction testing. Proc. of 25-th Intl. Conf. on Software Engineering, pp. 38-48. IEEE, 2003. DOI: 10.1109/ICSE.2003.1201186.
[38]. Grindal M., Offutt A.J., Andler S.F. Combination testing strategies: a survey. Software Testing, Verification, and Reliability, 15(3):167-199, 2005. DOI: 10.1002/stvr.319.
[39]. Perrouin G., Oster S., Sen S., Klein J., Baudry B., Le Traon Y. Pairwise testing for software product lines: comparison of two approaches. Software Quality Journal, 20(3-4):605-643, 2012. DOI: 10.1007/s11219-011-9160-9.
[40]. Johansen M.F., Haugen Ø, Fleurey F. Properties of realistic feature models make combinatorial testing of product lines feasible. Proc. of Intl. Conf. on Model Driven Engineering Languages and Systems, pp. 638-652. Springer, 2011. DOI: 10.1007/978-3-642-24485-8_47.
[41]. Кулямин В.В. Комбинаторная генерация программных конфигураций ОС. Труды ИСП РАН, 23:359-370, 2012. DOI: 10.15514/ISPRAS-2012-23-20.
[42]. Brabrand C., Ribeiro M., Tolêdo T., Borba P. Intraprocedural dataflow analysis for software product lines. Proc. Intl. Conf. on Aspect-Oriented Software Development (AOSD), pp. 13-24. ACM, 2012. DOI: 10.1007/978-3-642-36964-3_3.
[43]. Thaker S., Batory D., Kitchin D., Cook W. Safe composition of product lines. Proc. of Intl. Conf. on Generative Programming and Component Engineering (GPCE), pp. 95-104. ACM, 2007. DOI: 10.1145/1289971.1289989.
[44]. Heidenreich F. Towards systematic ensuring well-formedness of software product lines. Proc. of Intl. Workshop on Feature-Oriented Software Development (FOSD), pp. 69-74. ACM, 2009. DOI: 10.1145/1629716.1629730.
[45]. Apel S., Kästner C., Größlinger A., Lengauer C. Type safety for feature-oriented product lines. Automated Software Engineering, 17(3):251-300, 2010. DOI: 10.1007/s10515-010-0066-8.
[46]. Kästner C., Apel S., Thüm T., Saake G. Type checking annotation-based product lines. ACM Trans. Software Engineering and Methodology, 21(3):1-39, 2012. DOI: 10.1145/2211616.2211617.
[47]. Gruler A., Leucker M., Scheidemann K. Modeling and model checking software product lines. Proc. of IFIP Intl. Conf. on Formal Methods for Open Object-based Distributed Systems (FMOODS), pp. 113-131. Springer, 2008. DOI: 10.1007/978-3-540-68863-1_8.
[48]. Lauenroth K., Toehning S., Pohl K.. Model checking of domain artifacts in product line engineering. Proc. Intl. Conf. on Automated Software Engineering (ASE), pp. 269-280. IEEE, 2009. DOI: 10.1109/ASE.2009.16.
[49]. Classen A., Heymans P., Schobbens P.-Y., Legay A., Raskin J.-F. Model checking lots of systems: efficient verification of temporal properties in software product lines. Proc. Int. Conf. Software Engineering (ICSE), pp. 335-344. ACM, 2010. DOI: 10.1145/1806799.1806850.
[50]. Apel S., Speidel H., Wendler P., von Rhein A., Beyer D. Detection of feature interactions using feature-aware verification. Proc. of Intl. Conf. on Automated Software Engineering (ASE), pp. 372-375. IEEE, 2011. DOI: 10.1109/ASE.2011.6100075.

[51]. Cordy M., Heymans P., Legay A., Schobbens P.-Y., Dawagne B., Leucker M. Counterexample guided abstraction refinement of product-line behavioural models. Proc. of 22-nd ACM SIGSOFT Intl. Symposium on Foundations of Software Engineering (FSE 2014), pp. 190-201. ACM, 2014. DOI: 10.1145/2635868.2635919.

[52]. Thüm T., Schaefer I., Apel S., Hentschel M. Family-based deductive verification of software product lines. Proc. of the 11-th Intl. Conf. on Generative Programming and Component Engineering (GPCE '12), pp. 11-20. ACM, 2012. DOI: 10.1145/2371401.2371404.

[53]. Kästner C., von Rhein A., Erdweg S., Pusch J., Apel S., Rendel T., Ostermann K. Toward variability-aware testing. Proc. of Intl. Workshop on Feature-Oriented Software Development (FOSD), pp. 1-8. ACM, 2012. DOI: 10.1145/2377816.2377817.

Verification and analysis of variable operating systems

[1,2,3] *V.V. Kuliamin <kuliamin@ispras.ru>*
[1,4] *E.M. Lavrischeva <lavr@ispras.ru>*
[1] *V.S. Mutilin <mutilin@ispras.ru>*
[1,2,3] *A.K. Petrenko <petrenko@ispras.ru>*

[1] *Institute for System Programming RAS,*
A. Solzhenitsyn str., 25, Moscow, 109004, Russia
[2] *Lomonosov Moscow State University,*
Leninskie gory, 1, Moscow, 119991, Russia
[3] *FCS NRU Higher School of Economics,*
Myasnitskaya str., 20, Moscow, 101000, Russia
[4] *Moscow Institute of Physics and Technology,*
Institutskiy per., 9, Dolgoprudny, Moscow reg., 141700, Russia

Abstract. This paper regards problems of analysis and verification of complex modern operating systems, which should take into account variability and configurability of those systems. The main problems of current interest are related with conditional compilation as variability mechanism widely used in system software domain. It makes impossible fruitful analysis of separate pieces of code combined into system variants, because most of these pieces of code has no interface and behavior. From the other side, analysis of all separate variants is also impossible due to their enormous number. The paper provides an overview of analysis methods that are able to cope with the stated problems, distinguishing two classes of such approaches: analysis of variants sampling based on some variants coverage criteria and variation-aware analysis processing many variants simultaneously and using similarities between them to minimize resources required. For future development we choose the most scalable technics, sampling analysis based on code coverage and on coverage of feature combinations and variation-aware analysis using counterexample guided abstraction refinement approach.

Keywords: operating system; software product family; variability model; software verification; static analysis; model checking; type safety checking; source code coverage; covering array; counterexample-guided abstraction refinement.

DOI: 10.15514/ISPRAS-2016-28(3)-12

For citation: Kuliamin V.V., Lavrischeva E.M., Mutilin V.S., Petrenko A.K. [Verification and analysis of variable operating systems]. *Trudy ISP RAN/Proc. ISP RAS*, vol. 28, issue 3, 2016, pp. 189-208 (in Russian). DOI: 10.15514/ISPRAS-2016-1(2)-12

References

[1]. Jacobson I., Griss M., Jonsson P. Software Reuse, Architecture, Process and Organization for Business Success. Addison-Wesley, 1997. ISBN-13: 978-0201924763.
[2]. Bosch J. Design and Use of Software Architectures: Adopting and Evolving a Product Line Approach. Pearson Education, 2000. ISBN-13: 978-0201674941.
[3]. Clements P., Northrop L. Software Product Lines: Practices and Patterns. SEI Series in Software Engineering, Addison-Wesley, 2001. ISBN-13: 978-0201703320.
[4]. Pohl K., Böckle G., van der Linden F. J. Software Product Line Engineering: Foundations, Principles and Techniques. Springer-Verlag, 2005. DOI: 10.1007/3-540-28901-1.
[5]. Bachmann F., Clements P. Variability in software product lines. CMU/SEI Technical Report CMU/SEI-2005-TR-012, 2005.
[6]. Lotufo R., She S., Berger T., Czarnecki K., Wąsowski A. Evolution of the Linux kernel variability model. Proc. of SPLC'10, LNCS 6287:136-150, Springer, 2010. DOI: 10.1007/978-3-642-15579-6_10.
[7]. Lavrischeva E.M., Koval' G.I., Slabospitskaya O.O., Kolesnik A.L. [Product Line Development Management Specifics]. Problemy programmiovaniya [Problems of Software Developement], (3):40-49, 2009 (in Ukrainian).
[8]. Lavrischeva E.M., Slabospitskaya O.O., Koval' G.I., Kolesnik A.L. [Theoretical Aspects of Variability Management in Product Lines]. Vesnik KNU seria fiz.-mat. nauk [Notes of KNU, series on maths and physics], (1):151-158, 2011 (in Ukrainian).
[9]. Kang K., Cohen S., Hess J., Novak W., Peterson S. Feature-oriented domain analysis (FODA) feasibility study. CMU/SEI Technical Report CMU/SEI-90-TR-21, 1990.
[10]. Benavides D., Segura S., Ruiz-Cortés A. Automated analysis of feature models 20 years later: a literature review. Information Systems, 35(6):615–636, 2010. DOI: 10.1016/j.is.2010.01.001.
[11]. Chen L., Babar M.A. A systematic review of evaluation of variability management approaches in software product lines. Information and Software Technology, 53(4):344–362, 2011. DOI: 10.1016/j.infsof.2010.12.006.
[12]. Berger T., She S., Lotufo R., Wąsowski A., Czarnecki K. A study of variability models and languages in the systems software domain. IEEE Transactions on Software Engineering, 39(12):1611-1640, 2013. DOI: 10.1109/TSE.2013.34.
[13]. Zippel R. et al. Kconfig language. https://www.kernel.org/doc/Documentation/kbuild/kconfig-language.txt.
[14]. She S., Berger T. Formal semantics of the Kconfig language. Technical note, University of Waterloo, 2010.
[15]. Veer B., Dallaway J. The eCos component writer's guide, 2000.
[16]. eCos home page. http://ecos.sourceware.org.
[17]. Berger T., Rublack R., Nair D., Atlee J.M., Becker M., Czarnecki K., Wąsowski A. A survey of variability modeling in industrial practice. Proc. of the 7-th Intl. Workshop on

Variability Modelling of Software-intensive Systems (VaMoS'2013), article No. 7, ACM 2013. DOI: 10.1145/2430502.2430513.

[18]. Batory D. Feature models, grammars, and propositional formulas. Proc. of the 9-th Intl. Conf. on Software Product Lines (SPLC'05), LNCS 3714, pp. 7-20, 2005. DOI: 10.1007/11554844_3.

[19]. Wang H., Li Y., Sun J., Zhang H., Pan J. A semantic web approach to feature modeling and verification. Proc. of Workshop on Semantic Web Enabled Software Engineering (SWESE'05), p. 44, 2005.

[20]. OWL DL List of reasoners. http://owl.cs.manchester.ac.uk/tools/list-of-reasoners/

[21]. Haarslev V., Hidde K., Möller R., Wessel M. The RacerPro knowledge representation and reasoning system. Semantic Web, 3(3):267-277, 2012.

[22]. Benavides D., Segura S., Trinidad P., Ruiz-Cortés A. Using Java CSP solvers in the automated analyses of feature models. Generative and Transformational Techniques in Software Engineering, LNCS 4143:399-408. Springer, 2006. DOI: 10.1007/11877028_16.

[23]. White J., Doughtery B., Schmidt D., Benavides D. Automated reasoning for multi-step software product-line configuration problems. Proc. of the 13-th Sofware Product Line Conference, pp. 11-20, 2009.

[24]. Zhang W., Mei H., Zhao H. Feature-driven requirement dependency analysis and high-level software design. Requirements Engineering, 11(3):205-220, 2006. DOI: 10.1007/s00766-006-0033-x.

[25]. Hemakumar A. Finding Contradictions in Feature Models. Proc. of 12-th Intl. Conf. on Software Product Lines (SPLC'2008), v. 2, pp. 183-190, 2008.

[26]. Thüm T., Apel S., Kästner C., Kuhlemann M., Schaefer I., Saake G. A classification and survey of analysis strategies for software product lines. ACM Computing Surveys, 47(1):article 6, 2014. DOI: 10.1145/2580950.

[27]. Liebig J., von Rhein A., Kästner C., Apel S., Dörre J., Lengauer C. Scalable analysis of variable software. Proceedings of the 2013 9-th Joint Meeting on Foundations of Software Engineering, pp. 81-91. ACM, 2013. DOI: 10.1145/2491411.2491437.

[28]. Meinicke J., Thüm T., Schröter R., Benduhn F., Saake G. An overview on analysis tools for software product lines. Proc. of the 18-th Intl. Software Product Line Conf.: Companion Vol. for Workshops, Demonstrations and Tools – Vol. 2, pp. 94-101. ACM, 2014. DOI: 10.1145/2647908.2655972.

[29]. Dietrich C., Tartler R., Schröder-Preikshat W., Lohmann D. Understanding Linux feature distribution. Proceedings of the 2012 Workshop on Modularity in Systems Software, pp. 15-20. ACM, 2012. DOI: 10.1145/2162024.2162030.

[30]. Melo J., Flesborg E., Brabrand C., Wąsowski A. A quantitative analysis of variability warnings in Linux. Proceedings of the 10-th International Workshop on Variability Modelling of Software-intensive Systems, pp. 3-8. ACM, 2016. DOI: 10.1145/2866614.2866615.

[31]. Sayyad A. S., Ingram J., Menzies T., Ammar H. Scalable product line configuration: a straw to break the camel's back. Proc. of IEEE/ACM 28-th International Conference on Automated Software Engineering (ASE 2013), pp. 465-474. IEEE, 2013. DOI: 10.1109/ASE.2013.6693104.

[32]. Tartler R., Lohmann D., Dietrich C., Egger C., Sincero J. Configuration coverage in the analysis of large-scale system software. SIGOPS Oper. Syst. Rev., 45(3):10-14, 2012. DOI: 10.1145/2039239.2039242.

[33]. Tartler R., Dietrich C., Sincero J., Schröder-Preikschat W., Lohmann D. Static analysis of variability in system software: the 90000 #ifdefs Issue. Proc. of USENIX Annual Technical Conference (USENIX ATC 14), pp. 421-432, 2014.

[34]. Sloane N.J.A. Covering arrays and intersecting codes. Journal of combinatorial designs, 1(1):51-63, 1993. DOI: 10.1002/jcd.3180010106.

[35]. Hartman A., Raskin L. Problems and algorithms for covering arrays. Discrete Mathematics, 284(1):149-156, 2004. DOI: 10.1016/j.disc.2003.11.029.

[36]. Kuliamin V.V., Petukhov A.A. A survey of methods for constructing covering arrays. Programming and Computer Software, 37(3):121-146, 2011. DOI: 10.1134/S0361768811030029.

[37]. Cohen M.B., Gibbons P.B., Mugridge W.B., Colbourn C.J. Constructing test suites for interaction testing. Proc. of 25-th Intl. Conf. on Software Engineering, pp. 38-48. IEEE, 2003. DOI: 10.1109/ICSE.2003.1201186.

[38]. Grindal M., Offutt A.J., Andler S.F. Combination testing strategies: a survey. Software Testing, Verification, and Reliability, 15(3):167-199, 2005. DOI: 10.1002/stvr.319.

[39]. Perrouin G., Oster S., Sen S., Klein J., Baudry B., Le Traon Y. Pairwise testing for software product lines: comparison of two approaches. Software Quality Journal, 20(3-4):605-643, 2012. DOI: 10.1007/s11219-011-9160-9.

[40]. Johansen M.F., Haugen Ø, Fleurey F. Properties of realistic feature models make combinatorial testing of product lines feasible. Proc. of Intl. Conf. on Model Driven Engineering Languages and Systems, pp. 638-652. Springer, 2011. DOI: 10.1007/978-3-642-24485-8_47.

[41]. Kuliamin V.V. [Combinatoric generation of operating system software configurations]. Trudy ISP RAN/Proc. ISP RAS, 23:359-370, 2012 (in Russian). DOI: 10.15514/ISPRAS-2012-23-20.

[42]. Brabrand C., Ribeiro M., Tolêdo T., Borba P. Intraprocedural dataflow analysis for software product lines. Proc. Intl. Conf. on Aspect-Oriented Software Development (AOSD), pp. 13-24. ACM, 2012. DOI: 10.1007/978-3-642-36964-3_3.

[43]. Thaker S., Batory D., Kitchin D., Cook W. Safe composition of product lines. Proc. of Intl. Conf. on Generative Programming and Component Engineering (GPCE), pp. 95-104. ACM, 2007. DOI: 10.1145/1289971.1289989.

[44]. Heidenreich F. Towards systematic ensuring well-formedness of software product lines. Proc. of Intl. Workshop on Feature-Oriented Software Development (FOSD), pp. 69-74. ACM, 2009. DOI: 10.1145/1629716.1629730.

[45]. Apel S., Kästner C., Größlinger A., Lengauer C. Type safety for feature-oriented product lines. Automated Software Engineering, 17(3):251-300, 2010. DOI: 10.1007/s10515-010-0066-8.

[46]. Kästner C., Apel S., Thüm T., Saake G. Type checking annotation-based product lines. ACM Trans. Software Engineering and Methodology, 21(3):1-39, 2012. DOI: 10.1145/2211616.2211617.

[47]. Gruler A., Leucker M., Scheidemann K. Modeling and model checking software product lines. Proc. of IFIP Intl. Conf. on Formal Methods for Open Object-based Distributed Systems (FMOODS), pp. 113-131. Springer, 2008. DOI: 10.1007/978-3-540-68863-1_8.

[48]. Lauenroth K., Toehning S., Pohl K.. Model checking of domain artifacts in product line engineering. Proc. Intl. Conf. on Automated Software Engineering (ASE), pp. 269-280. IEEE, 2009. DOI: 10.1109/ASE.2009.16.

[49]. Classen A., Heymans P., Schobbens P.-Y., Legay A., Raskin J.-F. Model checking lots of systems: efficient verification of temporal properties in software product lines. Proc.

Int. Conf. Software Engineering (ICSE), pp. 335-344. ACM, 2010. DOI: 10.1145/1806799.1806850.

[50]. Apel S., Speidel H., Wendler P., von Rhein A., Beyer D. Detection of feature interactions using feature-aware verification. Proc. of Intl. Conf. on Automated Software Engineering (ASE), pp. 372-375. IEEE, 2011. DOI: 10.1109/ASE.2011.6100075.

[51]. Cordy M., Heymans P., Legay A., Schobbens P.-Y., Dawagne B., Leucker M. Counterexample guided abstraction refinement of product-line behavioural models. Proc. of 22-nd ACM SIGSOFT Intl. Symposium on Foundations of Software Engineering (FSE 2014), pp. 190-201. ACM, 2014. DOI: 10.1145/2635868.2635919.

[52]. Thüm T., Schaefer I., Apel S., Hentschel M. Family-based deductive verification of software product lines. Proc. of the 11-th Intl. Conf. on Generative Programming and Component Engineering (GPCE '12), pp. 11-20. ACM, 2012. DOI: 10.1145/2371401.2371404.

[53]. Kästner C., von Rhein A., Erdweg S., Pusch J., Apel S., Rendel T., Ostermann K. Toward variability-aware testing. Proc. of Intl. Workshop on Feature-Oriented Software Development (FOSD), pp. 1-8. ACM, 2012. DOI: 10.1145/2377816.2377817.

Enabling Data Driven Projects for a Modern Enterprise

Artyom Topchyan <a.topchyan@reply.de>
Yerevan State University,
0025 Alek Manukyan 1, Yerevan, Armenia.

Abstract. With the growing volume and demand for data a major concern for an Organization trying to implement Data Driven projects, is not only how to technically collect, cleanse, integrate, access, but even more so, how and why to use it. There is a lack of unification on a logical and technical level between Data Scientists, IT departments and Business departments, as it is very unclear where the data comes from, what it looks like, what it contains and how to process it in the context of existing systems. So in this paper we present a platform for data exploration and processing, which enables Data-Driven projects, that does not require a complete organizational revamp, but provides a workflow and technical basis for such projects.

Keywords: data-driven projects, crisp, Hadoop, data vault, distributed, information retrieval, sandbox, topic modelling, streaming processing, auto-scaling, mesos, kafka

DOI: 10.15514/ISPRAS-2016-28(3)-13

For citation: Topchyan A.R. Enabling Data Driven Projects for a Modern Enterprise. Trudy ISP RAN/Proc. ISP RAS, vol. 28, issue 3, 2016, pp. 209-230. DOI: 10.15514/ISPRAS-2016-28(3)-13

1. Introduction

More and more Organizations are aiming at implementing Data-Driven projects [1][2] which aim at increasing the quality, speed and/or quantity of information gained from Data collected by the Enterprise. The main goal is increasing the quality, speed, and/or quantity of information gain for the purpose of innovation (e.g. innovating a new methodology) or the economic benefit to an organization.

This is particularly challenging for existing Enterprises with years of organizational structure and system already in place, as completely changing the way data is accumulated, handled, shared and used is not feasible. To this end in this work we present a platform for data exploration and processing, which simplifies data driven project by means of intelligent automation. The main goal of the platform is to improve any project that relies on data analysis, but at the same time can coexist with the existing landscape and not require an immediate organizational revamp.

The solution is designed to address three real world view points of issues in a Data-Driven projects flow. We accumulated these viewpoints when working with such projects at large Organizations. These viewpoints are outlined below and represent the challenged a specific parts of the organizations usually faces in the development stages of a Data-Driven project. We outline the issue and shortly outline how we solve them by means of analysis, automation and logical structure.

1.1 Issues from the point of view of the IT department

The issues an IT department has, are often of a technical nature. One major problem is the inability to give users access to raw data. Usually data can only be shared by ways of export tools, which are vendor and apply some transformations to clean up the data, this is often very useful, but in some cases, which we will outline below lead to data loss and or corruption. Another problem commonly faced is the lack of processing resources to use for any type of exploratory large scale processing. Again most system and proprietary and solve a very specific use case the department has. This also extends the first 3 issues in the sense, that there is basically no unified, system independent way to provide data in a consistent format to users. And third there is the issue of unstructured data, such as documents, log files and others. These are most often not stored in a central system,. such as a database, but are scattered around the departments and are handled in very specific ways. This data is nonetheless immensely valuable when combined with user data, and the systems which can load and interpret this data, such as monitoring and operational systems are not designed to provide facilities for data export and analysis outside their specific context. We aim to solve this problem by implementing a so called Enterprise Data Vault, which provides flexibility to ingest any type of data, while preserving its structural and logical relationship. This approach also imposes and standardization on data extraction semantics as well as format. The main goal is so that all data is extracted in a consistent way in the scope of the same framework. A central goal is to also support near-real-time ingestion for source that can benefit from this and to facilitate dynamic and evolving schemas as well as a multitude of formats.

1.2 Issues from the point of view of Data Scientists

The main issues Data Scientist encounters at most Data-Driven projects are Organizational or Information sharing related. It is very often unclear who owns the data, how much there is, what it contains and where it is stored. It is practically quite challenging to keep the data with the same structure ad without introducing a lot of structural changes as systems evolve and change with time. This is a usual issue with change management in very large Organizations and is very complex to solve directly. Most of this information is contained or can be inferred from project and data documentation as

well. This information is readily available, but scattered around dozens of systems and departments with no simple way to search or analyse it from a unified interface. Even if the data documentation is found and stakeholders are contacted there is no guarantee the data is usable as it might contain dropped fields, wrong data types and so forth, is essence no data profile is available. This is often a data quality and governance issue, but this on its own can be challenging with a very large volume of data in some systems and is often ignored or not update as projects continue and technical staff comes and goes. We aim to solve this problem by building a large scale and intelligent index of all project documentation and and information about data and people responsible for it. This model should capture changes to data, project and technical personnel without being influenced or depending on manual updates and necessity for bookkeeping. We extract the mapping of data source to project based on documentation content, the mapping between data, projects and key knowledge owners based on the data and document metadata as well as documentation content analysis. These are then connected to each other based on the data in question and augmented with comprehensive data profiles, which offer data consistency and distribution at a glance. This will allow Data Scientist to understand a lot about the data even before getting access to it. Once a decision has been made, that the data is useful a so called isolated Sandbox environment will be provisioned, which has access to the data and a cluster of computing power. The Data Scientist will have full control in his isolated environment with tools in place to fetch program dependencies, collaboration and visioning. The environments are isolated, but are collocated on the same hardware with dynamic resource allocation and monitoring, which allows a high degree of efficiency in such a highly multi-tenant environment.

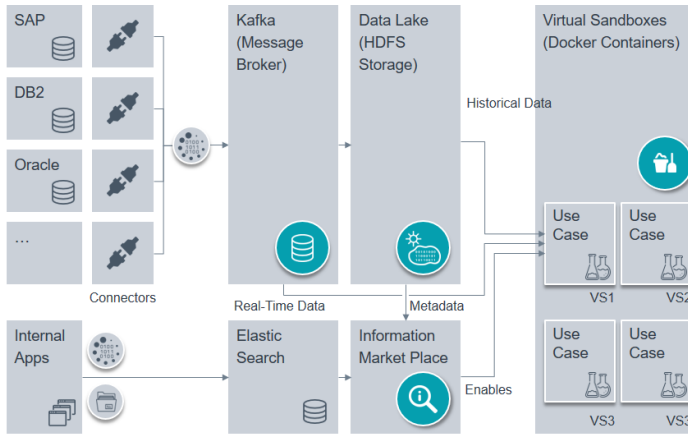


Fig. 1. Operational Data Platform

1.3 Issues from the point of view of the Business department

The business department being the most knowledgeable of business processes and what the data actually means in a business context is most often very much interested in a simple way of querying and exploring data by means of structure queries or analytical views. This is interesting for them in order to understand what effect their business decisions potentially informed by Data Scientist and Analysts, actually have. This is in most cases quite similar to what the Data Scientist and Analysts want as well. The key difference is, that the Business department is most interested in how effective all the DataDriven projects are, be it the value they bring or in case of failed projects, how much resources were wasted. We address this by the combination of the entire platform of the Enterprise Data Vault and Sandbox environments greatly increasing productivity and minimizing waiting time to access and process data, and the Information Marketplace greatly decreasing the time required to analyse and understand if a use case is viable considering the information present.

We define our proposed solution as the Operational Data Platform. The solution is based on modern concepts of resource scheduling [3][6] and immutability concepts to achieve flexibility and scalability. The entire platform is built around the concept of data streams [4]. The platform has been successfully deployed and is being used by a Large Automotive Organization in Germany to investigate Data-Driven project based on large variety of data, such as analysis time series Car Telematics Data to predict faults or patterns, analysing textual Quality Assurance data and others. A high level overview of the entire solution is presented in Fig.1 and contains all the building blocks and their connections. We will go into more detail about each individual component in the following the chapters. But first we will define in more detail, what we understand as a Data-Driven project at an Enterprise.

2. Data Driven Projects

To clarify the problem we are approaching lets define in more detail what a Data-Driven Project is and what the life cycle and goals are. A Data-Driven project aims at increasing the quality, speed and/or quantity of information gained from Data. Any type of data can be used varying in size, source and business/operational importance. Such projects usually involve Data Scientists, Business and IT working together to build up use cases by analysing, processing and presenting data in a meaningful way. The result of the project may be a report, dashboard, or a web service used by other systems. These are very involved projects and require a great degree of domain, statistical, modelling as well as large scale data processing knowledge. To highlight the problem we are solving, lets take a typical datadriven project lifecycle at a major enterprise. Most Data-Driven project

follow a variation of the Cross Industry Standard Process for Data Mining project lifecycle [8]. This varies from organization to organization depending on the maturity of the Data-Driven mindset, but CRISP is one of the most widely accepted approaches to such projects. The life-cycle usually consist of six stages of development, which can be iterated upon and repeated or completely abandoned. A slightly modified version of CRISP is:

1 Business Understanding

This initial phase focuses on understanding the project objectives and requirements from a business perspective, and then converting this knowledge into a data-driven problem definition, and a preliminary plan designed to achieve the objectives [8]. This can vary from common cases such as Fraud Analysis to large Scale Car Telemetry analysis. Outlining the case often takes on month or more.

2 Data Understanding

Once the cases is more or less clear it has to be clarified if there is any data to support it. The business department often knows what data can be used, but more substantive knowledge of the data is required, so it is a task of the business department and the Data Scientists to find out who owns and has knowledge of any data related to the case. This can take a very long time and is notoriously difficult in a large organization, because it is most often unclear who the data owner is and who is knowledgeable about this data. These are quite often different people. In our experience, this process might take upwards to two months' time and often it is discovered there is not substantial data to support the use case. This is already approximately 3 months on a case that potentially is not even possible.

3 Data acquisition and preparation

If the data is present the next challenge is to acquire even a small sample of the data, which is usually customer data and is not shared easily between departments. This is again a costly process and can take up to two months. Luckily Data Scientist can start work on at least sample data if it can be supplied. But this again does not guarantee any data will arrive in the end. The data has to be transferred and transformed into a usable state. This may also take a large amount of time and is quite often the most time consuming phase that involves technical work. In our experience this process is repeated multiple times throughout the project and each iteration may take weeks. At this stage it can be found out that the data is corrupted, with columns missing or being uninterpretable due to formatting loss or it is just very sparse.

4 Modeling

In this phase, various modeling techniques are selected and applied, and their parameters are calibrated to optimal values. This is dependant on what type of problem is being solved and is greatly influence by the type of data

available. Some techniques have specific requirements on the form of data. Therefore, stepping back to the data preparation phase is often needed [8]. Depending on the problem and budget this can take anywhere from one to two months.

5 Evaluation

At this stage some result can be shown and the models and approach evaluated, preliminary results discussed and it is decided if there any value in continuing the project.

6 Deployment

Creation of the model is generally not the end of the project. Even if the purpose of the model is to increase knowledge of the data, the knowledge gained will need to be organized and presented in a way that it is usable. Depending on the requirements, the deployment phase can be as simple as generating a report or as complex as establishing a fault-tolerant and monitorable process to repeat the modelling and provide periodic or real time results to the user. This is usually done in conjunction with the IT department. This process has to often be subject to the requirements and limitation of the departments hosting the solution, which greatly limits flexibility and performance. In our experience this is actually the most complicated step based on the problem and can take many months.

It should be noted, that quite often these project involved external suppliers, which means they are inherently more expensive the longer the projects take. So in essence a project can fail on at least 3 separate stages. Which might take months and can be very expensive and deliver few to no results. With our proposed solution we try to tackle this time and knowledge requirement and achieve faster success and faster failure time windows. In the solution we presenting, Data Scientist will have access to all of the project documentation and with the addition of intelligent search capabilities, they can quickly find what data is about or who to ask about it. In our experience this greatly facilitates the Business Understating. Using this information, they can easily transition to the Data Understanding phase and analyses the data source and discover the data-profile, what the data actually contains and if it is actually useful and contains the required information. If the data is there the only thing stopping the Data Scientists is the knowledge about the form of the data and authorization to access it. This is again streamlined as all the data is stored in a central repository in a very strict logical hierarchy and the data schema and access rights are presented with the data itself. This will greatly simplify some of the usual administrative and mechanical tasks a Data Scientist would have to go through in the Data preparation stage The last step is to get the authorization to use the data and request and analytical environment to process it. In contrast to how this is usually handled, we try to automate the process as much as possible. The only thing required is what

data is needed, for how much time and what kinda of processing power and tools are required. The data owner if automatically notified and they can specify which parts of the data can be used and this permission is granted based on a fine grained access control scheme. The requested environment is the automatically provisioned with all the analytical and collaboration tools built in. This aids the Data Scientist in the Modelling and evaluations steps as they have access to a fit for purpose environments, where many Data Scientists can collaborate and iterate on their findings. Once the case is ready our platform greatly simplifies the deployment process as the data, resource and tool requirements are fairly transparent at this stage. To this end lets go into more detail about the components of the solution that allow this flexibility, starting with how the data is collected, processed and stored.

3. Enterprise Data Reservoir

In order to enable all truly dynamic and Data-Driven projects, Analysts and Data Scientists need to have unimpeded access to basically all use case and customer related data. A logical first step is to aggregate all the data of the Enterprise in a central place, so that one central source of truth is viable to Data Scientists and the Business department. This is what has been traditionally done in the Data Warehousing world. Data Warehouses are central repositories of integrated data from one or more disparate sources. They store current and historical data and are used for creating analytical reports for knowledge workers throughout the enterprise. Examples of reports could range from annual and quarterly comparisons and trends to detailed daily sales analysis[16]. The problem with this approach is, that it is very structured by definition. Relationships are all predefined and data is usually transformed between the extraction and load steps to answer very specific questions. This is keys for performance and to reflect enterprise specific requirements for these views. For most Data Science use cases this is not optimal, because structured transformation tend to remove some useful data as they reflect the needs of the application, which may or may not align with the goals of the data scientist.

Let’s take a project that aims to analyze text descriptions of defects in different car models of an automotive manufacturer. These text descriptions also contain information about, which car models the issues are about. Now let’s say the business department is interested in using this data to build a report of faults by car to analyze the efficiency of the Quality Assurance department, but some older models that are not produced anymore or have their names changed since the inception of the system. So the view in the data-warehouse should contain the actual names and only the specific models the business department is interested in. This works very well and provides the report that the business department is interested in. Now a Data Science project is started to analyze these descriptions and use them to predict

possible future issues in newer models based on the problem description and historic data [10]. The data in the Data Warehouse would be extremely biased towards specific problem for specific cars, it would also essentially not contain some models or contain ambiguity between the model dictionary the data scientist has and what the data contains. would lead to the Clustering and Classifications models not generalizing to the entirety of possible issues and cars. The answer would be to go to the actual data source and use the raw data, in its original form. This is often very complicated or even not possible due to the structure of the Enterprise and the way data ownership is handled and is very costly to implement on project by project basis. The currently accepted solution for this is to load all enterprise raw data into a a single repository, a Data Lake [11]. A |Data Lake is a method of storing data within a system that facilitates the collocation of data in variable schemas and structural forms, usually object blobs or files. Data Lakes are a popular way to store data in a modern enterprise. The usual architecture is fairly similar to a Data Warehouse, with the exception of almost all transformation. The main role of a Data Lake is to serve as a single point of truth, which can be used to create use cases, which join and analyse data from multiple departments. It addresses issues of scalable and affordable storage, while keeping raw data intact by loading the data unchanged into a distributed file system, like the Hadoop File System [12] and provides and batch oriented integration layer for downstream consumers and use cases. This approach has a lot of merits, but most implementation lacks certain key aspects, which are more and more important for a modern business, such as self-describing data, tolerance to changes in the data source and support for low latency data sources. For the purpose of this platform we have adopted a variation of the Data Vault approach coupled with some concept of a Data lake implemented on top of a variation of a Lambda Architecture.

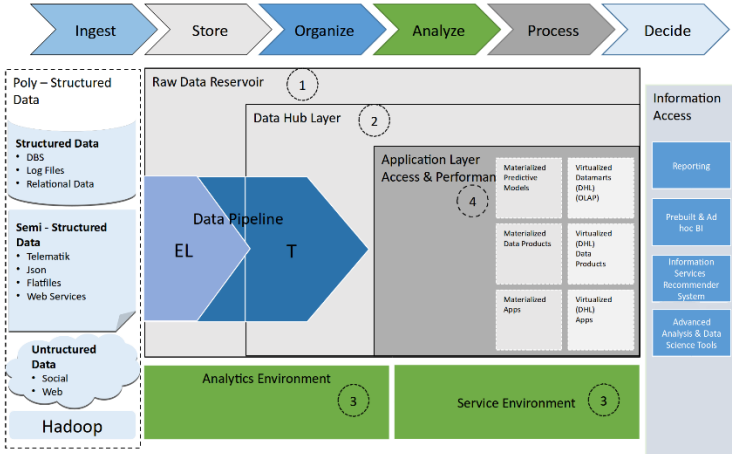
3.1. Data Reservoir

In contrast to the commonly accepted practice of just ingesting all the data as is into separate parts of the filesystem and then transforming it into a meaningful state, our approach to model the data in a more structured as we impose the format, structure and extraction semantics, but we still remain flexible as the data is still ingested in almost raw form and Data Vault modeling is only applied to sources for which it makes sense. Our approach is based on 6 layers:

1. Ingest
2. Store
3. Organize
4. Analyse

5. Process
6. Decide

The overall structure is outlined in 3. It includes all the Layer from Ingestion to Serving(Decision) layer and based on our experience key components are the Lambda Architecture underpinning this and the Organizational Hub layer, modelled as a Data Vault. The Ingest, Store, Process and Analyse are layers, which are mostly based on a variation of a Lambda Architecture and also include the Raw Data Storage layers, which is essentially a Data Lake implementation. This is outlined in Fig, 2.



Fig, 2. Data Vault Architecture

3.1.1 Lambda Architecture

Let present the Lambda Architecture from the point of view appropriate in our context of data processing. The Lambda architecture is a data-processing architecture designed to handle massive quantities of data by taking advantage of both batch- and stream-processing methods. This approach to system architecture, used in our context, attempts to balance latency, throughput, and fault tolerance by using batch processing to provide comprehensive and accurate views of batch data, while simultaneously using real-time stream processing to provide views of online data. The two view outputs can be joined before presentation [14]. In a classical Lambda Architecture [15]:

1. All data entering the system is dispatched to both a batch layer and a speed layer for processing.
2. The batch layer has two functions:
   (a) Managing the master dataset (an immutable, append-only

set of raw data)
(b) Pre-compute the batch views.

3. The serving layer indexes the batch views so that they can be queried in low-latency, ad-hoc way
4. The speed layer compensates for the high latency of updates to the serving layer and deals with recent data only.
5. Any incoming query can be answered by merging results from batch views and real-time views.

This interconnections of the layer is outlined in Fig.3.

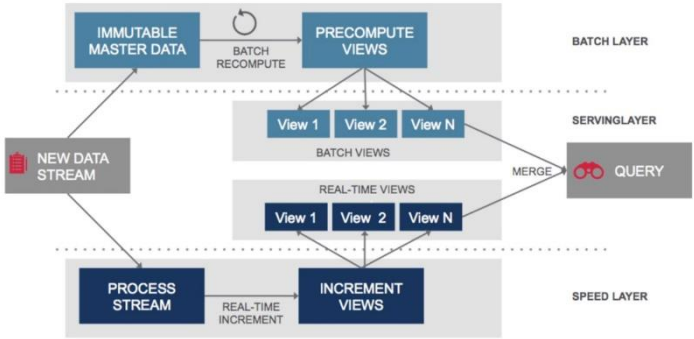


Fig. 3. Lambda Architecture

The main rational for a Lambda Architecture is to efficiently answer a query overthe entire dataset. The challenge is that running arbitrary functions of an unbounded set of data is very expensive. Thus the Lambda Architecture decouples this two processes and offloads only efficient simple queries to the real-time layer as outlined in 4. To achieve this the Lambda Architecture models the ingestion layer as a stream of datum's, which are ingested into a Distributed Message Queue, which allows to both create an Immutable Master Dataset, which is an append only historical log of all events and transactions, and a real-time layer, which answers very specific questions on the incoming stream of data. Having an Immutable set of data not only allows to easily build up analytical use cases, as well as creating reliable point in time snapshots of data, which allows to recreate the full dataset at any point in time. Most commonly in batch view in the Lambda architecture is where the data persistent to a Filesystem takes place, this is a simplification as achieving low latency exactly once persistence is quite challenging. So the persistence layer is essentially part of the processing layer.

We have certain variations from the normal Lambda Architecture. Specifically, the ingestion layer is fully decoupled from the processing layer and also supports near real-time incremental persistence to HDFS as opposed to a scheduled batch job, as

in the standard Lambda Architecture [14]. This allows the ingestion layer to be parallel to the actual views that have to build on the data. We have found this to be most beneficial for large Organizations, where it is sometimes unclear what the default views and queries are before analysing the data. This also allows for the same degree of consistency offered by the Lambda Architecture, but with less latency.

3.1.2 Data Hub

The Data stored in real-time in the data reservoir has to be structured in a meaningful way for actual business application to benefit from them. Views on the stored data are created based on the business demand, existing project, some analysis or processing a Data Scientist carried or a generic view of the raw data. This might be a static periodic view or a real-time streaming view integrated into a serving layer as described above. To keep the modelling consistent, we standardize on the Data Vault approach. Data vault modeling is a database modeling method that is designed to provide a modular way to incorporate multiple operational systems [9]. It is also a method of looking at historical data that, apart from the modeling aspect, deals with issues such as auditing, tracing of data, loading speed and resilience to change, which is critical for the Data acquisition and Processing steps. Data vault contains three different types of technical entities

1. Hubs. Hubs contain a list of unique business keys with low propensity to change.
2. Links. Associations or transactions between business keys (relating for instance the hubs for customer and product with each other through the purchase transaction) are modeled using link tables.
3. Satellite. The hubs and links form the structure of the model, but have no temporal attributes and hold no descriptive attributes. These are stored in separate tables called satellites.

All the principal data sources, such as CRM data, Telematics data, Accounting Data, Product detail, Quality Assurance data are modelled in the Data Vault approach and exposed as views and APIs. The combination of real-time data ingestion, processing and Data Vault modelling leads to a simple and flexible Data Model, which solves the problem we posed in this section in a more robust way. Historic data is easily accessible in real-time, while rigid defined views are still available for more structured use cases as well as enabling a multitude or real-time use cases.

3.1.3 Implementation details

The described data model and Lambda Architecture was implemented on top of the described platform. With all the services running as immutable services in a shared

cluster environment, managed by a central resource manager and service discovery solution.

The central part is the Apache Kafka broker [4], which serves as the main coupling layer for the entire solution. As described in the previous chapters we employ the Staged Even-Driven architecture for most of the transformation, that has to implemented on the platform, but the data collection and storage is modeled fully after the Lambda Architecture.

All data source and sink connectors are running by reading and writing messages to Kafka, most of the job progress tracking, reporting and orchestration is done by using Kafka. This allows for the connectors to run in a distributed fashion with automatic or manual scaling done via Apache Mesos [3]. More connectors are launched based on demand and are fully immutable, which means if a connectors crashed, a new instance will be transparently started somewhere else and it will resume work from the last offset. This is quite critical in order to always provide a consistent and up to date view of the source systems to the Data Scientist using the platform, which is a central part of almost all the CRISP stages. Currently we support a variety of connectors specifically developed for the described platform, all of which are based on the Kafka-Connect framework [20] which provides a comprehensive framework for building data extraction and loading connectors. Compared to other similar systems the advantages are a standardized way of keeping track of job progress and resuming on failures, as well as simplified scaling due to the reliance on Kafka for balancing between processes. In our experience this greatly simplifies operation, creating new connectors as well as being agnostic to low latency or batch extraction/load as all the data is loaded into a queue and this can happen both in real-time or periodically.

In the context of the architecture these represent the framework for extracting data from almost any system a large organization might have. As each connectors deals with rather different types of systems, these connectors are also a reference implementation for most types of storage systems.

- JDBC connector
- File Stream connector
- Elastic Search connector
- HDFS connector
- Binary File connector
- Others

An important functionality required for building a useful Data Vault model is correct data partitioning and handling data schema changes. If the model stops functioning after column name changes or a full scan is always required to execute common queries, then the model has diminished value as an organization wide repository of data.

As an example let's take sensor data or documents arriving in a stream to the platform. A very common query one would need to do to analyze this data is to sort

by date. If this data is stored as is in a single flat directory, this can be an extremely expensive query. For example, car sensor data may increase by terabytes each month. To address this common query pattern, we employ time based partitioning, which enables efficient filtering queries on specific data partitions. We employ a time based partitioning with a maximum granularity on a month. The actual record assignment to time partition depends on what delivery semantics we are using. Process-time (when the event was received) and event-time (when the event actually happened). The choice if either depends on data source.

Now let's also consider what happens if a column name or data type changes. This can lead to inconsistency and even break some process that are already running if this data is ingested. To solve this all data is stored in a efficient binary file format, that supports schema evolution, Apache Avro [18] This means that event is stored in an organized manner with the current schema always stored with the data, which makes the described example much easier to handle.

Another very important aspect is exactly once delivery semantics. We are storing our data in multiple storage services, such as the Hadoop File System(HDFS) [12] and Elastic Search [13] In the case of Elastic Search, which supports updates, data can be written multiple times, without greatly impacting the system, as each write essentially overwriting parts of the record. In the case of HDFS, data can be append-only by design. For which the HDFS connector is developed in such a way that a datum is ingested only once, based on its ID and latest state of ingestion being stored in a transaction log stored in HDFS and a two phased commit process. A first commit is happening when the data is read from Kafka and then a second one after it has been successfully written in HDFS. The second commit is written in the transaction log stored in HDFS.

4. Information Marketplace

One of the biggest challenges faced by an Organization when exploring possible use cases for Data Scientists is knowledge sharing and transfer. Large Organizations have a large number of departments, which vary widely based on their size, project they take on and the way these projects are completed and documented. This leads a large variety of data sources, column names and documentation being created on the same subject by a large number of stakeholders from different departments some of whom might not be part of the Organization anymore.

This leads to challenges for Data Scientists and the IT department, which have to identify the relevant information and people or documents describing the data, especially when the project involves more than one data source.

With the use of the described model, we can provide Data Scientists with easy access to well partitioned and self-describing data, which should simplify the problem.

On the other hand, what is provided, is a technical description of the data. This approach shows where the data comes from, how and when it was stored, and what

did it look like at a specific point time without breaking compatibility across the dataset. What is missing from this model is a functional description of the data,what the data actually means, how it is used and by whom. For example, documentation generated by the organizations. The problem is, such documentation is usually scattered around different departments and is large in volume. So it is often unclear if certain topics are documented or not and if yes, how to access them.

To bridge this issue, there are large undertakings for an Organization wide change management and pushes for standardization. On a technical level this changes translate to the centralization and standardization of project related documentation as well as rigid data views in a central database. This role cannot be filled by a normal Enterprise Data Warehouse and thus requires the creation of Organization wide specialty tools and repositories for Data and Knowledge and complicated integration layers providing each department with access and management capabilities.

In reality this is a vast and complex process and can cost a large amount of money and resources from the side of the Organization and in some cases might decrease productivity. Each individual department has an approach of managing projects and in most cases such a monolithic system allows for less flexibility for individual departments. This may lead to decreased productivity. The learning period for a complete change and standardization of such processes can bring an entire department to a halt for an extended period of time.

As one of the components of our system we propose an advanced text search and schema analysis based approach, which is simpler on the organizational level as it does not require to be integrated organization-wide, and as a minimum provides a much simplified approach for Data Scientists to explore the data. The Information Marketplace(IMP) is a tool that provides an Expert with an easy to use and rich way of exploring data and connections of data.

Now we will outline the functionality in more detail and technical implementation, such as the architecture and algorithms used.

4.1 Functionality

The goal is to provide all relevant information (data, descriptions, contact links) and make the existing environment searchable and discover new connections in the organizational data. The Information Marketplace provides the user with a guided search of:

- Project Documentation
- Knowledge Owners and Information
- Metadata
- Data Sources Schema and Structure
- Data Profiles

This includes information on both data source and project of the Organization as well as all new projects and data created using the platform. This means that the

Information Marketplace not only contains information about the data sources and documentation of the organisation, but there is a feedback loop that feeds all the generated data and documentation back into the IMP. Benefits are exploitative information discovery and faster implementation of analytical and data science use cases due to direct access to relevant project documentation and overview of related data as well as more information on what other people have tried on the platform.

The functionality is exposed as a reactive text-based search of the outlined data types. The search provides the following

- main views of the data:
- Document Full-Text Search View
- Search View based on Author, Time, File-Format, Department, Datasource
- Search View based on extracted keywords, contexts, summaries
- Related document View, based on contexts and keywords
- Datasource profile view

All these views include, match highlighting, facets for author, document type, topic, date and etc. Each document can also be viewed in more detail and the classified by data sources, author, extracted topics and user comments are outlined.

The relevance score of each document is calculated based on text matches to the document content, the topics describing that document, data source and authors.

The Information marketplace also contains and displays all the automatically extracted schema data from all data sources, this is based on a central Schema Registry, which is automatically populated for all data in the Data Vault.

4.2 Implementation Details

The Information Marketplace itself a stage based stream processing application. This architecture was used for its flexibility, and for its simplicity when reasoning about scalability and fault-tolerance. The IMP holds an index of 10s of GBs of documents and this is expected to grow even larger in the near future. Apart from the documents themselves, large topic models and word vectors models are used an applied on a stream of incoming documents. The high level architecture is presented in Fig. 4.

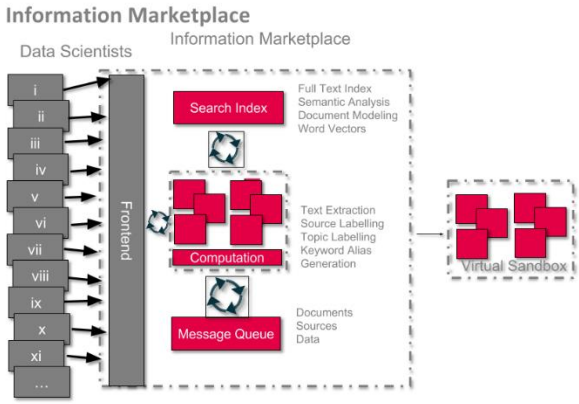


Fig. 4. IMP high level architecture

1. There are 8 principal stages of processing a document goes through before being accessible for searching:
2. Document load from source
3. Document original format to plain text conversion with metadata extraction
4. Map document to specific organizational data source
5. Extract document summary
6. Extract semantic representation of the document(contexts)
7. Find closely related/similar documents
8. Find Departments a data source might refer to
9. Index document

4.2.1 Document load and detection

This stage contains a distributed directory watcher, which watches for filesystem event and emits these events into a distributed message queue. Documents may reside on a filesystem in every Department. We need to detect all the files already there and detect the creation of new files to fulfill the low-latency requirement.

4.2.2 Document conversions

This stage implements a distributed streaming application, which consumes the events from the watcher and extracts plain-text content and metdata. This process scales by launching more instances. This is done manually or dynamically based on the stream of incoming documents. Processes can be pre-allocated based on the number of documents committed and average processing time. This functionality is

based on the work presented in [17] but solves the optimization problem by applying an evolutionary algorithm similarly to [7].

The consumption is balanced across the processes using the native Kafka group balancing and offset management [19].

4.2.3 Document context extraction stage

This stage extracts the contextual topics of each documents. The documents are modeled using [23] [22]. This processes are implemented with both learning and inference functions. We are using Online Latent Dirichlet Allocation, which allows online updates to the model based on incoming documents. So each document received is added to the model and after that the topics are extracted, this allows us to continuously update the model as well as extract topics from documents. The updated model is periodically committed to disk for storage, scaling is achieved by keeping multiple copies of the process running. The predicted topics are attached to the document as a metadata field, this is necessary later on for ranking the matches.

4.2.4 Document data source classification

This stage is responsible for attaching a data source label to the document. Often it is not known if a document is referring to a certain data source in the company. This is achieved by using set distance between the schemas in the schema registry, short descriptions of the sources and sentence per sentence chunks of the incoming document, based on a modified Jacard Distance metric [21].

Scaling in achieved by launching more of these processes. The consumption is balanced across the processes.

4.2.5 Document Index

This stage is the process that actually indexes the data into a search engine, in this case Elastic Search is used to create a reverse token index of the documents. The data is keyed with the filenames and timestamps and a unique offset is generated via a hash function to ensure, that no duplicate data is written into elastic search even during node failures.

5. Analytical Sandbox Environment

With the use of the Data Hub and the IMP, a Data Scientist can find and request data required for their use cases in a simplified fashion and build up analytical use cases, such as the one described in the previous chapters.

But another issue often faced by large organizations is that even if the data is stored in an easily consumed format, and the Data Scientist knows what data they want to investigate, there are a number of points there are still unclear. Such as how to actually access and process the data, where to develop reproducible results on this data, how and where these will be deployed and used in production.

To cover this points a few things are missing, such as:

- A Consistent environment definition for use case development and deployment
- A flexible yet secure environment for the Data Scientists and Analysts to work
- On demand access to services and processing power
- Dynamic Security service
- A Collaboration Service
- Fault tolerant, flexible storage for project data and models

To this end we propose the analytical sandbox environments, which are generated, immutable environments provided to Data Scientists and Analysts and where they can build up their project on the data. It is a fully isolated environment, where the user can install or download any extra tools they require and is accessible via an analytical and console view. The environment serves as a gateway to the data and the processing back-end of the cluster. The way the sandbox is defined makes it inherently self-describing, thus making it simpler to deploy in production as the software requirements, resource requirements and data requirements are already defined when requesting the sandbox.

5.1 Implementation

The sandbox environment is also implemented on top of the ODP architecture. Each sandbox is developed as a separate service running inside the environment. We try to provide as many built in services commonly used by Analysts and Data Scientist is Data-Driven use cases. Scientific environment, such as Python, R, Scala and processing environment such as Spark and Hive are included alongside the Hadoop environment.

To provide full isolation for security and resource sharing reasons the sandbox is implemented as a Linux container running a specified set of services, such as:

- Secure Shell access
- Ipython Console, R console, Scala Console
- Hive, Hue, Hadoop
- Pyspark, Spark, RSpark

The services are running in a single environment with shared resource. For collaboration a code repository is provided for each sandbox project and can be shared amongst collaborates. For convenience and fault tolerance the sandbox user directory is mounted unto a Network mount, running on top of a distributed File System to ensure data does not get lost on sandbox restarts as the entire environment is recreated on failures.

As we are running the sandboxes in a shared environment, network isolation and sharing is a large concern. To this end we use a network abstraction to allocate a

private IP address for each sandbox [24] [25]. All the user facing services are integrated into the Organizations central user and rights management platform.

This solution has shown to scale well when only limited hardware is available. As the sandboxes are running in a shared environment with a single resource scheduler controlling all the resource, scaling this solution would be trivial as new sandboxes could be allocated to new nodes both on premise and cloud.

6. Conclusion

We described a proposed end-to-end environment for creating and running Data-Driven projects at a large scale Enterprise. The described platform can efficiently manage the resources large number of users and services. On the data modelling side we proposed a flexible way to organize and transform stored Data Sets in order to become ready to answer analytical questions and generate value. We proposed a flexible way for discovering data and interconnections of the data, based on metadata, functional descriptions and Documentation, in an automated and intelligent way, while not requiring a full departmental restructure. We also proposed a dynamic and scalable sandbox environment to allow collaborative and shared creation of Data Science use cases based on the data and simplify the deployment of these use cases into the production. We believe the approach, proposed platform and its architecture provide a well structured environment to simplify Data-Driven projects at large organizations.

References

[1]. Rahman, Nayem, and Fahad Aldhaban. "Assessing the effectiveness of big data initiatives."2015 Portland International Conference on Management of Engineering and Technology (PICMET). IEEE, 2015.

[2]. Davenport, Thomas H, and Jill Dych´e. "Big data in big companies."International Institute for Analytics, 2013

[3]. Apache Mesos. http://mesos.apache.or, 2015.

[4]. Dunning, Ted, and Ellen Friedman. Streaming Architecture: New Designs Using Apache Kafka and Mapr Streams. O'Reilly Medi, 2016.

[5]. Welsh, Matt, D. Culler, and E. Brewer. "SEDA: an architecture for highly concurrent server applications.".Proceedings of the 18th Symposium on Operating Systems Principles (SOSP-18), Banff, Canada, 2001.

[6]. Verma, Abhishek, et al. "Large-scale cluster management at Google with Borg."//Proceedings of the Tenth European Conference on Computer Systems. ACM, 2015.

[7]. Artyom Topchyan, Tigran Topchyan. Muscle-based skeletal bipedal locomotion using neural evolution. Ninth International Conference on Computer Science and Information Technologies Revised Selected Papers1-6, 2013.

[8]. Shearer C. The CRISP-DM model: the new blueprint for data mining. J Data Warehousing ;5:1, 22, 2000

[9]. Dan Linstedt. Super Charge your Data Warehouse. Dan Linstedt. ISBN 978-0-9866757-1-3, 2010.

[10]. A. Maksai, J. Bogojeska and D. Wiesmann. "Hierarchical Incident Ticket Classification with Minimal Supervision,". IEEE International Conference on Data Mining, Shenzhen,, 2014, pp.923-928, 2014
[11]. Alex Gorelik. The Enterprise Big Data Lake: Delivering on the Promise of Hadoop and Data Science in the Enterprise. O'Reilly Medi, 2016
[12]. Tom White. Hadoop: The definitive guide. O'Reilly Medi, 2012
[13]. Dixit, Bharvi. Elasticsearch essentials, 2016
[14]. Marz, Nathan, and James Warren. Big Data: Principles and best practices of scalable real-time data systems. Manning Publications Co, 2015
[15]. Michael Hausenblas and Nathan Bijnens. Lambda Architecture. http://lambda-architecture.net/, 2015
[16]. Patil, Preeti S.; Srikantha Rao; Suryakant B.Patil. Optimization of Data Warehousing System: Simplification in Reporting and Analysis. //IJCA Proceedings on International Conference and workshop on Emerging Trends in Technology (ICWET) (Foundation of Computer Science) 9 (6):33–37, 2011
[17]. Newell, Andrew, et al. "Optimizing distributed actor systems for dynamic interactive services.".Proceedings of the Eleventh European Conference on Computer Systems. ACM, 2016
[18]. Apache Avro Project. https://avro.apache.org/docs/current, 2015.
[19]. Apache Kafka Project. http://kafka.apache.org/documentation.html, 2015.
[20]. Confluent Kafka-Connect. http://docs.confluent.io/2.0.0/connect, 2015.
[21]. Cohen, William, Pradeep Ravikumar, and Stephen Fienberg. "A comparison of string metrics for matching names and records.".Kdd workshop on data cleaning and object consolidation. Vol. 3, 2003
[22]. Hoffman, Matthew, Francis R. Bach, and David M. Blei. "Online learning for latent dirichlet allocation.".Advances in neural information processing systems, 2010
[23]. Blei, David M, Andrew Y. Ng, and Michael I.Jordan. "Latent dirichlet allocation.".Journal of machine Learning research 3.Jan: 993-1022, 2003
[24]. Project Calico. https://www.projectcalico.org, 2015.
[25]. Jain, Raj and Subharthi Paul. "Network virtualization and software defined networking for cloud computing: a survey.".IEEE Communications Magazine 51.11, 24-31, 2013

Поддержка выполнения проектов, ориентированных на данные, в современных предприятиях

Топчян А.Р. <a.topchyan@reply.de>
Ереванский государственный университет,
0025, Армения, г. Ереван, ул. А. Манукяна, дом 1

Аннотация. С ростом объема и спроса на данные основными проблемами организаций, которые пытается реализовать проекты, становится не только то, чтобы технически собрать, очистить, интегрировать данные и обеспечить к ним доступ, а в большей степени обеспечение понимания того, как и зачем их следует использовать. Отсутствует взаимопонимание на логическом и техническом уровнях между специалистами по обработке и анализу данных, ИТ-подразделениями и бизнес-подразделениями, поскольку неясно, откуда происходят данные, как они выглядит, что

содержат, и как их следует обрабатывать в контексте существующих систем. В этой статье мы представляем платформу для исследования и обработки данных, что позволяет выполнять ориентированные на данные проекты без полной переделки организационной структуры предприятия при наличии поддержки требуемых процессов и технических средств.

Keywords: проекты, ориентированные на данные; crisp; Hadoop; data vault; sandbox; Mesos; Kafka

DOI: 10.15514/ISPRAS-2016-28(3)-13

Для цитирования: Топчян А.Р. Поддержка выполнения проектов, ориентированных на данные, в современных предприятиях. Труды ИСП РАН, том 28, вып, 3, 2016, стр. 209-230. DOI: 10.15514/ISPRAS-2016-28(3)-14

Список литературы

[1]. Rahman, Nayem, and Fahad Aldhaban. Assessing the effectiveness of big data initiatives. 2015 Portland International Conference on Management of Engineering and Technology (PICMET). IEEE, 2015.
[2]. Thomas H. Davenport and Jill Dyche. Big data in big companies. International Institute for Analytics, 2013
[3]. Apache Mesos. http://mesos.apache.or, 2015.
[4]. Ted Dunning and Ellen Friedman. Streaming Architecture: New Designs Using Apache Kafka and Mapr Streams. O'Reilly Medi, 2016.
[5]. Matt Welsh D. Culler, and E. Brewer. SEDA: an architecture for highly concurrent server applications. Proceedings of the 18th Symposium on Operating Systems Principles (SOSP-18), Banff, Canada, 2001.
[6]. Abhishek Verma et al. Large-scale cluster management at Google with Borg. Proceedings of the Tenth European Conference on Computer Systems. ACM, 2015.
[7]. Artyom Topchyan, Tigran Topchyan. Muscle-based skeletal bipedal locomotion using neural evolution. Ninth International Conference on Computer Science and Information Technologies Revised Selected Papers1-6, 2013.
[8]. Shearer C. The CRISP-DM model: the new blueprint for data mining. J Data Warehousing; 5:1, 22, 2000
[9]. Dan Linstedt. Super Charge your Data Warehouse. 1-3, 2010.
[10]. A. Maksai, J. Bogojeska and D. Wiesmann. Hierarchical Incident Ticket Classification with Minimal Supervision,. IEEE International Conference on Data Mining, Shenzhen,, 2014, pp.923-928, 2014
[11]. Alex Gorelik. The Enterprise Big Data Lake: Delivering on the Promise of Hadoop and Data Science in the Enterprise. O'Reilly Medi, 2016
[12]. Tom White. Hadoop: The definitive guide. O'Reilly Medi, 2012
[13]. Bharvi Dixit. Elasticsearch essentials, 2016
[14]. Nathan Marz and James Warren. Big Data: Principles and best practices of scalable real-time data systems. Manning Publications Co, 2015
[15]. Michael Hausenblas and Nathan Bijnens. Lambda Architecture. http://lambda-architecture.net/, 2015
[16]. Preeti S. Patil; Srikantha Rao; Suryakant B.Patil. Optimization of Data Warehousing System: Simplification in Reporting and Analysis. IJCA Proceedings on International

Conference and workshop on Emerging Trends in Technology (ICWET) (Foundation of Computer Science) 9 (6):33–37, 2011
[17]. Newell, Andrew, et al. Optimizing distributed actor systems for dynamic interactive services. Proceedings of the Eleventh European Conference on Computer Systems. ACM, 2016
[18]. Apache Avro Project. https://avro.apache.org/docs/current, 2015.
[19]. Apache Kafka Project. http://kafka.apache.org/documentation.html, 2015.
[20]. Confluent Kafka-Connect. http://docs.confluent.io/2.0.0/connect, 2015.
[21]. William Cohen Pradeep Ravikumar, and Stephen Fienberg. A comparison of string metrics for matching names and records. Kdd workshop on data cleaning and object consolidation. Vol. 3, 2003
[22]. Matthew Hoffman, Francis R. Bach, and David M. Blei. Online learning for latent dirichlet allocation. Advances in neural information processing systems, 2010
[23]. David M. Blei, Andrew Y. Ng, and Michael I.Jordan. Latent dirichlet allocation. Journal of machine Learning research 3.Jan: 993-1022, 2003
[24]. Project Calico. https://www.projectcalico.org, 2015.
[25]. Raj Jain and Subharthi Paul. Network virtualization and software defined networking for cloud computing: a survey. IEEE Communications Magazine 51.11, 24-31, 2013

Виды признаков и их роль в дифференцировании классов при оценке не полностью описанного объекта[1]

[1, 2] *В.Н. Юдин <yudin@ispras.ru>*
[1, 3] *Л.Е. Карпов <mak@ispras.ru>*
[4] *В.Ю. Абрамов <v_abramov@list.ru>*
[1] *Институт системного программирования РАН,*
Россия, 109004, г. Москва, ул. А. Солженицына, д. 25.
[2]*Московский областной научно-исследовательский клинический институт им. М.Ф. Владимирского, Россия, 129110, г. Москва, ул. Щепкина, д. 61/2*
[3]*Московский государственный университет имени М.В. Ломоносова,*
Россия, 119991, г. Москва, Ленинские горы, д. 1
[4]*НИИ скорой помощи им. Н.В. Склифосовского,*
Россия, 129010, г. Москва, Большая Сухаревская площадь, д. 3.

Аннотация. Разработанный метод в рамках прецедентного подхода к принятию решений позволяет решить проблему выбора наиболее подходящих прецедентов в условиях, когда объект исследования не полностью описан и оценивается неоднозначно. Особенность предлагаемого подхода – в том, что он ориентирован на работу в условиях нефиксированного набора признаков (атрибутов). Это актуально для многих приложений, особенно, при поддержке врачебных решений, когда на процесс принятия решений накладываются ограничения по времени и ресурсам. Чтобы добиться успеха, необходимо дифференцировать возможную принадлежность объекта, расширив его признаковое пространство. Эта задача, в свою очередь, сводится к изучению роли признаков и их сочетаний (по аналогии с дифференциальной диагностикой и семиотикой в медицине). Для выбора порядка, извлечения недостающих признаков, используются введенные понятия: ранг, устойчивые сочетания признаков, частота появления, доступность признака и категории объектов.

Ключевые слова: добыча данных; вывод на основе прецедентов; база прецедентов; дифференциальный ряд; мера близости; устойчивые сочетания признаков.

DOI: 10.15514/ISPRAS-2016-28(3)-14

[1] Работа поддержана грантами Российского фонда фундаментальных исследований № 15-01-02362 и № 15-07-02355.

Для цитирования: Юдин В.Н., Карпов Л.Е., Абрамов В.Ю. Виды признаков и их роль в дифференцировании классов при оценке не полностью описанного объекта. Труды ИСП РАН, том 28, вып. 3, 2016 г., стр. 231-240. DOI: 10.15514/ISPRAS-2016-28(3)-14

1. Введение

Вывод на основе прецедентов – метод принятия решений, в котором используются знания о предыдущих ситуациях или случаях (прецедентах). В такой терминологии прецедент рассматривается как *объект*, включающий в себя описание проблемы, описание решения проблемы и результат применения решения (исход). Накопленная совокупность прецедентов, наполняемая как смоделированными типовыми случаями, так и случаями из практики, образует так называемую базу прецедентов. При рассмотрении новой проблемы (текущего случая) в базе прецедентов находится похожий прецедент. Можно попытаться использовать ранее принятое для него решение, возможно, адаптировав к текущему случаю, вместо того, чтобы искать решение каждый раз сначала. После того, как обработка текущего случая завершится, новый прецедент должен быть внесен в базу прецедентов вместе со своим решением для возможного последующего использования.

Однако чтобы найти наиболее подходящий прецедент, нужно иметь способ измерения близости прецедента и текущего случая. Часто используемым методом в выборе наиболее подходящих прецедентов является *метод ближайшего соседа*. В его основе лежит тот или иной способ измерения степени близости прецедента и текущего случая. В качестве основы измерений в пространстве всех признаков можно ввести какую-либо метрику, определив в этом пространстве точку, соответствующую текущему случаю. На основе выбранной метрики можно отыскивать ближайшую точку, которая и представит прецедент. К сожалению, во многих случаях ввести метрику не удается. Тогда вместо метрики используется так называемая мера близости. Это означает, что вместо метрического пространства используется топологическое.

2. Структуризация базы прецедентов

Один из способов определения меры близости – структуризация множества прецедентов, например, разбиение базы прецедентов на классы эквивалентности, при котором все случаи одного и того же класса считаются равными. В основу такого разбиения кладутся знания о предметной области (фоновое знание), полученные с помощью методов добычи данных (Data Mining) – классификации и кластеризации [1, 2, 3].

В задачах распознавания образов обычно предполагается, что в основе описаний объектов лежит набор признаков, общий для объектов всех классов (за основу принято признаковое описание случая, когда он описывается набором своих характеристик). Иными словами, классы и исследуемые объекты располагаются в едином признаковом пространстве. В реальных

приложениях это условие часто не выполняется. Как сами окружающие объекты, так и описания классов, могут иметь собственные пространства признаков. Например, в медицине каждое заболевание характеризуется своим набором существенных признаков. И, наконец, исследуемый случай может иметь набор показателей, не совпадающий с наборами показателей введенных в систему классов (в медицине – заболеваний), часто из-за дефицита времени, ресурсов, а иногда и квалификации исследователя.

Ряд признаков, которыми обладает исследуемый случай, может не входить в общее признаковое пространство имеющихся классов, а некоторые признаки могут оказываться несущественными для данного конкретного случая. Такие признаки в дальнейшем не будут нами рассматриваться. С другой стороны, у исследуемого случая по разным причинам могут отсутствовать признаки (например, не проделаны важные измерения, не завершены важные исследования), которые являются существенными по отношению к некоторым классам.

3. Дифференцирование классов выбором разделяющего признака

Уже довольно давно в мире развивается технология, во многом опирающаяся на методы рассуждения по прецедентам [4-7]. В ИСП РАН на базе разработанного исследовательского программного комплекса «Универсальный Анализатор» и системы поддержки врачебных решений «Спутник врача», создаваемой на клинической базе МОНИКИ им. М. Ф. Владимирского, также проводится разработка новой технологии [2, 8-9]. В развиваемом подходе база прецедентов состоит из совокупности прецедентов и описаний классов, каждое из которых включает в себя перечень существенных признаков (причем классы – это структура, накладываемая на совокупность прецедентов сверху). Оценить случай – значит выявить его принадлежность тому или иному классу. Отношения между текущим случаем и классами выявляются в проекциях классов на пространство признаков объекта – текущего случая. Недостаточно полно описанный объект может попасть в проекцию класса, к которому он на самом деле не принадлежит, только потому, что у него не хватает признака, который дифференцировал бы его от этого класса. Проиллюстрируем это на простом примере.

Два непересекающихся класса, A и B (рис. 1), описаны в пространстве признаков $\{x_1, x_2\}$. Текущий случай O представлен одним признаком x_1, признак x_2 отсутствует. В пространстве признаков $\{x_1\}$ проекции классов пересекаются, и объект попадает в это пересечение.

Классы нужно дифференцировать, добавляя значения недостающих признаков для текущего случая. В медицине подобная задача носит название *дифференциальная диагностика* [9]. На практике подобное добавление может быть затруднено из-за нехватки средств, времени или оборудования. Но

главная причина заключается в том, что реальные приложения редко укладываются в рамки фиксированного признакового пространства.

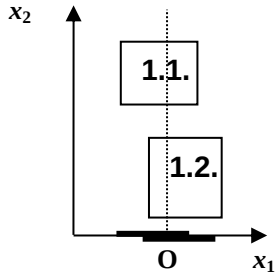


Рис. 1. Одновременное отнесение недостаточно полно описанного объекта к двум классам.

Fig. 1. Not fully described object may be assigned to several different classes.

Формально сущность предложенного метода оценки не полностью описанных объектов сводится к следующему:

- описание объекта (случая) – набор признаков.
- описание класса – многомерный параллелепипед в пространстве признаков, минимально объемлющий прецеденты класса.
- оценка объекта – сравнение случая с проекциями классов на пространство своих признаков.
- *дифференциальный ряд* случая – набор классов, в пересечение проекций которых объект попал.
- если объект попал в область пересечения проекций классов, то наиболее близкими к нему считаются прецеденты этих классов, находящиеся в той же области пересечения. В этом заключается смысл искомой *меры близости* [2, 3, 8] отражающей сходство текущего объекта (случая) и выбранного прецедента.

В зависимости от сложности пересечения, все прецеденты делятся на группы. Находящиеся в общей с текущим случаем области пересечения, естественно считать более близкими к нему, чем те, что находятся только в одном из классов. В конечном счете, прецеденты самого высокого ранга близости находятся в области пересечения всех классов, образующих дифференциальный ряд текущего случая (рис. 2).

Первоначальный отбор прецедентов может не дать ощутимого результата. Например, наличие в текущем случае всего лишь одного признака «высокая температура» (в медицине это носит название лихорадка неясного генеза) даст обилие пересекающихся классов. Тогда нужно либо согласиться, что с таким набором признаков проблему не решить, либо наращивать набор исследуемых признаков.

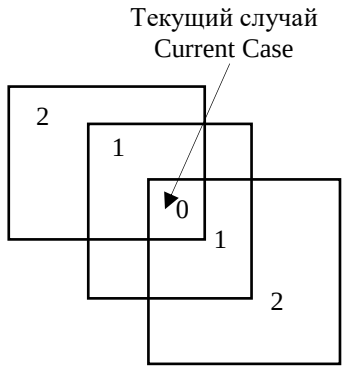


Рис. 2. Степени близости прецедентов (цифрами обозначены расстояния между текущим случаем и прецедентами).

Fig. 2. Similarity degrees (digits designate the degree of similarity between the new case and existing cases).

4. Признаки и их роль в дифференцировании классов

Задача исследователя – расширить признаковое пространство текущего случая так, чтобы однозначно оценить его принадлежность тому или иному классу. Когда случай находится в области пересечения классов, в условиях нехватки времени или ресурсов выявлять все оставшиеся признаки нет возможности. Нужно сформулировать критерий извлечения новых признаков для разделения классов, *определяя приоритеты этого извлечения*. Для этого используются несколько новых понятий: устойчивые сочетания признаков, ранг признака в классе по его степени информативности. Предлагается ввести еще ряд дополнительных критериев отбора: по частоте использования признака в приложении, по категории объекта, по доступности признака.

Не все признаки, которые присутствуют в описании класса, одинаково информативны. Так, в медицине, есть так называемые патогномоничные симптомы, которые имеют абсолютное диагностическое значение (в частности, маркеры рака, гепатита, инфаркта) и позволяют установить наличие заболевания. В общем случае, отвлекаясь от медицины и ее приложений, мы называем признаки класса, имеющие наибольшую информативность, *идеальными*. Они однозначно идентифицируют свой класс, а в других классах не встречаются. Но в той же медицине патогномоничные симптомы не всегда обнаруживаются при соответствующих болезнях, либо обнаруживаются не во всех стадиях или не при всех формах течения. При отсутствии подобных симптомов необходимо принимать во внимание ряд других признаков, относительно более характерных, чем другие, для сопоставляемого заболевания. Есть признаки, появляющиеся в классе с

вероятностью, в несколько раз превышающей вероятность их появления в прецедентах других классов. Назовем такие признаки *детерминирующими* (билирубин, печеночные ферменты при гепатитах). Конечно, для окончательной оценки не следует ориентироваться на один такой признак, следует обязательно учитывать его сочетание с другими.

И, наконец, часто встречается еще одна группа признаков – *сопутствующие*. Они не являются характерными признаками класса (в медицине, например, это симптомы, которые могут сопутствовать основному заболеванию: лихорадка, скорость оседания эритроцитов и т. д.). Их наличие можно считать необходимым, но не достаточным условием принадлежности к классу. Роль сопутствующих признаков в дифференцировании классов ничтожна. Резюмируя, будем считать, что признаки в описании каждого класса в базе прецедентов ранжированы, а именно, подчинены отношению порядка: идеальные – детерминирующие – сопутствующие. Достоверно идентифицировать состояние объекта может только идеальный признак. При его отсутствии, даже если к рассмотрению привлечен целый ряд детерминирующих признаков, говорить о принадлежности к классу можно лишь условно.

При выявлении любых дополнительных признаков должны учитываться их ранги в каждом из классов дифференциального ряда. Само собой разумеется, что исследование признака, который во всех классах ряда относится к сопутствующим, не даст такого эффекта, как если бы это был признак более высокого ранга.

Для окончательного определения класса, к которому относится текущий случай, не всегда можно ориентироваться на единственный признак, нужно учитывать его возможную связь с другими признаками. Так, в медицине особое диагностическое значение имеет устойчиво наблюдаемая совокупность симптомов, определяемая как синдром (семиотика – направление в медицине, где изучаются симптомы различных заболеваний, в особенности их сочетания и их роль в дифференциальной диагностике, так называемая синдромная специфичность). Здесь видна прямая связь с методом выявления знаний при добыче данных, который носит название *анализ ассоциаций*. Этот метод весьма полезен и часто успешен при обработке описаний классов в базе прецедентов на предмет выявления устойчивых сочетаний и рангов признаков.

Вводимое понятие *Устойчивые сочетания признаков* – дополнительный путь к дифференцированию классов. Взаимосвязь *Признаки - Сочетания признаков* похожа на взаимосвязь *Признаки - Классы*. И в том, и в другом случае – это связь многие-к-многим. Если первая связь изначально была отражена в структуре данных базы прецедентов, то вторая только воплощается на текущем этапе. По аналогии с классами, в базу прецедентов заносятся сочетания и входящие в них признаки. На практике такой подход уже давно используется. В медицине синдром как устойчивый набор признаков может

указывать не на одно, а на ряд заболеваний. С другой стороны, заболевание может проявляться не одним, а рядом синдромов. Оба эти факта указывают на связь *Классы - Сочетания признаков* как на связь многие-к-многим. Эта связь тоже должна быть отражена в базе прецедентов.

Текущий случай при его оценке в базе прецедентов попадает в дифференциальный ряд сочетаний признаков, где в перекрестии находятся признаки случая, а в остальной части лепестков – признаки, пока отсутствующие в описании случая, но которые при наличии смогут образовать с первыми устойчивое сочетание. Естественно, при принятии решения, какой из признаков выявлять в первую очередь, выбирается сочетание, где в наборе признаков случая не хватает только одного признака из известного устойчивого сочетания (или наименьшего числа таких признаков).

Итак, если первая стадия оценки случая – получение дифференциального ряда классов, то вторая – получение дифференциального ряда сочетаний признаков. По связи *Классы - Сочетания признаков* можно выбрать набор классов, который соответствует этим сочетаниям. Третья стадия – к двум наборам применяется операция конъюнкции, в результате которой возникает уменьшенный дифференциальный ряд классов.

Итак, когда текущий случай находится в области пересечения классов, выявлять все его недостающие признаки в условиях нехватки времени или ресурсов нет возможности. В этой ситуации для выбора рекомендуется использовать введенные ранее понятия *Ранг признака* и *Устойчивые сочетания признаков*.

Попытаемся описать остальные критерии отбора:

- Новый признак хотя бы в одном из классов дифференциального ряда должен иметь высокий ранг (идеальный или детерминирующий, но не сопутствующий).
- Выявляются в первую очередь признаки, которые имеют бóльшую частоту появления (на уровне всей базы прецедентов). Эту величину можно получить приблизительно, поддерживая в описании базы прецедентов при каждом признаке счетчик использований данного признака, значение которого делится на значение счетчика использований всех признаков базы. В базе вводится еще один дополнительный тег для признака.
- Учитывается категория исследуемого объекта. Выбираются только признаки, соответствующие данной категории. Для примера можно опять обратиться к медицине: конкретное лечебное учреждение во многом определяет контингент больных, находящихся там, их заболевания, характерные симптомы. В базу прецедентов вводится сущность *Категория*, и отношение *Категория - Признак* вида многие-к-многим.

- Выбираются наиболее доступные признаки. Это довольно широкий термин, под которым понимается ряд параметров: стоимость выявления признака, наличие аппаратуры для его выявления, целый ряд параметров предпочтения (в медицине известен термин неинвазивность) и ряд других. В зависимости от приложения, это один или несколько дополнительных тегов признака на уровне базы.

5. Заключение

Подход к оценке не полностью описанных объектов востребован, особенно в такой области, как медицина, хотя медицинскими приложениями этот подход не ограничивается. Понятие дифференциального ряда и мера близости в оценке объектов являются оригинальными, они разработаны и подробно описаны в более ранних работах авторов.

Список литературы

[1]. I. Watson and F. Marir. Case-based reasoning: A review. The Knowledge Engineering Review, 9(4):327-354, 1994.

[2]. Л. Е. Карпов, В. Н. Юдин. Интеграция методов добычи данных и вывода по прецедентам в медицинской диагностике и выборе лечения. Сборник докладов 13-й Всероссийской конференции "Математические методы распознавания образов (ММРО-13)", октябрь, 2007, стр. 589-591.

[3]. Valery. Yudin, Leonid Karpov. The Case-Based Software System for Physician's Decision Support. Sami Khari, Lenka Lhotska, Nadia Pisanti (eds.), "Information Technology in Bio- and Medical Informatics, ITBAM 2010", Proceedings of the First International Conference, Bilbao, Spain. Lecture Notes in Computer Science Sublibrary: SL 3, Springer Verlag, Berlin, Heidelberg, 2010, pp. 78-85.

[4]. Agnar Aamodt, Enric Plaza. Case-based reasoning: Foundational issues, methodological variations, and system approaches. AI Communications, 7(1). pp. 39-59, 1994.

[5]. I. Watson and F. Marir. Case-based reasoning: A review. The Knowledge Engineering Review, 9(4):327-354, 1994.

[6]. Mario Lenz, Brigitte Bartsch-Spörl, Hans-Dieter Burkhard. Case-Based Reasoning Technology: From Foundations to Applications. Springer, 2003.

[7]. Lorraine McGinty, David C. Wilson, Case-Based Reasoning Research and Development: 8th International Conference on Case-Based Reasoning, ICCBR 2009 Seattle, WA, USA, July 20-23, 2009 Proceedings (Lecture Notes in Artificial Intelligence).

[8]. В.Н. Юдин. Мера близости в системе вывода на основе прецедентов. Доклады 12-й Всероссийской конференции Математические Методы Распознавания Образов (ММРО-12), МАКС Пресс, Москва 2005, стр. 241-244 .

[9]. В. Н. Юдин, Л. Е. Карпов, А. В. Ватазин. Методы интеллектуального анализа данных и вывода по прецедентам в программной системе поддержки врачебных решений, М., Альманах клинической медицины, № 17 в двух частях, Москва 2008, ч. 1, стр. 266-269.

Feature's types and their role in differentiating classes for estimation of not fully described object

[1,2] *V.N. Yudin <yudin@ispras.ru>*
[1,3] *L.E. Karpov <mak@ispras.ru>*
[4] *V.Y. Abramov <v_abramov@list.ru>*
[1]*Institute for System Programming of the Russian Academy of Sciences, 25, Alexander Solzhenitsyn st., Moscow, 109004, Russia*
[2]*Moscow Regional Research and Clinical Institute n.a. M.F.Vladimirsky 61/2, Shepkina, Moscow, 129110, Russia*
[3]*Lomonosov Moscow State University, GSP-1, Leninskie Gory, Moscow, 119991, Russia*
[4]*N.V. Sklifosovsky Research Institute for Emergency Medicine of Moscow Healthcare Department, 3, Bolshaya Sukharevskaya Square, Moscow, 129010, Russia*

Abstract. Authors are developing precedent approach to solving the problem of optimal decision making. The method they develop makes it possible to make the most adequate precedent selection in conditions where the object under consideration is not fully described, and cannot be estimated unambiguously. The originality of the approach offered by authors is in its focus on functioning with varying set of features (attributes). It is important for different applications, but it is especially important while supporting physician's decision making, who often has a lack of time and resources. The method presumes the need in differentiating possible object membership that may be done by widening of its feature space. This task may in its turn be reduced to investigating of feature's roles and their combinations (as in differential diagnosis and semiotics in medicine). In order to determine in what way should one retrieve missing features the authors offer to use the following conceptions: range, persistent feature combination, frequency of occurrence, availability of a feature, and object category. This work is supported by Russian Foundation for Basic Research (projects 15-01-02362 and 15-07-02355).

Keywords: data mining; case-based reasoning; case base; differential set; measure of closeness; persistent feature combinations.

DOI: 10.15514/ISPRAS-2016-28(3)-14

For citation: Yudin V.N., Karpov L.E., Abramov V.Y. Feature's types and their role in differentiating classes for estimation of not fully described object. *Trudy ISP RAN /Proc. ISP RAS*, vol. 28, issue 3, 2016, pp. 231-240 (in Russian). DOI: 10.15514/ISPRAS-2016-28(3)-14

References

[1]. Ian H. Witten, Eibe Frank and Mark A. Hall. Data Mining: Practical Machine Learning Tools and Techniques. 3rd Edition. Morgan Kaufmann, 2011. pp. 664.
[2]. L. E. Karpov, V. N. Yudin. Integration of data Mining and Case-Based Reasoning methods in medical diagnostics and treatment choosing. Sbornik dokladov 13-j

Vserossijskoj konferentsii Matematicheskie metody raspoznavaniya obrazov [Proc. of 13-th All-Russian conference Math. methods of pattern recognition], October 2007, MAKS Press, 2007, pp. 589-591 (in Russian).

[3]. Valery. Yudin, Leonid Karpov. The Case-Based Software System for Physician's Decision Support. Sami Khari, Lenka Lhotska, Nadia Pisanti (eds.), "Information Technology in Bio- and Medical Informatics, ITBAM 2010", Proc. of the First International Conference, Bilbao, Spain. Lecture Notes in Computer Science Sublibrary: SL 3, Springer Verlag, Berlin, Heidelberg, 2010, pp. 78-85.

[4]. Agnar Aamodt, Enric Plaza. Case-based reasoning: Foundational issues, methodological variations, and system approaches. AI Communications, 7(1), 1994, pp. 39-59.

[5]. I. Watson and F. Marir. Case-based reasoning: A review. The Knowledge Engineering Review, 9(4):327-354, 1994.

[6]. Mario Lenz, Brigitte Bartsch-Spörl, Hans-Dieter Burkhard. Case-Based Reasoning Technology: From Foundations to Applications. Springer, 2003.

[7]. Lorraine McGinty, David C. Wilson, Case-Based Reasoning Research and Development: 8th International Conference on Case-Based Reasoning, ICCBR 2009 Seattle, WA, USA, July 20-23, 2009 Proceedings (Lecture Notes in Artificial Intelligence).

[8]. Yudin V. N. Measure of closeness in Case-based Reasoning System. Dokladyi 12-y Vserossiyskoy konferentsii Matematicheskie Metodyi Raspoznavaniya Obrazov (MMRO-12) [Proc. of 12-th All-Russia Conf. on Mathematical Methods in Pattern Recognition], MAKS Press, Moscow, 2005, pp. 241-244 (in Russian).

[9]. Yudin V. N., Karpov L. E., Vatazin A. V. Data mining and case-based reasoning methods in physician's decision making support software system. Almanah klinicheskoy meditsinyi [Almanac of Clinical Medicine], Moscow, 2008, v. 17, part 1, pp. 266-269, (in Russian).

Румянцев Ю.А., Захаров П.Н., Абрашитова Н.А., Шматок А.В., Рыжих В.О., Гудимчук Н.Б., Атауллаханов Ф.И. Применение ПЛИС для расчета деполимеризации микротрубочки методом броуновской динамики. Труды ИСП РАН, том 28, вып. 3, 2016 г., с. 241-266.

Применение ПЛИС для расчета деполимеризации микротрубочки методом броуновской динамики

[1,3] *Ю.А.Румянцев <yarumyantsev@gmail.com>*
[2] *П.Н. Захаров <pavel.n.zaharov@gmail.com>*
[1] *Н.А. Абрашитова <natascha.abraschitowa@gmail.com >*
[1] *А.В. Шматок <papercompute@gmail.com >*
[3] *В.О. Рыжих <vo.ryzhikh@mail.ru>*
[2,3,4] *Н.Б.Гудимчук <gudimchuk@phys.msu.ru>*
[2,3,4] *Ф.И.Атауллаханов <ataullakhanov.fazly@gmail.com>*
[1] *НПО РОСТА,*
123103, Россия, Москва, ул. Живописная, д. 3 к. 1
[2] *Центр теоретических проблем физико-химической фармакологии РАН,*
119991, Россия, Москва, ул. Косыгина 4
[3] *Московский государственный университет имени М.В. Ломоносова,*
119991, Россия, Москва, Ленинские горы, д. 1.
[4] *Федеральный научно-клинический центр детской гематологии, онкологии и иммунологии имени Дмитрия Рогачева,*
117997, Россия, Москва ГСП-7, ул. Саморы Машела, д. 1

Аннотация. В данной работе рассмотрена аппаратная реализация расчета деполимеризации белковой микротрубочки методом броуновской динамики на кристалле программируемой логической интегральной схеме (ПЛИС) Xilinx Virtex-7 с использованием высокоуровневого транслятора с языка Си Vivado HLS. Реализация на ПЛИС сравнивается с параллельными реализациями этого же алгоритма на многоядерном процессоре Intel Xeon и графическом процессоре Nvidia K40 по критериям производительности и энергоэффективности. Алгоритм работает на броуновских временах и поэтому требует большого количества нормально распределенных случайных чисел. Оригинальный последовательный код был оптимизирован под многоядерную архитектуру с помощью OpenMP, для графического процессора - с помощью OpenCL, а реализация на ПЛИС была получена посредством высокоуровневого транслятора Vivado HLS. В работе показано, что реализация на ПЛИС быстрее CPU в 17 раз и быстрее GPU в 11 раз. Что касается энергоэффективности (производительности на ватт), ПЛИС была лучше CPU в 227 раз и лучше GPU в 75 раз. Ускоренное на ПЛИС приложение было разработано с помощью SDK, включающего готовый проект ПЛИС, имеющий PCI Express интерфейс для связи

с хост-компьютером, и софтверные библиотеки для общения хост-приложения с ПЛИС ускорителем. От конечного разработчика было необходимо только разработать вычислительно ядро алгоритма на языке Си в среде Vivado HLS, и не требовалось специальных навыков ПЛИС разработки.

Ключевые слова: Высокопроизводительные вычисления; ПЛИС; микротрубочки; высокоуровневый синтез; броуновская динамика

DOI: 10.15514/ISPRAS-2016-28(3)-15

Для цитирования: Румянцев Ю.А., Захаров П.Н., Абрашитова Н.А., Шматок А.В., Рыжих В.О., Гудимчук Н.Б., Атауллаханов Ф.И. Применение ПЛИС для расчета деполимеризации микротрубочки методом броуновской динамики. Труды ИСП РАН, том 28, вып. 3, 2016 г., стр. 241-266. DOI: 10.15514/ISPRAS-2016-28(3)-15.

1. Введение

Высокопроизводительные вычисления проводят на процессорах (CPU), объединенных в кластеры и/или имеющих аппаратные ускорители – графические процессоры на видеокартах (GPU) или программируемые логические интегральные схемы (ПЛИС) [1]. Современный процессор сам по себе является отличной платформой для высокопроизводительных вычислений. К достоинствам CPU можно отнести многоядерную архитектуру с общей когерентной кэш-памятью, поддержку векторных инструкций, высокую частоту, а также огромный набор программных средств, компиляторов и библиотек, обеспечивающий высокую гибкость программирования. Высокая производительность платформы GPU основывается на возможности запустить тысячи параллельных вычислительных потоков на независимых аппаратных ядрах. Для GPU доступны хорошо зарекомендовавшие себя средства разработки (CUDA, OpenCL), снижающие порог использования GPU платформы для прикладных вычислительных задач. Несмотря на это, в последнее десятилетие ПЛИС все чаще стали использоваться в качестве платформы для ускорения задач, в том числе использующих вещественные вычисления [2]. ПЛИС обладают уникальным свойством, резко отличающим их от CPU и GPU, а именно возможностью построить конвейерную аппаратную схему под конкретный вычислительный алгоритм. Поэтому, несмотря на значительно меньшую тактовую частоту, на которой работают ПЛИС (по сравнению с CPU и GPU), на некоторых алгоритмах на ПЛИС удается добиться большей производительности [3]–[5]. С другой стороны, меньшая частота работы означает меньшее энергопотребление, и ПЛИС практически всегда более эффективны, чем CPU и GPU, если использовать метрику «производительность на ватт» [5].

Одним из классических приложений, требующих высокопроизводительных вычислений является метод молекулярной динамики, использующийся для

расчета движения систем атомов или молекул. В рамках этого метода взаимодействия между атомами и молекулами описываются в рамках законов Ньютоновской механики с помощью потенциалов взаимодействия. Расчет сил взаимодействия проводится итеративно и представляет существенную вычислительную сложность, учитывая большое количество атомов/молекул в системе и большое количество расчетных итераций. Ускорению расчетов молекулярной динамики было уделено много внимания в литературе в различных системах: суперкомпьютерах [6], кластерах [7], специализированных под молекулярно-динамические расчеты машинах [8]–[10], машинах с ускорителями на основе GPU [11] и ПЛИС [12]–[17]. Было продемонстрировано, что ПЛИС может являться конкурентной альтернативой в качестве аппаратного ускорителя для молекулярно-динамических вычислений во многих случаях, однако на сегодняшний день не существует консенсуса о том, для каких именно задач и алгоритмов выгоднее применять платформу ПЛИС.

В данной работе мы рассматриваем важный частный случай молекулярной динамики – броуновскую динамику. Основная особенность метода броуновской динамики по сравнению с молекулярной динамикой заключается в том, что, молекулярная система моделируется более грубо, т.е. в качестве элементарных объектов моделирования выступают не отдельные атомы, а более крупные частицы, такие как отдельные домены макромолекул или целые макромолекулы. Молекулы растворителя и другие малые молекулы в явном виде не моделируются, а их эффекты учитываются в виде случайной силы. Таким образом удается значительно снизить размерность системы, что позволяет увеличить интервал времени, покрываемый модельными расчетами на порядки.

Нам неизвестны описанные в литературе попытки исследовать эффективность ПЛИС по сравнению с альтернативными платформами для ускорения задач броуновской динамики. Поэтому мы предприняли исследование данного вопроса на примере задачи моделирования деполимеризации микротрубочки методом броуновской динамики.

Микротрубочки – это трубки диаметром около 25 нм и длиной от нескольких десятков нанометров до десятков микрон, состоящие белка тубулина и входящие в состав внутреннего скелета живых клеток. Ключевой особенностью микротрубочек является их динамическая нестабильность, т.е. возможность спонтанно переключаться между фазами полимеризации и деполимеризации [18]. Это поведение важно прежде всего для захвата и перемещения хромосом микротрубочками во время клеточного деления. Кроме того, микротрубочки играют важную роль во внутриклеточном транспорте, движении ресничек и жгутиков и поддержании формы клетки [19]. Механизмы, лежащие в основе работы микротрубочек, исследуются уже несколько десятков лет, но лишь недавно развитие вычислительных

технологий позволило описывать поведение микротрубочек на молекулярном уровне. Наиболее подробная молекулярная модель динамики микротрубочек, созданная недавно нашей группой на основе метода броуновской динамики, была реализована базе CPU и позволяла рассчитывать времена полимеризации/деполимеризации микротрубочек порядка нескольких секунд [20]. Это пролило свет на ряд важных аспектов динамики микротрубочек, однако, тем не менее, многие ключевые экспериментально наблюдаемые явления остались за рамками теоретического описания, т.к. они происходят в микротрубочках на временах десятков и даже сотен секунд [21]. Таким образом, для прямого сравнения теории и эксперимента критически важно достигнуть ускорения расчетов динамики микротрубочек хотя бы на порядок величины.

В данной работе мы исследуем возможность ускорения расчетов броуновской динамики микротрубочки на ПЛИС и сравниваем результаты, полученные при реализации одного и того же алгоритма динамики микротрубочек на трех разных платформах, по критериям производительности и энергоэффективности.

2. Математическая модель

2.1 Общие сведения о структуре микротрубочки

Структурно микротрубочка представляет собой цилиндр, состоящий из 13 цепочек – протофиламентов (Рис.1).

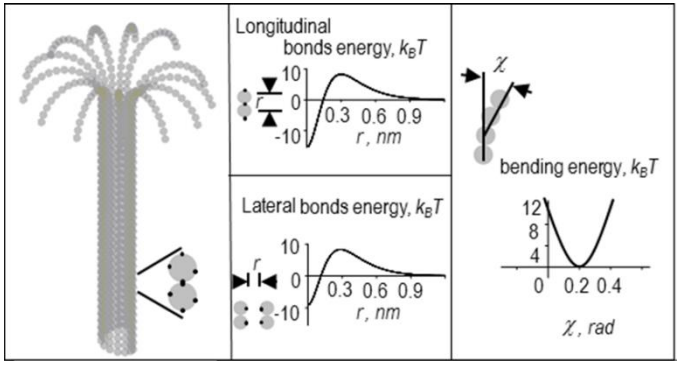


Рис. 1. Слева – схема модели микротрубочки. Серым показаны субъединицы тубулина, черными точками – центры взаимодействия между ними. Справа - вид энергетических потенциалов взаимодействия между тубулинами.

Fig. 1. On the left – the scheme of a microtubule model. Grey shows tubulin subunits, black spots – centers of interaction between them. Right – the kind of energy potential interaction between tubulins.

Каждый протофиламент построен из димеров белка тубулина. Соседние протофиламенты связаны друг с другом боковыми связями и сдвинуты относительно друг друга на расстояние 3/13 длины одного мономера, так что микротрубочка имеет спиральность. При полимеризации димеры тубулина присоединяются к концам протофиламентов, причем протофиламенты микротрубочки стремятся принимать прямую конформацию. При деполимеризации боковые связи между протофиламентами на конце микротрубочки разрываются, и протофиламенты закручиваются наружу. При этом от них случайным образом отрываются олигомеры тубулина.

2.2 Моделирование деполимеризации микротрубочки методом броуновской динамики

Используемая здесь молекулярная модель микротрубочки была впервые представлена в статье [20]. Поскольку задачей настоящего исследования являлось сравнение производительности различных вычислительных платформ, мы ограничились моделированием только деполимеризации микротрубочки.

Вкратце, микротрубочка моделировалась как набор сферических частиц, представляющих собой мономеры тубулина. Мономеры могли двигаться только в соответствующей им радиальной плоскости, т.е. в плоскости, проходящей через ось микротрубочки и соответствующий протофиламент. Таким образом, положение и ориентация каждого мономера полностью определялись тремя координатами: двумя декартовыми координатами центра мономера и углом ориентации. Каждый мономер имел четыре центра взаимодействия на своей поверхности: два центра бокового взаимодействия и два центра продольного взаимодействия. Энергия тубулин-тубулинового взаимодействия зависела от расстояния r между сайтами взаимодействия на поверхности соседних субъединиц и от угла наклона между соседними мономерами тубулина в протофиламенте.

Боковые и продольные взаимодействия между димерами тубулина определялись потенциалом, имеющим следующий вид:

$$v\ (r\ ) = A\ \cdot \left(\frac{r}{r_o}\right)^2 \cdot \exp\left(\frac{-r}{r_o}\right) - b\ \cdot \exp\left(\frac{-(r)^2}{d \cdot r_o}\right) \qquad (1)$$

где A и b определяли глубину потенциальной ямы и высоту энергетического барьера, r_0 и d – параметры, задающие ширину потенциальной ямы и форму потенциала в целом. Параметр A принимал различные значения для боковых и продольных связей, так что боковые взаимодействия были слабее продольных, все остальные параметры совпадали для обоих типов связей (полный список параметров и их значений представлен в Table 1 в [20]). Продольные взаимодействия внутри димера моделировались как неразрывные пружины с квадратичным энергетическим потенциалом $u(r)$:

$$u(r) = \frac{1}{2} k \cdot r^2 \qquad (2)$$

где k - жесткость связи тубулин-тубулинового взаимодействия.

Энергия изгиба $g(\chi)$ связана с поворотом мономеров друг относительно друга и также описывалась квадратичной неразрывной функцией:

$$g(\chi) = \frac{B}{2}(\chi - \chi_o)^2 \qquad (3)$$

где χ - угол между соседними мономерами тубулина в протофиламенте, χ_0 - равновесный угол между двумя мономерами, B - изгибная жесткость.

Полная энергия микротрубочки записывалась следующим образом:

$$U_{total} = \sum_{n=1}^{13} \sum_{i=1}^{K_n} \left(v_{k,n}^{lateral} + v_{k,n}^{longitudinal} + u_{k,n} + g_{k,n} \right) \qquad (4)$$

где n - номер протофиламента, i - номер мономера в n-ом протофиламенте, K_n - число субъединиц тубулина в n-ом протофиламенте, $v_{k,n}^{lateral}$ - энергия бокового взаимодействия между мономерами, $v_{k,n}^{longitudinal}$ - энергия продольного взаимодействия между димерами.

Эволюция системы рассчитывалось с помощью метода Броуновской динамики [22]. Изначальной конфигурацией микротрубочки была короткая «затравка», содержащая 12 мономеров тубулина в каждом протофиламенте. Мы рассматривали только деполимеризацию МТ и моделировали все тубулины с равновесным углом $\chi_0 = 0.2$ рад. Координаты всех мономеров системы на i-ой итерации выражались следующим образом:

$$\begin{cases} q_{k,n}^{i} = q_{k,n}^{i-1} - \dfrac{dt}{\gamma_q} \cdot \dfrac{\partial U_{total}}{\partial q_{k,n}^{i}} + \sqrt{2 k_B T \dfrac{dt}{\gamma_q}} \cdot N(0,1) \\ \tau_{k,n}^{i} = \tau_{k,n}^{i-1} - \dfrac{dt}{\gamma_\tau} \cdot \dfrac{\partial U_{total}}{\partial \tau_{k,n}^{i}} + \sqrt{2 k_B T \dfrac{dt}{\gamma_\tau}} \cdot N(0,1) \end{cases} \qquad (5)$$

$$\gamma_q = 6\pi r \eta$$
$$\gamma_\tau = 8\pi r^3 \eta$$

где dt - шаг по времени, U_{total} выражается через (4), k_B - постоянная Больцмана, T- температура, $N(0,1)$ – случайное число из нормального распределения, сгенерированное с помощью алгоритма вихрь Мерсенна [23]. γ_q и γ_τ - вязкостные коэффициенты сопротивления для сдвига и поворота соответственно, рассчитанные для сфер радиуса $r = 2$ нм.

Производная полной энергии по независимым координатам $q^i_{k,n}$ выражалась через боковую, продольную составляющие энергии взаимодействия между соседними димерами и внутри димера, а также энергию изгиба:

$$\frac{\partial U_{total}}{\partial q^i_{k,n}} = \frac{\partial v^{lateral}_{k,n}}{\partial q^i_{k,n}} + \frac{\partial v^{longitudinal}_{k,n}}{\partial q^i_{k,n}} + \frac{\partial u_{k,n}}{\partial q^i_{k,n}} + \frac{\partial g_{k,n}}{\partial q^i_{k,n}} \qquad (6)$$

Для ускорения расчетов были использованы аналитические выражения для всех градиентов энергии:

$$\frac{\partial v^{lateral}_{k,n}}{\partial q^i_{k,n}} = \left(A_{lateral} \cdot \frac{r}{{r_o}^2} \cdot \exp\left(\frac{-r}{r_o}\right) \cdot \left(2 - \frac{r}{r_o}\right) + \frac{2 \cdot b_{lateral} \cdot r}{d \cdot r_o} \cdot \exp\left(\frac{-(r)^2}{d \cdot r_o}\right) \right) \cdot \frac{\partial r}{\partial q^i_{k,n}} \quad (7)$$

$$\frac{\partial v^{longitudinal}_{k,n}}{\partial q^i_{k,n}} = \left(A_{longitudinal} \cdot \frac{r}{{r_o}^2} \cdot \exp\left(\frac{-r}{r_o}\right) \cdot \left(2 - \frac{r}{r_o}\right) + \frac{2 \cdot b_{longitudinal} \cdot r}{d \cdot r_o} \cdot \exp\left(\frac{-(r)^2}{d \cdot r_o}\right) \right) \cdot \frac{\partial r}{\partial q^i_{k,n}} \quad (8)$$

$$\frac{\partial u_{k,n}}{\partial q^i_{k,n}} = k \cdot r \cdot \frac{\partial r}{\partial q^i_{k,n}} \quad (9)$$

$$\frac{\partial g_{k,n}}{\partial q^i_{k,n}} = \frac{\partial g_{k,n}}{\partial \chi_{k,n}} \cdot \frac{\partial \chi_{k,n}}{\partial q^i_{k,n}} = B \cdot (\chi - \chi_o) \cdot \frac{\partial \chi_{k,n}}{\partial q^i_{k,n}} \quad (10)$$

Следует отметить, что размер данной задачи сравнительно мал. Мы рассматривали только 12 слоев мономеров, что дает полное число частиц равное 156. Однако, это нисколько не уменьшает значимость вычислений, т.к. в реальных расчетах достаточно вычислять положение крайних нескольких (порядка 10) слов мономеров, т.к. при росте микротрубочки дальние от конца микротрубочки молекулы тубулина образуют устойчивую цилиндрическую конфигурацию, и брать их в расчет нет смысла.

2.3 Псевдокод алгоритма расчета

Алгоритм является итеративным по времени с шагом 0.2 нс. Существуют массив трехмерных координат молекул, а также массивы сил поперечного (латерального) и продольного (лонгитудального) взаимодействий. На каждой итерации по времени последовательно выполняются два вложенных цикла по молекулам, в первом производится вычисление сил взаимодействия по известным координатам, во втором – обновляются сами координаты. В цикле вычисления сил взаимодействия необходимо прочитать координаты трех молекул, одна центральная и две соседние («левая» и «верхняя», см. Рис.2), а результатом вычисления будет сила поперечного взаимодействия между центральной и левой молекулами и сила продольного взаимодействия между центральной и верхней.

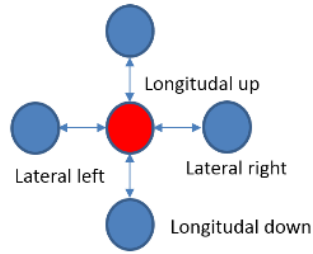


Рис. 2. Схема расположения взаимодействующих субъединиц в модели микротрубочки

Fig. 2. The scheme of arrangement of interacting subunits in the microtubule model

В итоге после этого цикла оказываются вычисленными все силы взаимодействия между всеми молекулами. В цикле обновления координат по известным силам вычисляются изменения координат, а также берутся в расчет случайные добавки для учета Броуновского движения. Таким образом, псевдокод алгоритма можно записать следующим образом.

Вход: массив координат молекул M = {x, y, teta}. Граничные условия на силы взаимодействия.

Выход: массив координат M после K шагов по времени

```
for t in {0.. K-1} do
    for i in {0.. 13} // количество протофиламентов
    for j in {0.. 12} do // количество слоев молекул
        Mc <- M[i,j]
        Ml <- M[i+1,j]
        Mu <- M[i,j+1]
        // по формулам (7, 8, 9, 10)
        F_lat[i,j]  <- calc_calteral(Mc, Ml)
        F_long[i,j] <- calc_long(Mc, Mu)
    end for

    for i in {0.. 13}
    for j in {0.. 12} do
        // по формулам (5)
        M[i,j] <- update_coords(F_lat[i,j], F_long[i,j])
    end for
end for
```

3. Программная реализация на CPU и GPU

3.1 Реализация на CPU

Была предпринята попытка максимально распараллелить код на CPU Intel Xeon E5-2660 2.20GHz под управлением ОС Ubuntu 12.04 с помощью библиотеки OpenMP. Параллельная секция начиналась до цикла по времени. Циклы расчета сил взаимодействия и обновления координат были распараллелены с помощью директивы *omp for schedule(static)*, между циклами была вставлена барьерная синхронизация. Массивы, содержащие силы взаимодействия и координаты молекул, были объявлены как *private* для каждого потока.

При реализации расчетов на CPU было обнаружено, что размер задачи не позволял ее эффективно распараллелить. Зависимость времени выполнения одной итерации от числа параллельных потоков была немонотонна. Минимальное время расчета одной итерации по времени было получено при использовании всего 2 потоков (ядер CPU). Объясняется это тем что, с увеличением количества потоков растет время на копирование данных между потоками и на их синхронизацию. При этом размер задачи очень мал, чтобы выигрыш от увеличения количества ядер превысил эти накладные расходы. При этом эксперименты показали, что задача слабо масштабируется при увеличении размера (weak scaling), т.е. при одновременном увеличении размера задачи и числа параллельных потоков время вычисления оставалось примерно одинаковым. В итоге лучшим результатом на данном CPU было 22 мкс на одну итерацию по времени при использовании двух ядер CPU. Код не был векторизован из-за сложности вычислений сил взаимодействия.

3.2 Реализация на GPU

Мы запускали OpenCL реализацию на граф процессоре Nvidia Tesla K40. Циклы, вычисляющие силы взаимодействия и обновления координат были распараллелены, главный цикл по времени был итеративным. Были реализованы два варианта – с одной и несколькими рабочим группами (work groups). В первом случае было выделено по одному рабочему потоку (work item) на каждую молекулу. В каждом потоке был цикл по времени, в котором вычислялись силы и координаты молекулы потока. При этом применялась барьерная синхронизация после вычисления сил и после обновления координат. В этом случае участие хоста не требовалось для вычислений, он только занимался управлением и запуском ядер.

Во втором случае были два типа потоков, в одном просто вычислялись силы для одной молекулы, во втором – обновлялись координаты. Главный цикл по времени был на хосте, который управлял запуском и синхронизацией ядер на каждой итерации цикла по времени.

Наибольшая производительность была получена в расчетах с одной группой потоков и барьерной синхронизацией между ними. Без использования генераторов псевдослучайных чисел одна итерация вычислялась в течение 5 мкс, если использовать один генератор чисел на все потоки, то время работы возрастало до 9 мкс, а при максимальном заполнении общей памяти (shared memory) удавалось включить 7 независимых генераторов, при этом время вычисления одной итерации по времени составило 14 мкс, что было в 1.57 раза быстрее реализации на CPU.

Загруженность ядер GPU составила 7% от одного мультипроцессора (SM), при этом общая память, где размещались массивы сил, координат и буферы данных генераторов псевдослучайных чисел, была заполнена на 100%. Т.е. с одной стороны размер задачи был явно мал для полной загрузки GPU, с другой стороны при увеличении размера задачи пришлось бы использовать глобальную DDR память, что могло бы привести к ограничению роста производительности.

4. Реализация на ПЛИС

4.1 Описание платформы

Вычисления на ПЛИС производились на платформе ПЛИС RB-8V7 производства фирмы НПО “Роста”. Она представляет собой 1U блок для установки в стойку. Блок состоит из 8 кристаллов ПЛИС Xilinx Virtex-7 2000T. Каждая ПЛИС имеет 1 GB внешней DDR3 памяти и PCI Express x4 2.0 интерфейс к внутреннему PCIe коммутатору. Блок имеет два интерфейса PCIe x4 3.0 к хост-компьютеру через оптические кабели, которые должны быть соединены со специальным адаптером, установленным в хост-компьютер.

В качестве хост-компьютера был использован сервер с CPU Intel Xeon E5-2660 2.20 GHz, работающий под управлением ОС Ubuntu 12.04 LTS – такой же как и для вычислений просто на CPU с помощью OpenMP. Программное обеспечение, работающее на CPU хост-компьютера «видит» блок RB-8V7 как 8 независимых ПЛИС устройств, подключенных по шине PCI Express. Далее будет описываться взаимодействие CPU только с одной ПЛИС XC7V72000T, при этом система позволяет использовать ПЛИС независимо и параллельно.

Ускоренное с помощью ПЛИС приложение было разработано с помощью SDK со следующей моделью. На CPU хост-компьютера (далее просто CPU) работает основная программа, которая использует ускоритель ПЛИС для наиболее вычислительно емких процедур. CPU передает данные в ускоритель и обратно через внешнюю DDR память, подключенную к ПЛИС, а также управляет работой вычислительного ядра в ПЛИС. Вычислительное ядро создается заранее на языке C/C++, верифицируется и транслируется в RTL код с помощью средства Vivado HLS. RTL код вычислительного ядра вставляется в основной ПЛИС проект, в котором уже реализована необходимая логика

управления и передачи данных, включающая PCI Express ядро, DDR контроллер и шину на кристалле (Рис. 3). Основной ПЛИС проект иногда называют Board Support Package (BSP), он разрабатывается производителем оборудования, и от пользователя не требуется его модификации. Вычислительное ядро HLS после запуска само обращается в DDR память, считывает оттуда входной буфер данных для обработки и записывает туда же результат вычислений. На уровне языка C++ обращение в память происходит через аргумент функции верхнего уровня вычислительного ядра типа указатель.

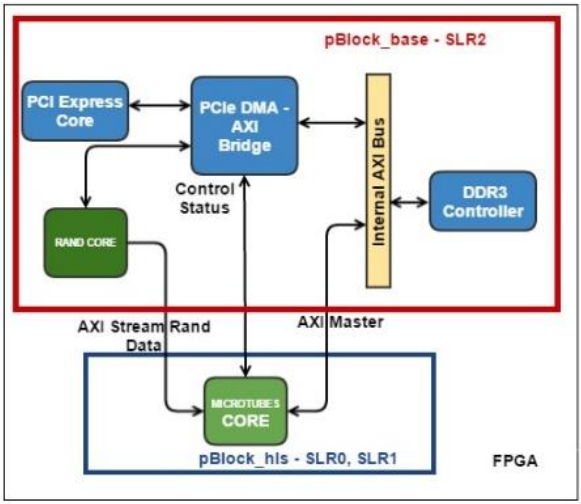


Рис. 3. Блок-схема проекта ПЛИС. Синим и желтым цветами отмечены блоки, входящие в BSP. Зеленым обозначены вычислительные HLS ядра. Также обозначено разбиение ядер проекта на блоки (pBlocks) для наложения пространсвенных ограничений при трассировке.

Fig. 3. A block diagram of the FPGA design. Blue and yellow colors mark blocks included in the BSP. Green color mark HLS computing cores. Splitting the cores of the project on the blocks (pBlocks) is also marked to impose space restrictions on tracking.

Для создания ускоренного приложения была разработана методология, состоящая из нескольких шагов. Во-первых, оригинальный последовательный код компилировался в среде Vivado HLS, и проверялось, что скомпилированный таких образом код не изменяет выходных данных опорного последовательного кода. Во-вторых, из этого кода выделялась основная вычислительная и подходящая для ускорения часть; эта часть отделялась от основного кода с помощью функции-обертки. После чего создавалось две копии такой функции и логика проверки на соответствие результатов обеих частей. Первая копия была опорной реализацией алгоритма

в Vivado HLS, а вторая была оптимизирована для трансляции в RTL код. Оптимизации включали в себе переписывание кода, такие как использование статических массивов вместо динамических, использование специальных функций для ввода/вывода в HLS ядро, методы экономии памяти и переиспользования результатов вычислений. После каждого изменения результат функции сравнивался с результатом опорной реализации. Другим методом оптимизации было использование специальных директив Vivado HLS, не меняющих логическое поведение, но влияющих на конечную производительность RTL кода. На данной стадии следует оставаться до тех пор, пока не будут получены удовлетворительные предварительные результаты трансляции C в RTL, такие как производительность схемы и занимаемые ресурсы.

Следующая стадия – это имплементация разработанного вычислительного ядра в системе Vivado вне контекста основного проекта. Здесь задача добиться отсутствия временных ошибок уже разведенного дизайна внутри разработанного вычислительного ядра. Если на этом этапе наблюдаются временные ошибки, то можно применять другие параметры имплементации, либо возвращаться на предыдущую стадию и пытаться изменить C++ код или использовать другие директивы.

На следующей стадии необходимо имплементировать вычислительное ядро уже вместе с основным проектом и его временными и пространственными ограничениями. На данной стадии также необходимо добиться отсутствия временных ошибок. Если они наблюдаются, то можно либо изменить частоту работы вычислительной схемы, наложить другие пространственные ограничения на размещение схемы на кристалле, либо опять заняться изменением C++ кода и/или использовать другие директивы.

Последняя стадия разработки – это проверка на соответствие результатов, полученных на реальном запуске в железе и с помощью опорной модели на CPU. Проходит она на небольшом промежутке времени, при этом считается, что на более длительных запусках (когда сравнить с CPU уже проблематично) ПЛИС решение выдает правильные результаты.

4.2 Работа в среде Vivado HLS

В работе использовалось два Vivado HLS ядра (Рис. 3): основное ядро, реализующее алгоритм молекулярной динамики микротрубочек (MT ядро), и ядро для генерации псевдослучайных чисел (RAND ядро). Нам пришлось разделить алгоритм на два вычислительных ядра по следующей причине. Кристалл ПЛИС Virtex-7 2000T – это самый большой кристалл ПЛИС семейства Virtex-7 на рынке. Он на самом деле состоит из четырех кристаллов кремния, соединенных на подложке множеством соединений и объединенных в один корпус микросхемы. По терминологии Xilinx каждый такой кристалл называется SLR (Super Logic Region). При использовании таких больших

ПЛИС всегда возникают проблемы с цепями, пересекающими границы SLR. Xilinx рекомендует вставлять регистры на такие цепи с обеих сторон границы SLR.

Полное HLS ядро, включающее и MT и RAND ядра, требовали аппаратных ресурсов больше, чем было доступно в одном SLR, поэтому были цепи, которые пересекали границу независимых кристаллов кремния. На стадии трансляции с языка C++ в RTL Vivado HLS ничего «не знает» о том, какие цепи будут впоследствии пересекать границу, и поэтому не может заранее вставить дополнительные регистры синхронизации. Поэтому мы приняли решение разделить ядра на два, пространственно ограничить их в разные SLR и вставить регистры синхронизации на интерфейсные цепи между ядрами на уровне RTL.

4.2.1 Ядро MT

Данный алгоритм очень хорошо подходит для реализации на ПЛИС, потому для сравнительно небольшого количества данных из памяти (координаты двух молекул) необходимо вычислить сложную функцию сил взаимодействия и удается построить длинный вычислительный конвейер.

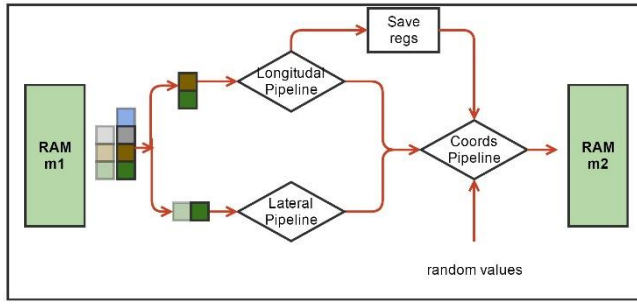


Рис. 4. Блок-схема аппаратной вычислительной процедуры ядра MT. Зеленым обозначены аппаратные блоки памяти для хранения координат молекул. Обозначены вычислительные конвейеры сил и обновления координат, а также блок Save Regs для хранения промежуточных результатов вычислений. Псевдослучайные числа поступают в конвейер обновления координат из другого HLS ядра.

Fig. 4 Block diagram of hardware procedure of MT core. Green color designates hardware memory blocks to store coordinates of molecules. Instruction pipelines to calculate forces and to update coordinates are marked, as well as block Save Regs for storing intermediate results of calculations. The pseudo-random numbers receive in the pipeline for coordinate updates from other HLS core.

Каждая молекула, т.е. мономер тубулина, взаимодействует только с четырьмя своими соседями (Рис. 2). На каждой итерации по времени надо сначала вычислить силы взаимодействия, а затем обновить координаты молекул.

Функции вычисления сил взаимодействия включают в себя множество арифметических, экспоненциальных и тригонометрических операторов. Нашей первой задачей было синтезировать конвейер для этих функций. Рабочим типом данных был вещественный тип float. Vivado HLS синтезировала такие функции в виде конвейеров, работающих на частоте 200 МГц, с латентностью порядка 130 тактов. При этом конвейеры были однотактовые (или, как говорят, с интервалом инициализации равной 1), что означает, что на вход они могли принимать координаты новых молекул каждый такт, а затем после начальной задержки (латентности) – выдавать обновленные значения сил также каждый такт. Выходные силы взаимодействия использовались для обновления координат, что тоже было конвейеризовано. Для обновления каждой координаты каждой молекулы были необходимо независимые псевдослучайные нормально распределенные числа, получаемые из другого HLS ядра. Если взять три молекулы («текущую», «левую» и «верхнюю») то получилось возможным объединить конвейеры вычисления сил и обновления координат в один конвейер, реализующий все вычисления для одной молекулы. Такой конвейер имел латентность равную 191 такт (Рис. 4).

Алгоритм проходит по всем молекулам в цикле. На каждой итерации цикла необходимо иметь координаты трех молекул: одна молекула рассматривается как «текущая», также есть «левая» и «правая» молекулы. Соответственно рассчитываются силы взаимодействия между этими тремя молекулами. Далее при обновлении координат текущей молекулы левая и верхняя компоненты сил взаимодействия брались из расчета на текущей итерации, а нижняя и правая компоненты брались либо из граничных условий, либо с предыдущих итераций из локального регистрового файла Save Regs (Рис. 4).

Количество молекул N в системе было небольшим (13 протофиламентов x 12 молекул = 156 молекул). На каждую молекулу требуется 12 байт. Схема использовала два массива координат m1 и m2, общим объемом меньше 4 КБ, соответственно эти данные легко помещались во внутреннюю память ПЛИС – BRAM, реализованную внутри HLS ядра. Схема была устроена таким образом, что на четных итерациях по времени координаты считывались из массива m1 (и записывались в m2), а на нечетных – наоборот. С точки зрения алгоритма можно было читать и писать в один массив координат, но Vivado HLS не могла создать схему, способную на одном и том же такте читать и писать один и тот же аппаратный массив, что требуется для работы однотактового конвейера. Поэтому было принято решение удвоить количество независимых блоков памяти.

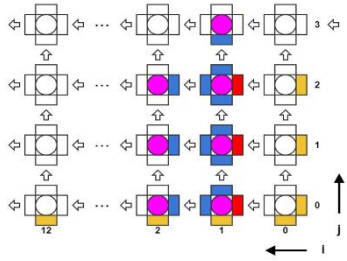


Рис. 5. Схема конвейерного расчета взаимодействий тубулинов в микротрубочке.

Fig. 5. The scheme of pipeline calculating of tubulin interactions in the microtubule.

Оказалось возможным реализовать три полных параллельных конвейера, способных обновлять координаты трех молекул каждый такт (Рис. 5). Тогда во избежание простаивания конвейеров необходимо было увеличить пропускную способность к локальной памяти и читать координаты семи молекул каждый такт. Это проблема легко решилась, практически не меняя исходный C++ код, а лишь за счет использования специальной директивы, физически разбивающей исходный массив данных по четырем независимым аппаратным блокам памяти. Т.к. память BRAM в ПЛИС является двупортовой, то из четырех блоков памяти можно прочитать 8 значений за такт. Но, так как три конвейера требуют координаты 7ми молекул за такт (см рис 5), это решило проблему.

```
#pragma HLS DATA_PACK variable=m1, m2
#pragma HLS ARRAY_PARTITION variable=m1, m2 cyclic factor=4 dim=2
```

Табл. 1: Производительность и утилизация схемы HLS с тремя полными конвейерами

Table. 1: Performance and utilization of HLS scheme with three complete pipelines

Период	L	II	BRAM	DSP	FF	LUT	Утилизация
5 нс	191 такт	1 такт	52	498	282550	331027	Абсолютная
			2 %	23 %	11 %	27 %	Относительная

В табл. 1 приводится утилизация схемы HLS (т.е. количество потребляемых ей аппаратных ресурсов ПЛИС, в абсолютных и относительных единицах для кристалла Virtex-7 2000T) и ее производительность. L – это задержка или латентность схемы, т.е. количество тактов между подачей в конвейер первых входных данных и получением первых выходных данных, II – это интервал инициализации (или пропускная способность) конвейера, означающее через сколько тактов на вход конвейера можно подавать следующие данные.

Утилизация приводится как в абсолютных величинах (сколько требуется триггеров FF или таблиц LUT) для реализации схемы, так и в относительных к полному количеству данного ресурса в кристалле.

Как видно из табл. 1 латентность L полного конвейера была равна 191 такту, при этом каждый конвейер должен был обработать третью часть все молекул, что дает теоретическую оценку времени вычисления одной итерации равную

$$\tau_{ПЛИС} = (L+N/3)*5нс=1.2 мкс$$

Из табл. 1 также видно, что в кристалле осталось еще много неиспользованной логики, но дальше увеличивать количество параллельных конвейеров непрактично. Будет уменьшаться только второе слагаемое, а начальная задержка все равно будет давать значительный вклад во время работы. При этом увеличение количество логики усложнит размещение и трассировку схемы на следующих стадиях разработки проекта в Vivado.

4.2.2 Ядро RAND

Как было указано, алгоритм учитывает Броуновское движение молекул, одним из методов расчета которого является прибавление нормальной случайной добавки к изменению координат на каждой итерации по времени. Необходимо очень много нормально распределенных случайных чисел, на каждую итерацию – по N*3 чисел, что дает поток $420^{*}10^{6}$ чисел/с. Такой поток не может быть загружен с хоста, поэтому его необходимо генерировать внутри ПЛИС «на лету». Для этого, как и в опорном коде для CPU, был выбран генератор вихрь Мерсенна, дающий равномерно распределенные псевдослучаные числа. Далее к ним применялось преобразование Бокса-Мюллера и на выходе получались нормально распределенные последовательности. Исходный открытый код вихря Мерсенна был модифицирован для получения аппаратного конвейера с интервалом инициализации в 1 такт. Алгоритм требует 9 нормальных чисел каждый такт, поэтому ядро RAND включало в себя 10 независимых генераторов вихря Мерсенна, т.к. преобразование Бокса-Мюллера требует два равномерно распределенных числа для получения 2х нормально распределенных. В табл. 2 приводится утилизация ядра RAND.

Табл. 2: Утилизация ядра RAND

Table. 2: Utilization of the RAND core

BRAM	DSP	FF	LUT	Утилизация
30	41	48395	64880	Абсолютная
1.2 %	9 %	0.1 %	5.3 %	Относительная

Видно, что такое ядро требует значительную часть DSP ресурсов кристалла, и это ядро было бы сложно разместить в одном SLR с ядром MT, т.к. сумма

утилизаций двух ядер хотя бы по DSP ресурсу 31% больше чем может вместить один SLR (25 %).

4.3 Создание битстрима

После интеграции вычислительных ядер в Vivado на проект были наложены пространственные ограничения на размещение IP блоков. Используемая ПЛИС Virtex-7 2000T имеет 4 независимых кристалла кремния (SLR0, SLR1, SLR2, SLR3). Было показано, что ядро MT не умещалось в один SLR, поэтому было решено создать два региона размещение (pBlock): pBloch_hls для размещения только MT ядра и pBlock_base для размещения остальных ядер проекта (Рис. 3). Регион размещения pBlock_hls включал в себя SLR0 и SLR1, pBock_base – SLR2. Такой подход позволил разместить логику оптимальным образом, вставить регистры синхронизации на интерфейсы, пересекающие регионы размещения (а значит и SLR) и добиться положительных времянных результатов после трассировки проекта.

5. Результаты

5.1 Производительность

Результаты работы всех трех реализаций (CPU, GPU и ПЛИС) были логически верифицированы относительного оригинального кода и признаны состоятельными. Сравнение производительностей производилось замером времени работы программ на 10^7 итераций алгоритма и вычислением времени, требующегося для расчета одной итерации. При этом производительность GPU и ПЛИС платформ брали в расчет время передачи данных между хост-компьютером и ускорителем.

Для оценки производительности обычно используется метрика операций в секунду. Для данного алгоритма нам оказалось сложным вычислить точное значений вещественных операций, поэтому мы просто сравниваем времена работы алгоритма для вычисления одной итерации, определяя производительность CPU платформы равной 1. Результаты сравнения приводятся в табл. 3, во втором столбце которой приводятся времена вычисления одной итерации алгоритма в микросекундах, а в третьем – относительная производительность платформ.

Табл. 3: Сравнение производительности трех платформ

Table 3: Comparison of the performance of the three platforms

Платформа	Время, мкс	Производительность
CPU	22	1
GPU	14	1,6
ПЛИС	1.3	17

Из таблицы видно, что реализация на GPU быстрее CPU всего в 1.6 раза, в то время как ПЛИС быстрее CPU в 17 раз. Это означает, что ПЛИС быстрее GPU в 11 раз. Полученное экспериментально время работы ПЛИС равно 1.3 мкс на итерацию больше расчетного времени в 1.2 мкс из-за учета накладных расходов на передачу данных по шине PCI Express.

5.2 Энергоэффективность

Для измерения энергопотребления мы использовали следующие средства. Для CPU платформы – утилиту Intel Power Gadget. Для GPU платформы - утилиту Nvidia-smi. Для ПЛИС – специальные программно-аппаратные средства, включенные в состав блока RB-8V7. Во всех случаях замерялась разница в потреблении всего чипа до запуска задачи и во время вычислений. Результаты приведены в таблице 4.

Табл. 4: Сравнение энергопотребления вычислительных платформ

Table 4: Comparison of power consumption of computing platforms

Платформа	Мощность, Вт	E_x, W^{-1}	E_X^{rel}
CPU	89.6	0.011	1
GPU	67	0.033	3
ПЛИС	9.6	2.5	227

Во втором столбце таблицы приводится мощность, выделяющаяся при расчете на разных платформах. В третьем столбце приводятся значения абсолютной энергоэффективности (производительности на Вт) для данной задачи, определяемой по формуле

$$E_x = {}^{P_x}/_{Pow_x}$$

В четвертом столбце приводятся значения относительной энергоэффективности разных платформ для данной задачи, вычисляемой по формуле

$$E_X^{rel} = {}^{E_x}/_{E_{CPU}}$$

Для обеих формул, x = {CPU, GPU, ПЛИС}.

Видно, что у ПЛИС есть большое преимущество в энергоэффективности перед другими платформами, что может сыграть роль в средне и долгосрочной перспективе использования ПЛИС ускорителей в датацентрах при оплате счетов за электроэнергию. Достигается это в первую очередь за счет того, что ПЛИС работают на порядок меньшей частоте.

6. Обсуждение

В ранее опубликованных работах технология ПЛИС неоднократно применялась к решению задач молекулярной динамики [12]–[17]. Исследователям из лаборатории CAAD Бостонского Университета удалось разработать эффективное ядро для расчета короткодействующих межмолекулярных сил, которое было реализовано на плате ProcStar-III (производство фирмы Gidel), с установленным кристаллом ПЛИС Altera Stratix-III SE260. Плата имела PCI Express интерфейс к хост-компьютеру. Было показано, что разработанное ускоренное решение было в 26 раз быстрее чистой реализации на CPU на бенчмарке Apoal. В работе [24] авторы перенесли часть пакета для расчета молекулярной динамики LAMMPS на ПЛИС. Ускоренная часть включала в себя вычисления дальнодействующих взаимодействий. Разработанное аппаратное ядро состояло из четырех одинаковых независимых конвейеров, работающих параллельно. Задача была выполнена на суперкомпьютере Maxwell, каждый узел которого состоит из одного процессора Intel Xeon и двух кристаллов ПЛИС Xilinx Virtex-4 [25]. Авторы заявили, что разработанное ускоренное решение легко масштабировалось на множество узлов суперкомпьютера Maxwell. Из анализа производительности только ускорителя следовало, что на двух узлах компьютера можно было получить ускорение в 13 раз по сравнению с чисто программным решением. Однако полное время работы гибридного решения было хуже чисто программного из-за того, что время на пересылку данных между CPU и внешней памятью SDRAM, подключенной к ПЛИС занимало 96% времени работы всего алгоритма. Но в работе утверждается, что если улучшить интерфейс передачи данных, то можно получить полный выигрыш в скорости в 8-9 раз.

В настоящей работе мы применили ПЛИС к расчету движения ансамбля белковых молекул методом броуновской динамики. Наша программно-аппаратная реализация алгоритма деполимеризации микротрубочки показала, что производительность ПЛИС при расчете одной траектории микротрубочки в 17 раз превосходила производительность CPU и в 11 раз производительность GPU. Полученное ускорение при расчете деполимеризации микротрубочки методом броуновской динамики позволяет осуществить расчет на временах порядка нескольких десятков и даже сотен секунд. Это позволит предсказывать поведение реальных микротрубочек на экспериментально доступных временах и проанализировать механизмы динамической нестабильности микротрубочек, что будет предметом будущей работы в данном направлении.

Полученный выигрыш на задаче броуновской динамики позволяет говорить о перспективности применения ПЛИС для решения данного типа задач. Насколько нам известно, это первая попытка сравнить производительность и

энергоэффективность различных типов аппаратных ускорителей на данном алгоритме.

Долгое время главной проблемой использования ПЛИС являлось отсутствие высокоуровневых средств программирования. Традиционные языки описания аппаратуры всегда требуют значительного времени для реализации алгоритма, в то время как первые высокоуровневые трансляторы [26] генерировали RTL код низкого качества. Однако, несколько лет назад компании Altera и Xilinx стали уделять значительные ресурсы этой проблеме и выпустили на рынок свои высокоуровневые средства программирования (Altera SDK for OpenCL и Xilinx Vivado HLS). Данные трансляторы генерируют намного более эффективный код и позволяют прикладному программисту использовать языки C/C++ (Xilinx) и OpenCL (Altera) для создания качественных аппаратных вычислительных схем. В последнее время появилось множество работ, в которых использовались средства высокоуровневого синтеза для разработки ПЛИС ускорителей [27]–[29]. Например, в работе [28] с помощью средства Vivado HLS реализован алгоритм оптического потока на платформе Xilinx Zynq-7000. Разработанная система имела производительность сравнимую с реализацией на CPU, при этом потребление энергии было в 7 раз меньше. Авторы особенно подчеркивали, что использование средств HLS по сравнению с традиционными RTL языками значительно сократило срок разработки. Использование средства Vivado HLS в ходе выполнения настоящей работы также позволило значительно сократить время и трудоемкость разработки и привлекать к программированию разработчиков, не владеющих специальными навыками работы с ПЛИС. Все это позволяет говорить об ПЛИС как о состоявшейся платформе для высокопроизводительных вычислений в области молекулярной и броуновской динамики.

Признательности

Работа была поддержана грантом РФФИ, проект № 16-04-01862 А. Авторы благодарят НПО РОСТА за предоставленное оборудование. Программы, использованные для расчетов в этом исследовании, могут быть найдены по ссылке: https://github.com/urock/FpgaMicrotubule.

Список литературы

[1]. B. Liu, D. Zydek, H. Selvaraj, and L. Gewali. «Accelerating High Performance Computing Applications: Using CPUs, GPUs, Hybrid CPU/GPU, and FPGAs». *In 2012 13th International Conference on Parallel and Distributed Computing, Applications and Technologies*, 2012, pp. 337–342.

[2]. Wim Vanderbauwhede и K. Benkrid. *High-Performance Computing Using FPGAs*. Springer, 2013.

[3]. J. Fowers, G. Brown, P. Cooke, and G. Stitt. «A Performance and Energy Comparison of FPGAs, GPUs, and Multicores for Sliding-window Applications». *In Proceedings of the ACM/SIGDA International Symposium on Field Programmable Gate Arrays*, New York, NY, USA, 2012, pp. 47–56.

[4]. K. Sano, Y. Hatsuda, and S. Yamamoto. «Multi-FPGA Accelerator for Scalable Stencil Computation with Constant Memory Bandwidth». *IEEE Trans. Parallel Distrib. Syst.* 2014, vol. 25, no. 3, pp. 695–705.

[5]. K. Benkrid, A. Akoglu, C. Ling, Y. Song, Y. Liu, and X. Tian. «High Performance Biological Pairwise Sequence Alignment: FPGA Versus GPU Versus Cell BE Versus GPP». *Int. J. Reconfig. Comput.*, vol. 2012, 2012.

[6]. B. G. Fitch, A. Rayshubskiy, M. Eleftheriou, T. J. C. Ward, M. Giampapa, M. C. Pitman, and R. S. Germain. «Blue Matter: Approaching the Limits of Concurrency for Classical Molecular Dynamics». *In Proceedings of the ACM/IEEE SC 2006 Conference*, 2006, pp. 44–44.

[7]. K. J. Bowers, D. E. Chow, H. Xu, R. O. Dror, M. P. Eastwood, B. A. Gregersen, J. L. Klepeis, I. Kolossvary, M. A. Moraes, F. D. Sacerdoti, J. K. Salmon, Y. Shan, and D. E. Shaw, «Scalable Algorithms for Molecular Dynamics Simulations on Commodity Clusters». *In Proceedings of the ACM/IEEE SC 2006 Conference*, 2006, pp. 43–43.

[8]. Y. Komeiji, M. Uebayasi, R. Takata, A. Shimizu, K. Itsukashi, and M. Taiji. «Fast and accurate molecular dynamics simulation of a protein using a special-purpose computer». *J. Comput. Chem.*, vol. 18, no. 12, pp. 1546–1563, 1997.

[9]. D. E. Shaw, J. P. Grossman, J. A. Bank, B. Batson, J. A. Butts, J. C. Chao, M. M. Deneroff, R. O. Dror, A. Even, C. H. Fenton, A. Forte, J. Gagliardo, G. Gill, B. Greskamp, C. R. Ho, D. J. Ierardi, L. Iserovich, J. S. Kuskin, R. H. Larson, T. Layman, L.-S. Lee, A. K. Lerer, C. Li, D. Killebrew, K. M. Mackenzie, S. Y.-H. Mok, M. A. Moraes, R. Mueller, L. J. Nociolo, J. L. Peticolas, T. Quan, D. Ramot, J. K. Salmon, D. P. Scarpazza, U. Ben Schafer, N. Siddique, C. W. Snyder, J. Spengler, P. T. P. Tang, M. Theobald, H. Toma, B. Towles, B. Vitale, S. C. Wang, and C. Young. «Anton 2: Raising the Bar for Performance and Programmability in a Special-purpose Molecular Dynamics Supercomputer». *In Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, Piscataway, NJ, USA, 2014, pp. 41–53.

[10]. M. Taiji, T. Narumi, Y. Ohno, N. Futatsugi, A. Suenaga, N. Takada, and A. Konagaya. «Protein Explorer: A Petaflops Special-Purpose Computer System for Molecular Dynamics Simulations». *In Supercomputing, 2003 ACM/IEEE Conference*, 2003, pp. 15–15.

[11]. C. Rodrigues, D. Hardy, J. Stone, K. Schulten, and W.-Mei Hwu. «GPU acceleration of cutoff pair potentials for molecular modeling applications». *Proceedings of the 5th conference on Computing frontiers*, 2008, pp. 273–282.

[12]. S. R. Alam, P. K. Agarwal, M. C. Smith, J. S. Vetter, and D. Caliga. «Using FPGA Devices to Accelerate Biomolecular Simulations». *Computer*, vol. 40, no. 3, 2007, pp. 66–73.

[13]. N. Azizi, I. Kuon, A. Egier, A. Darabiha, and P. Chow. «Reconfigurable molecular dynamics simulator». *In 12th Annual IEEE Symposium on Field-Programmable Custom Computing Machines, 2004. FCCM 2004*, pp. 197–206.

[14]. Y. Gu, T. VanCourt, and M. C. Herbordt. «Improved Interpolation and System Integration for FPGA-Based Molecular Dynamics Simulations». *In 2006 International Conference on Field Programmable Logic and Applications*, 2006, pp. 1–8.

[15]. V. Kindratenko and D. Pointer. «A case study in porting a production scientific supercomputing application to a reconfigurable computer». 2006, pp. 13–22.

[16]. R. Scrofano, M. B. Gokhale, F. Trouw, and V. K. Prasanna. «Accelerating Molecular Dynamics Simulations with Reconfigurable Computers». *IEEE Trans. Parallel Distrib. Syst.*, vol. 19, no. 6, 2008, pp. 764–778.

[17]. M. Chiu and M. C. Herbordt. «Molecular Dynamics Simulations on High-Performance Reconfigurable Computing Systems». *ACM Trans Reconfigurable Technol Syst*, vol. 3, no. 4, 2010, pp. 23:1–23:37.

[18]. T. Mitchison and M. Kirschner. «Dynamic instability of microtubule growth». *Nature*, vol. 312, no. 5991, 1984, pp. 237–242.

[19]. A. Desai and and T. J. Mitchison. «Microtubule Polymerization Dynamics». *Annu. Rev. Cell Dev. Biol.*, vol. 13, no. 1, 1997, pp. 83–117.

[20]. P. Zakharov, N. Gudimchuk, V. Voevodin, A. Tikhonravov, F. I. Ataullakhanov, and E. L. Grishchuk. «Molecular and Mechanical Causes of Microtubule Catastrophe and Aging». *Biophys. J.*, vol. 109, no. 12, , 2015, pp. 2574–2591.

[21]. M. K. Gardner, M. Zanic, C. Gell, V. Bormuth, and J. Howard, «Depolymerizing Kinesins Kip3 and MCAK Shape Cellular Microtubule Architecture by Differential Control of Catastrophe». *Cell*, vol. 147, no. 5, , 2011, pp. 1092–1103.

[22]. D. L. Ermak and J. A. McCammon, «Brownian dynamics with hydrodynamic interactions». *J. Chem. Phys.*, v. 69, issue 4, , 1978, pp. 1352–1360.

[23]. M. Matsumoto and T. Nishimura, «Mersenne twister: a 623-dimensionally equidistributed uniform pseudo-random number generator». ACM Trans. Model. Comput. Simul., vol. 8, 1998, pp. 3–30.

[24]. S. Kasap and K. Benkrid, «Parallel processor design and implementation for molecular dynamics simulations on a FPGA-Based supercomputer». *J. Comput.*, vol. 7, no. 6, 2012, pp. 1312–1328.

[25]. R. Baxter, S. Booth, M. Bull, G. Cawood, J. Perry, M. Parsons, A. Simpson, A. Trew, A. McCormick, G. Smart, R. Smart, A. Cantle, R. Chamberlain, and G. Genest, «Maxwell - a 64 FPGA Supercomputer». *In Second NASA/ESA Conference on Adaptive Hardware and Systems (AHS 2007)*, 2007, pp. 287–294.

[26]. J. M. P. Cardoso, P. C. Diniz, and M. Weinhardt, «Compiling for reconfigurable computing: A survey». *ACM Comput. Surv. CSUR*, vol. 42, no. 4, p. 13, 2010.

[27]. Y. Liang, K. Rupnow, Y. Li, D. Min, M. N. Do, and D. Chen, «High-level synthesis: productivity, performance, and software constraints». *J. Electr. Comput. Eng.*, vol. 2012, 2012, p. 1.

[28]. J. Monson, M. Wirthlin, and B. L. Hutchings, «Implementing high-performance, low-power FPGA-based optical flow accelerators in C». IEEE 24th International Conference on Application-Specific Systems, Architectures and Processors, 2013, pp. 363–369.

[29]. T. Hussain, M. Pericàs, N. Navarro, and E. Ayguadé, «Implementation of a Reverse Time Migration kernel using the HCE High Level Synthesis tool». Field-Programmable Technology (FPT), 2011 International Conference, 2011, pp. 1–8.

PGA HPC Implementation of Microtubule Brownian Dynamics Simulations

[1, 3] *Rumyanstev Y.A. <yarumyantsev@gmail.com>*
[2] *Zakharov P.N. <pavel.n.zaharov@gmail.com>*
[1] *Abrashitova N. A. <natascha.abraschitowa@gmail.com >*
[1] *Shmatok A.V. <papercompute@gmail.com >*
[3] *Ryzhikh V.O. <vo.ryzhikh@mail.ru>*
[2,3,4] *Gudimchuk N.B. <gudimchuk@phys.msu.ru>*
[2,3,4] *Ataullakhanov F.I. <ataullakhanov.fazly@gmail.com>*
[1] *ROSTA LTD,*
Jivopisnaya str., 3/1, Moscow, 123103, Russia
[2] *Center for Theoretical Problems of Physico-chemical Pharmacology, Russian Academy of Sciences,*
Kosigina str, 4, Moscow, 119991, Russia
[3] *Lomonosov Moscow State University,*
Leninskie gori, 1, Moscow, 119991, Russia
[4] *Federal Research Center of Pediatric Hematology, Oncology and Immunology named after Dmitriy Rogachev*
Samory Mashela, 1, Moscow, 117997, Russia (FRC-PHOI)

Abstract. This paper presents high performance simulation of microtubule molecular dynamics implemented on Xilinx Virtex-7 FPGA using high level synthesis tool Vivado HLS. FPGA implementation is compared to multicore Intel Xeon CPU and Nvidia K40 GPU implementations in terms of performance and energy efficiency. Algorithm takes into account Brownian motion thus heavily uses normally distributed random numbers. Original sequential code was optimized for different platforms using OpenMP for CPU, OpenCL for GPU and Vivado HLS for FPGA. We show that in terms of performance FPGA achieved 17x speed up against CPU and 11x speedup against GPU for our best optimized CPU and GPU versions. As to power efficiency, FPGA outperformed CPU 227 times and GPU 75 times. FPGA application is developed using SDK, which has Board Support Package including FPGA project framework where accelerator kernel (designed in Vivado HLS) IP core is to be integrated, and host-side libraries used to communicate with FPGA via PCI Express. Developed flow does not require expert FPGA skills and can be used by programmer with little knowledge of hardware design methodology that could use C\C++ language for complete development of FPGA accelerated solution.

Keywords: HPC; FPGA; Microtubule; HLS; Brownian Dynamics

DOI: 10.15514/ISPRAS-2016-28(3)-15

For citation: Rumayanstev Y.A., Zakharov P.N., Abrashitova N. A., Shmatok A.V., Ryzhikh V.O., Gudimchuk N.B., Ataullakhanov F.I. PGA HPC Implementation of Microtubule Brownian Dynamics Simulations. *Trudy ISP RAN / Proc. ISP RAS*, vol. 28, issue 3, 2016, pp. 241-266 (in Russian). DOI: 10.15514/ISPRAS-2016-28(3)-15

References

[1]. B. Liu, D. Zydek, H. Selvaraj, and L. Gewali. «Accelerating High Performance Computing Applications: Using CPUs, GPUs, Hybrid CPU/GPU, and FPGAs». *In 2012 13th International Conference on Parallel and Distributed Computing, Applications and Technologies*, 2012, pp. 337–342.

[2]. Wim Vanderbauwhede and K. Benkrid. *High-Performance Computing Using FPGAs*. Springer, 2013.

[3]. J. Fowers, G. Brown, P. Cooke, and G. Stitt. «A Performance and Energy Comparison of FPGAs, GPUs, and Multicores for Sliding-window Applications». *In Proceedings of the ACM/SIGDA International Symposium on Field Programmable Gate Arrays*, New York, NY, USA, 2012, pp. 47–56.

[4]. K. Sano, Y. Hatsuda, and S. Yamamoto. «Multi-FPGA Accelerator for Scalable Stencil Computation with Constant Memory Bandwidth». *IEEE Trans. Parallel Distrib. Syst.* 2014, vol. 25, no. 3, pp. 695–705.

[5]. K. Benkrid, A. Akoglu, C. Ling, Y. Song, Y. Liu, and X. Tian. «High Performance Biological Pairwise Sequence Alignment: FPGA Versus GPU Versus Cell BE Versus GPP». *Int. J. Reconfig. Comput.*, vol. 2012, 2012.

[6]. B. G. Fitch, A. Rayshubskiy, M. Eleftheriou, T. J. C. Ward, M. Giampapa, M. C. Pitman, and R. S. Germain. «Blue Matter: Approaching the Limits of Concurrency for Classical Molecular Dynamics». *In Proceedings of the ACM/IEEE SC 2006 Conference*, 2006, pp. 44–44.

[7]. K. J. Bowers, D. E. Chow, H. Xu, R. O. Dror, M. P. Eastwood, B. A. Gregersen, J. L. Klepeis, I. Kolossvary, M. A. Moraes, F. D. Sacerdoti, J. K. Salmon, Y. Shan, and D. E. Shaw, «Scalable Algorithms for Molecular Dynamics Simulations on Commodity Clusters». *In Proceedings of the ACM/IEEE SC 2006 Conference*, 2006, pp. 43–43.

[8]. Y. Komeiji, M. Uebayasi, R. Takata, A. Shimizu, K. Itsukashi, and M. Taiji. «Fast and accurate molecular dynamics simulation of a protein using a special-purpose computer». *J. Comput. Chem.*, vol. 18, no. 12, pp. 1546–1563, 1997.

[9]. D. E. Shaw, J. P. Grossman, J. A. Bank, B. Batson, J. A. Butts, J. C. Chao, M. M. Deneroff, R. O. Dror, A. Even, C. H. Fenton, A. Forte, J. Gagliardo, G. Gill, B. Greskamp, C. R. Ho, D. J. Ierardi, L. Iserovich, J. S. Kuskin, R. H. Larson, T. Layman, L.-S. Lee, A. K. Lerer, C. Li, D. Killebrew, K. M. Mackenzie, S. Y.-H. Mok, M. A. Moraes, R. Mueller, L. J. Nociolo, J. L. Peticolas, T. Quan, D. Ramot, J. K. Salmon, D. P. Scarpazza, U. Ben Schafer, N. Siddique, C. W. Snyder, J. Spengler, P. T. P. Tang, M. Theobald, H. Toma, B. Towles, B. Vitale, S. C. Wang, and C. Young. «Anton 2: Raising the Bar for Performance and Programmability in a Special-purpose Molecular Dynamics Supercomputer». *In Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, Piscataway, NJ, USA, 2014, pp. 41–53.

[10]. M. Taiji, T. Narumi, Y. Ohno, N. Futatsugi, A. Suenaga, N. Takada, and A. Konagaya. «Protein Explorer: A Petaflops Special-Purpose Computer System for Molecular Dynamics Simulations». *In Supercomputing, 2003 ACM/IEEE Conference*, 2003, pp. 15–15.

[11]. C. Rodrigues, D. Hardy, J. Stone, K. Schulten, and W.-Mei Hwu. «GPU acceleration of cutoff pair potentials for molecular modeling applications». *Proceedings of the 5th conference on Computing frontiers*, 2008, pp. 273–282.

[12]. S. R. Alam, P. K. Agarwal, M. C. Smith, J. S. Vetter, and D. Caliga. «Using FPGA Devices to Accelerate Biomolecular Simulations». *Computer*, vol. 40, no. 3, 2007, pp. 66–73.

[13]. N. Azizi, I. Kuon, A. Egier, A. Darabiha, and P. Chow. «Reconfigurable molecular dynamics simulator». *In 12th Annual IEEE Symposium on Field-Programmable Custom Computing Machines, 2004. FCCM 2004*, pp. 197–206.

[14]. Y. Gu, T. VanCourt, and M. C. Herbordt. «Improved Interpolation and System Integration for FPGA-Based Molecular Dynamics Simulations». *In 2006 International Conference on Field Programmable Logic and Applications*, 2006, pp. 1–8.

[15]. V. Kindratenko and D. Pointer. «A case study in porting a production scientific supercomputing application to a reconfigurable computer». 2006, pp. 13–22.

[16]. R. Scrofano, M. B. Gokhale, F. Trouw, and V. K. Prasanna. «Accelerating Molecular Dynamics Simulations with Reconfigurable Computers». *IEEE Trans. Parallel Distrib. Syst.*, vol. 19, no. 6, 2008, pp. 764–778.

[17]. M. Chiu and M. C. Herbordt. «Molecular Dynamics Simulations on High-Performance Reconfigurable Computing Systems». *ACM Trans Reconfigurable Technol Syst*, vol. 3, no. 4, 2010, pp. 23:1–23:37.

[18]. T. Mitchison and M. Kirschner. «Dynamic instability of microtubule growth». *Nature*, vol. 312, no. 5991, 1984, pp. 237–242.

[19]. A. Desai and T. J. Mitchison. «Microtubule Polymerization Dynamics». *Annu. Rev. Cell Dev. Biol.*, vol. 13, no. 1, 1997, pp. 83–117.

[20]. P. Zakharov, N. Gudimchuk, V. Voevodin, A. Tikhonravov, F. I. Ataullakhanov, and E. L. Grishchuk. «Molecular and Mechanical Causes of Microtubule Catastrophe and Aging». *Biophys. J.*, vol. 109, no. 12, , 2015, pp. 2574–2591.

[21]. M. K. Gardner, M. Zanic, C. Gell, V. Bormuth, and J. Howard, «Depolymerizing Kinesins Kip3 and MCAK Shape Cellular Microtubule Architecture by Differential Control of Catastrophe». *Cell*, vol. 147, no. 5, , 2011, pp. 1092–1103.

[22]. D. L. Ermak and J. A. McCammon, «Brownian dynamics with hydrodynamic interactions». *J. Chem. Phys.*, v. 69, issue 4, , 1978, pp. 1352–1360.

[23]. M. Matsumoto and T. Nishimura, «Mersenne twister: a 623-dimensionally equidistributed uniform pseudo-random number generator». ACM Trans. Model. Comput. Simul., vol. 8, 1998, pp. 3–30.

[24]. S. Kasap and K. Benkrid, «Parallel processor design and implementation for molecular dynamics simulations on a FPGA-Based supercomputer». *J. Comput.*, vol. 7, no. 6, 2012, pp. 1312–1328.

[25]. R. Baxter, S. Booth, M. Bull, G. Cawood, J. Perry, M. Parsons, A. Simpson, A. Trew, A. McCormick, G. Smart, R. Smart, A. Cantle, R. Chamberlain, and G. Genest, «Maxwell - a 64 FPGA Supercomputer». *In Second NASA/ESA Conference on Adaptive Hardware and Systems (AHS 2007)*, 2007, pp. 287–294.

[26]. J. M. P. Cardoso, P. C. Diniz, and M. Weinhardt, «Compiling for reconfigurable computing: A survey». *ACM Comput. Surv. CSUR*, vol. 42, no. 4, p. 13, 2010.

[27]. Y. Liang, K. Rupnow, Y. Li, D. Min, M. N. Do, and D. Chen, «High-level synthesis: productivity, performance, and software constraints». *J. Electr. Comput. Eng.*, vol. 2012, 2012, p. 1.

[28]. J. Monson, M. Wirthlin, and B. L. Hutchings, «Implementing high-performance, low-power FPGA-based optical flow accelerators in C». IEEE 24th International Conference on Application-Specific Systems, Architectures and Processors, 2013, pp. 363–369.

[29]. T. Hussain, M. Pericàs, N. Navarro, and E. Ayguadé, «Implementation of a Reverse Time Migration kernel using the HCE High Level Synthesis tool». Field-Programmable Technology (FPT), 2011 International Conference, 2011, pp. 1–8.

Возможности гибридного метода аппроксимации конвективных потоков при моделировании течений сжимаемых сред

М.В. Крапошин, <m.kraposhin@ispras.ru>
Институт системного программирования РАН,
Россия, Москва, ул. Солженицына, д. 25

Аннотация. Для моделирования течений в широком диапазоне чисел Маха предложен гибридный метод аппроксимации конвективных слагаемых, основанный на схеме Курганова-Тадмора и разновидности метода проекций PISO (Pressure Implicit With Splitting Operators). Особенность данного метода состоит в неявной выражении конвективных потоков из схемы Курганова-Тадмора и введении специальной функции-переключателя, обеспечивающей в зависимости от локальных характеристик потока переход от «сжимаемой» схемы (Курганова-Тадмора) к «несжимаемой» схеме (стандартная аппроксимация, используемая в методе PISO). Использование такой гибридной схемы позволяет получить следующие преимущества: а) за счёт неявного учёта диффузионных слагаемых шаг по времени не ограничен скоростью распространения волн диффузионным механизмом; б) за счёт аппроксимации конвективных слагаемых неявным способом и перехода к стандартным схемам PISO шаг по времени ограничивается только потоковым числом Куранта; в) при необходимости разрешения акустических волн достаточно снижения шага по времени до достижения акустическим числом Куранта значений меньше 1 во всей области; г) использование схемы Курганова-Тадмора позволяет получить неосциллирующее решение в задачах с распространением акустических сигналов или при M > 0.3. В данной работе выполнено тестирование реализации гибридного метода для широкого класса задач с известным аналитическим решением и экспериментальными данными: а) сжимаемые течения — распространение волны в прямом канале (Задача Сода), обтекание плоского клина, обтекание обратного уступа сверхзвуковым потоком, обтекание прямого уступа сверхзвуковым потоком, течение в сверхзвуковом сопле при наличии прямого скачка уплотнения в закритической части; б) несжимаемые течения — дозвуковое течение ламинарного вязкого потока в канале круглого сечения, обтекание цилиндра в ламинарном режиме; обтекание цилиндра турбулентным потоком, течение струй газов со смешением; в) промышленные и академические верификационные задачи — истечение струи газа из сверхзвукового сопла, истечение квазиравновесной расширяющейся струи плазмы в вакуум; г) качественное исследование адекватности модели для задач промышленного масштаба — моделирование течения в высокоскоростном компрессоре, модель гидродинамики

водокольцевого насоса. Все материалы работы и исходный код свободно доступны через проект GiHub https://github.com/unicfdlab.

Ключевые слова:математические модели, численное моделирование, численные схемы, сжимаемые течения, акустика, вычислительная гидро- аэро и газодинамика, свободное программное обеспечение.

DOI: 10.15514/ISPRAS-2016-28(3)-16

Для цитирования: Крапошин М.В. Возможности гибридного метода аппроксимации конвективных потоков при моделировании течений сжимаемых сред. Труды ИСП РАН, том 28, вып, 3, 2016, стр. 267-326. DOI: 10.15514/ISPRAS-2016-28(3)-16

1. Список сокращений и обозначений

NASA — National Aero Space Agency

PISO — Pressure Implicit With Splitting Operators

SIMPLE — Semi-Implicit Method for Pressure Linked Equations

U — Скорость

R — Индивидуальная газовая постоянная

C_p — удельная изобарная теплоёмкость

C_v — удельная изохорная теплоёмкость

γ — показатель адиабаты

p — давление среды

T — температура среды

ρ — плотность среды

Y — массовая доля компоненты или фазы в объёме среды

M — число Маха

a — локальная скорость звука

D — коэффициент диффузии

2. Введение

Современный рост вклада численного моделирования в процессы проектирования и эксплуатации инженерных сооружений связан в первую очередь с ростом вычислительных мощностей. В то же время одним из основных факторов, тормозящих применение численных методов в промышленности, является жёсткая привязка последних к выбору математической модели (даже в рамках единого физического представления) и как результат — к конкретной прикладной области. Примером такого разделения численных методов может служить такой параметр сплошной среды как сжимаемость, разбивающая механику жидкости и газа на два больших класса — сжимаемые и несжимаемые течения. Изменение класса

решаемой прикладной задачи приводит к изменению выбора численного метода, программных средства его реализации и набора исходных данных, вплоть до смены технологического процесса решения. Численные методы, предназначенные для решения дозвуковых задач (в первую очередь в смысле несжимаемости) малопригодны для решения транс- и сверхзвуковых задач в силу их особенностей, обусловленных построением соответствующих численных схем. В первом случае чаще всего используются методы проекций (такие как PISO/SIMPLE), а во втором случае — методы на основе приближённого решения задачи Римана. Метод проекций, хорошо зарекомендовавший себя в дозвуковых течениях ($M < 0.3$, [27]), обладает высокой устойчивостью решения, обеспечивающей возможность интегрирования с большим шагом по времени. При этом переход в «сжимаемую область» сопровождается паразитными осцилляциями, которые ставят под сомнение целесообразность использования этого инструмента для случаев с $M > 0.3$ или для разрешения акустических волн. Ещё одним важным достоинством метода проекций является его расширяемость и относительная простота разработки сопряжённых моделей благодаря стандартной процедуре связывания плотности, скорости и давления [28].

Методы, основанные на решении задачи Римана, обеспечивают монотонность и сходимость, но при этом крайне зависят от выбора шага по времени, который определяется скоростью распространения возмущений. Поскольку эта скорость складывается из локальной скорости среды и скорости распространения акустических возмущений, то в случае, когда последняя существенно превышает первую, а интерес представляет исключительно первый механизм движения, шаг по времени становится излишне малым, что делает вычислительные затраты неэффективными и сводит на нет все остальные преимущества этого метода.

При этом существует ряд задач, в которых возникает потребность в применении обоих методов — переходные процессы с периодическим изменением скорости среды от дозвуковой до сверхзвуковой или же моделирование областей, в различных участках которых поток может быть как дозвуковым, так и сверхзвуковым. Примерами таких течений являются течения плазмы, в том числе под действием переменного источника импульса, истечения из резервуара высокого давления в вакуум, двухфазные течения, течения в высокоскоростных компрессорах, ракетных двигателях и пр.

Для решения данной проблемы был предложен и реализован гибридный метод моделирования сжимаемых течений [1]. Впоследствии метод был расширен на случай течений многокомпонетной смеси идеальных газов или двухфазной среды.

Метод реализован в качестве самостоятельного приложения-«решателя» на основе открытой библиотеки OpenFOAM [2].

3. Краткое описание гибридного метода

Основная идея ранее предложенного гибридного метода [1] состоит во введении функции переключателя, которая в зависимости от состояния среды в двух соседних ячейках «смешивает» конвективные потоки, аппроксимированные с использованием метода Курганова-Тадмора, с потоками, вычисленными в соответствии со стандартным методом проекций, адаптированным к сжимаемым течениям. Процедура вычисления потоков гибридным методом включает в себя следующие этапы.

1. Выражение потоков (энергии, импульса, массовых долей) в неявном виде относительно интенсивных переменных в соответствии с процедурой KT/KNP (Kurganov-Tadmor/Kurganov-Noelle-Petrova) и в соответствии с методом проекций.

2. Введение функции-переключателя для «смешивания» конвективных потоков, вычисляемых двумя разными методами.

3. Формирование системы линейных алгебраических уравнений для каждого балансового соотношения и решение этой системы.

4. Формирование алгебраического уравнения для давления на основе дискретизированного уравнения неразрывности. Поиск нового поля давления, удовлетворяющего условию неразрывности, коррекция массовых потоков в соответствии с новым давлением среды.

4. Реализация гибридного метода

Предложенный гибридный метод и его модификации для случаев течения многокомпонентных и двухфазных гомогенных сред был реализован с использованием открытой библиотеки OpenFOAM в виде самостоятельных приложений-«решателей», см. рис. 1. Реализованные приложения находятся в открытом доступе и хранятся в архиве веб-сервиса GitHub по адресу: https://github.com/unicfdlab.

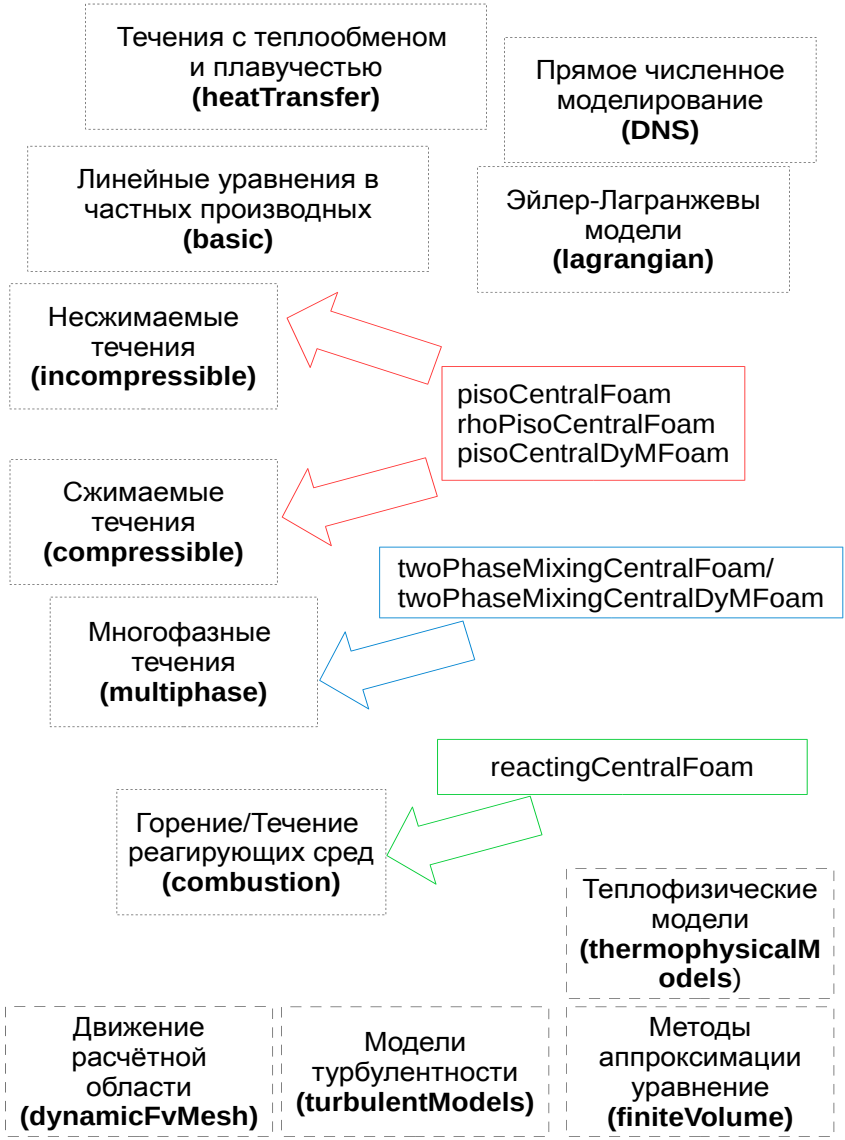


Рис. 1. Приложения-решатели, реализующие гибридный метод и расширяющие стандартный набор моделей пакета OpenFOAM

Fig. 1. Solver applications implementing hybrid method and extending the standard model set of the OpenFOAM package

Разработанные приложения расширяют существующий функционал стандартных моделей OpenFOAM и разделяются на три группы в соответствии с принятым в OpenFOAM способом классификации (рис. 1), полное описание возможностей приложений приведено в табл. 1.

- Сжимаемые и несжимаемые течения — решатели pisoCentralFoam/rhoPisoCentralFoam/pisoCentralDyMFoam.
- Сжимаемые течения реагирующих сред (совершенных газов) — решатель reactingCentralFoam.
- Гомогенная двухфазная модель течения двух сжимаемых сред — twoPhaseMixingCentralFoam/twoPhaseMixingCentralDyMFoam.

В зависимости от набора используемых стандартных библиотек OpenFOAM, меняются дополнительные возможности разработанных приложений. Например, выбор модели турбулентности осуществляется стандартными средствами OpenFOAM и по сути вносит изменения в дискретизацию диффузионных слагаемых в уравнениях изменения импульса и энергии. Отдельные версии приложений, поддерживающие работу с подвижной сеткой (dynamicFvMesh) позволяют осуществлять решение задач в случаях с движущейся расчётной областью.

Табл. 1. Список приложений- «решателей», разработанных на основе библиотеки OpenFOAM и реализующих гибридный метод моделирования сжимаемых течений.

Table 1. List of applications-"solvers" developed on basis of the OpenFOAM library and implementing a hybrid method for modeling compressible flows.

№№	Наименование	Описание
	Сжимаемые и несжимаемые течения однофазные течения	
1.	pisoCentralFoam	Модель ламинарного или турбулентного течения сжимаемой среды (совершенный газ) при числах Маха от 0 до 6 с возможностью переключения между стационарной или нестационарной численной схемой.
2.	rhoPisoCentralFoam	Модель ламинарного или турбулентного течения сжимаемой среды с уравнением состояния реального газа при числах Маха от 0 до 6 и с возможностью переключения между стационарной или нестационарной численной схемой.
3.	pisoCentralDyMFoam	Нестационарная модель ламинарного или турбулентного течения сжимаемой среды (совершенный газ) при числах Маха от 0 до 6 с возможностью моделирования случаев в условиях подвижной расчётной области.
	Сжимаемые течения реагирующих сред (совершенных газов)	
4.	reactingCentralFoam	Модель ламинарного или турбулентного течения сжимаемой многокомпонентной среды (совершенный газ) при числах Маха от 0 до 6 и учётом кинетики химических превращений составляющих потока.
	Двухфазные течения	
5.	twoPhaseMixingCentralFoam	Модель ламинарного или турбулентного течения гомогенной двухфазной сжимаемой смеси.
6.	twoPhaseMixingCentralDyMFoam	Модель ламинарного или турбулентного течения гомогенной двухфазной сжимаемой смеси моделирования случаев в условиях подвижной расчётной области.

5. Рассмотренные задачи

С целью тестирования гибридного метода были рассмотрены следующие группы задач.

1. **Валидационные задачи** для случая сжимаемого течения. В эту группу входят задачи с относительно простой геометрией области течения, имеющие либо аналитическое решение, либо эталонные данные из эксперимента, либо результаты численного моделирования с помощью других численных методов или пакетов. Всего в эту группу входят следующие задачи: а) распространение волны в прямом канале (Задача Сода); б) обтекание плоского клина; в) обтекание обратного уступа сверхзвуковым потоком; г) обтекание прямого уступа сверхзвуковым потоком; д) течение в сверхзвуковом сопле при наличии прямого скачка уплотнения в закритической части.

2. **Валидационные задачи** для случая несжимаемого течения: а) дозвуковое течение ламинарного вязкого потока в канале круглого сечения; б) обтекание цилиндра в ламинарном режиме; в) обтекание цилиндра турбулентным потоком; г) течение струй газов со смешением.

3. **Промышленные верификационные задачи:** а) истечение струи газа из сверхзвукового сопла; б) истечение квазиравновесной расширяющейся струи плазмы в вакуум.

4. **Задачи промышленного масштаба:** а) моделирование течения в высокоскоростном компрессоре; б) моделирование гидродинамики водокольцевого насоса.

6. Результаты тестирования

Проведённое тестирование охватило широкий круг задач — сжимаемые и несжимаемые течения, течения в подвижных областях, многокомпонентные и двухфазные течения.

6.1 Распространение волны в прямом канале (Задача Сода)

Рассматривается случай распространения ударной волны в цилиндрическом канале (ударной трубе). Волна создаётся расширением сжатого воздуха с высоким давлением и температурой, в область, заполненную газом с более низкими давлением и температурой. На рис. 2 представлена схема рассматриваемой задачи. Газ слева и справа от перегородки — воздух при разных температурах и давлениях. Задача является одномерной и имеет аналитическое решение [3]. В начальный момент времени области с разным давлением разделены диафрагмой, в момент разрыва диафрагмы начинается распространение волны сжатия в сторону газа низкого давления, волны

разрежения в сторону газа высокого давления, а также контактного разрыва. В зависимости от соотношения давлений в правой и левой частях ударной трубы перетекание газа может происходить со звуковой и дозвуковой скоростями. При тестировании решателя pisoCentralFoam моделировались оба этих случая (см. табл. 2).

Табл. 2.Начальныеусловиявзадачераспадаразрыва.

Table 2.The initial conditions for the Sod's problem.

Область	Докритическоеистечение		Критическоеистечение	
	Давление, Па	Температура, К	Давление, Па	Температура, К
Слеваотдиафрагмы	P_4=6,897*10^4	T_4=288,89	P_4=6,897*10^4	T_4=288,89
Справа от диафрагмы	P_1=5,897*10^4	T_1=288,89	P_1=6,897*10^3	T_1=231,11

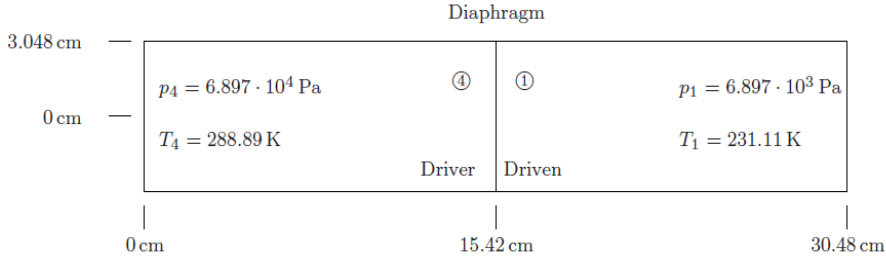


Рис. 2. Схема расчетной области для случая распространения волны в канале

Fig. 2. Computational domain for the case of wave propagation in the channel (Sod's problem)

Тестирование решателя pisoCentralFoam проводилось в одномерной (количество ячеек - 100), двумерной (1000 ячеек), осесимметричной двумерной (1000 ячеек) и полностью трёхмерной постановке (30000 ячеек) с целью определения влияния размерности задачи на результат. На рис. 3 показаны расчётные сетки для двумерного (2а) и трёхмерного (2б) случаев.

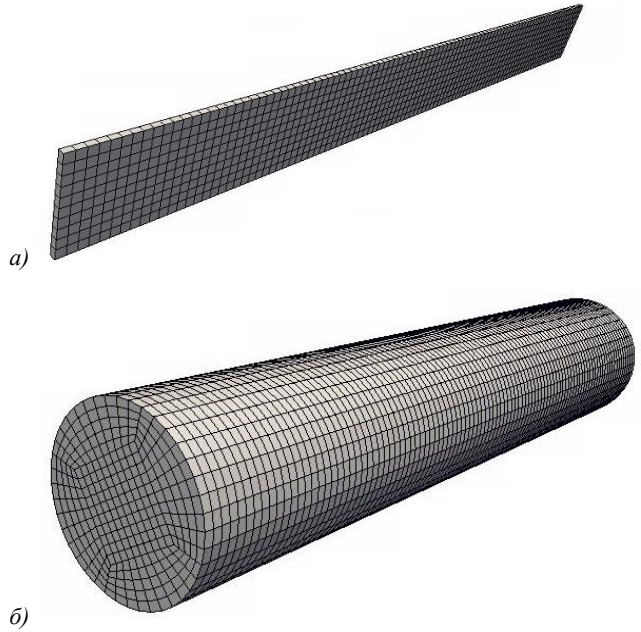


Рис. 3. Варианты расчетной сетки (а — двумерная, б — трехмерная) для задачи Сода

Fig. 3. Types of computational mesh (а — two-dimensional, and б — three-dimensional) used for the Sod problem

Расчёты проводились до момента времени t=0.00025с. Как показали результаты, распределение давления по оси трубы не зависит от размерности задачи. На рис. 4 и 5 приводится сравнение результатов, полученных по одномерному расчёту (т. к. было установлено, что размерность задачи не влияет на результат) с аналитическим решением. Из представленных графиков видно отсутствие осцилляций и хорошая согласованность результатов, полученных с помощью разработанного метода, с аналитическим решением. Вторым важным результатом является совпадение решения, полученного с использованием гибридного метода и исходной схемы Курганова-Тадмора.

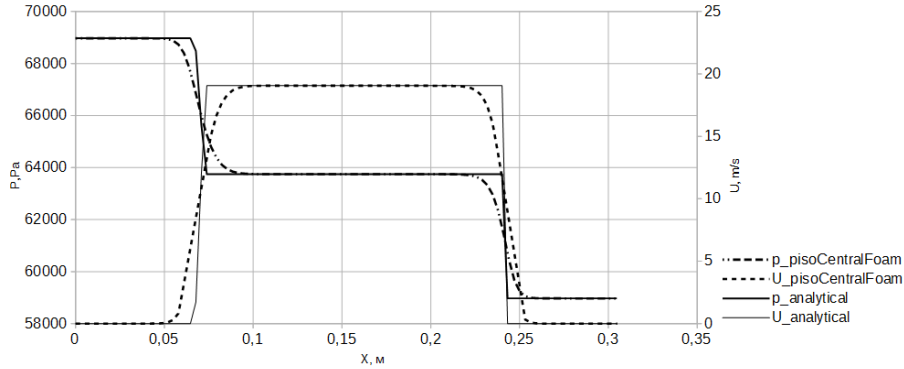


Рис. 4. Сравнение расчётного и аналитического распределения давления и скорости вдоль оси трубы для случая докритического течения. Обозначения: «analytical» - аналитическое решение, «pisoCentralFoam» — численное решение, полученное с помощью реализация гибридного метода в OpenFOAM.

Fig. 4. Comparison of calculated and analytical distributions of pressure and velocity along the tube axis in the case of subcritical flow. Legend: «analytical» - analytical solution, «pisoCentralFoam» - the numerical solution obtained with the implemented hybrid method

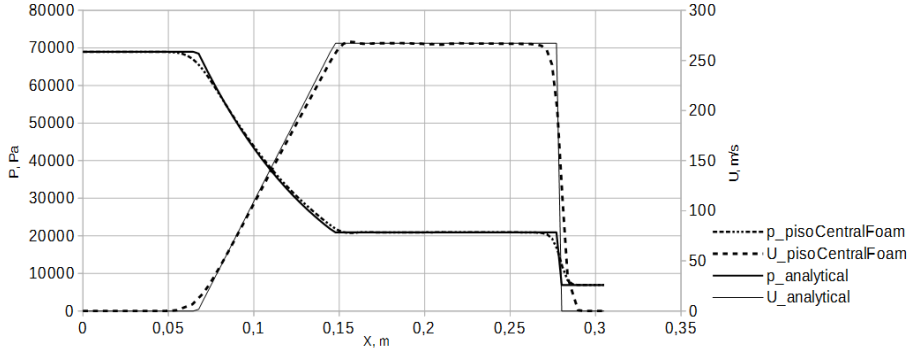


Рис. 5. Сравнение расчётного и аналитического распределения давления и скорости вдоль оси трубы для случая критического течения. Обозначения: «analytical» - аналитическое решение, «pisoCentralFoam» — численное решение полученное с помощью реализация гибридного метода в OpenFOAM.

Fig. 5. Comparison of calculated and analytical distributions of pressure and velocity along the tube axis in the case of critical flow. Legend: «analytical» - analytical solution, «pisoCentralFoam» - numerical solution obtained with the implemented hybrid method

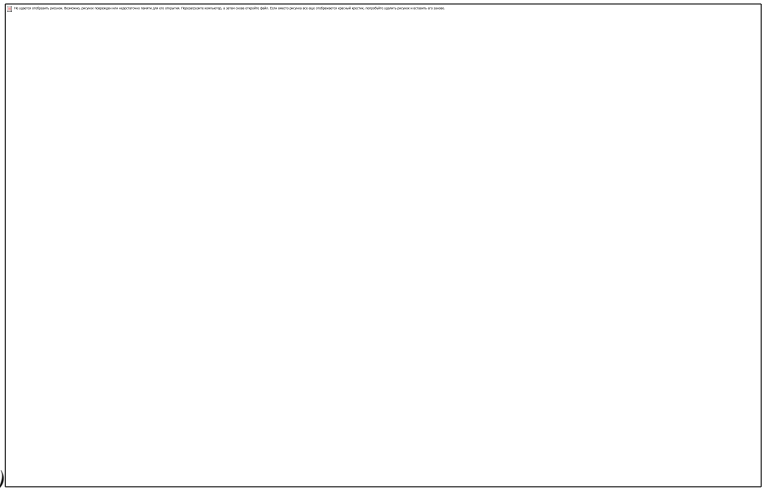
6.2 Обтекание плоского клина

Рассматривалось сверхзвуковое обтекание плоского клина (рис. 6). В качестве исходных данных использовались материалы, представленные в работе [4]. Число Маха набегающего потока M=2.5. Рабочая среда – сухой воздух, молярная масса – 28.96 г/моль, удельная газовая постоянная – 287.05 Дж/кг/К. Принималась гипотеза о возможности моделирования рабочей среды в качестве идеального газа. Давление и температура в набегающем потоке – 101350 Па и 288.9 К.

Изобарная теплоёмкость принималась Ср=1004 Дж/кг/К, показатель адиабаты $\gamma = 1.4$. Скорость звука среды – a = 340.73 м/с. Скорость набегающего потока $\boldsymbol{U} = \boldsymbol{M} \cdot \boldsymbol{a}$ = 851.84 м/с. Динамическая вязкость среды принималась равной 18.4 мкПа-с. Число Прандтля Pr = 1.

Как известно, для данной задачи существует приближенное аналитическое решение в рамках теории косых скачков уплотнений (см. [4]). Таким образом, с помощью данной задачи можно проверить возможности схемы по воспроизведению скачков уплотнений: их положения и «размытости», что позволяет в целом оценить близость численного решения к аналитическому. В качестве параметра, характеризующего близость численного решения к аналитическому, рассматривалось число Маха, скачкообразно меняющееся при прохождении через скачок уплотнения. С этой целью были расставлены точки отбора расчётных значений параметров потока, отстоящие на 0.05 м по координате Y от твёрдой стенки.

а)



б)

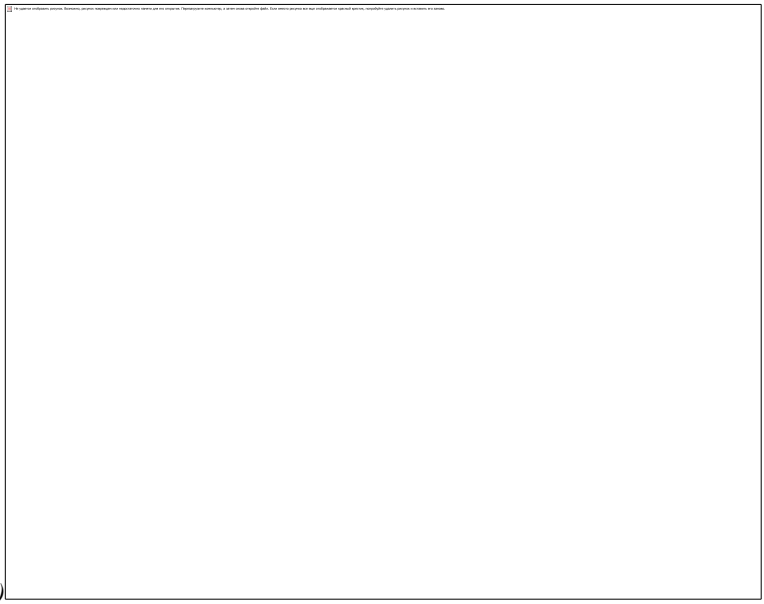


Рис. 6. Расчётная схема для случая набегания потока на клин (а) и сетка в расчётной области для рассматриваемого случая (б)

Fig. 6. Geometry and settings for the case of flow attack on the inclined wedge (a) and the mesh in the computational domain for this case (б)

Для данного случая строилась двухблочная двумерная сетка (рис. 6б). Первый блок представлял собой прямоугольник размерами 0.1522x0.3048 м, второй

(над клином) – прямоугольную трапецию высотой 0.3048 м, равной нижнему основанию, и с боковой стороной, наклонённой к оси OX на 15°. Для проверки сходимости проводилось моделирование для трёх уровней сгущения сетки. Первоначальное разбиение блоков на ячейки: 75x50 ячеек (первый блок – 25x50; второй блок – 50x50). Для получения более грубой и более точной сетки количество ячеек на каждый блок уменьшалось и увеличивалось в 1.5 раза от базового соответственно. Сеточная сходимость проиллюстрирована на рис. 8.

Помимо этого изучалось влияние порядка аппроксимации производной по времени на сходимость решения. Рассматривались неявные схемы первого и второго порядка. Поле давления, показывающее положение скачка уплотнения представлено на рис. 7.

Рис. 7. Поле давления при обтекании косого уступа сверхзвуковым потоком

Fig. 7. The field of pressure at the supersonic flow attack on the inclined wedge

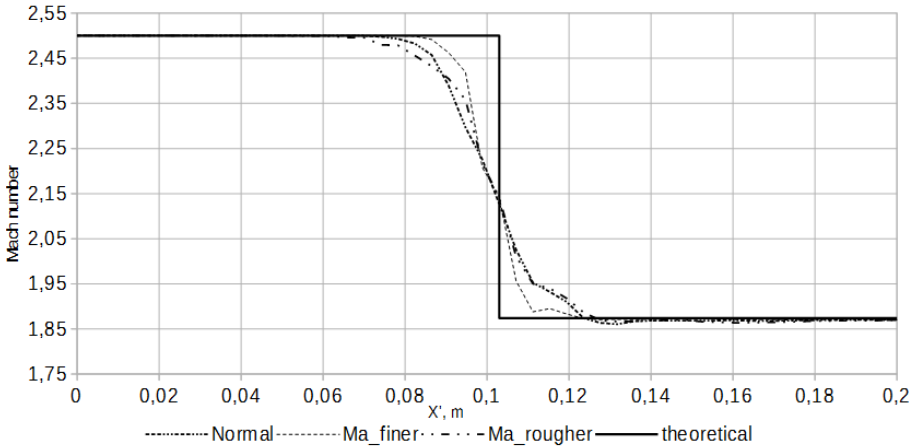


Рис. 8. Сеточная сходимость и сравнение численного и точного решения для случая набегания потока на клин

Fig. 8. Mesh convergence and comparison of numerical and exact solutions for the case of attack of stream inclined wedge

Анализ проведённых расчётов позволяет сделать вывод о том, что использование схемы второго порядка, хоть и приближает численное решение к теоретической зависимости, может приводить к появлению осцилляций. Поведение гибридного метода также оказалось идентичным схеме Курганова-Тадмора.

6.3 Обтекание обратного уступа сверхзвуковым потоком

В качестве третьего тестового примера рассматривалась классическая задача из теории отрывных течений - плоское сверхзвуковое обтекание обратного уступа. На рис. 9 представлена схематичная картина течения. Поток при прохождении кромки уступа расширяется, образуя веер волн разрежения. Наличие преграды в виде горизонтальной поверхности за уступом обуславливает отрыв вязкого потока. Присоединение потока ведёт к образованию λ-образного скачка уплотнения.

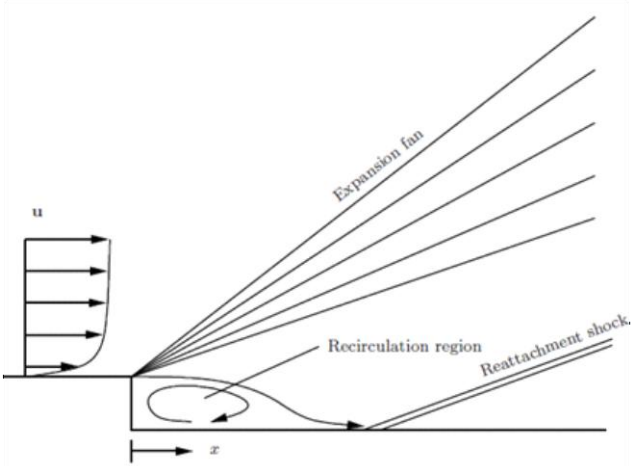


Рис. 9. Схема течения при сверхзвуковом обтекании обратного уступа

Fig. 9. Geometry and seetings for case of supersonic flow over backward step

В качестве исходных данных использовались материалы, представленные в [6]. Число Маха набегающего потока M=2.5. Рабочая среда – сухой воздух, молярная масса – 28.96 г/моль, удельная газовая постоянная – 287.05 Дж/кг/К. Принималась гипотеза о возможности моделирования рабочей среды в качестве совершенного газа. Использовалась стандартная k-ω SST модель турбулентности. Задание параметров турбулентности k, ω на входе расчетной области производилась посредством известных зависимостей: $k=3/2*(\bar{U} I)^2$, где $\bar{U}$ - средняя скорость течения, а I – интенсивность турбулентности, принимаемая равной 5%; $\omega = \varepsilon/(k \cdot C_\mu)$, где кинетическая энергия турбулентности $\varepsilon = C_\mu^{3/4} \cdot k^{3/2}/l$, $l=0,07L$, где L- характерный размер, а коэффициент C_μ принимается равным 0.09.

Статическое давление в набегающем потоке – 13316.6Па, давление торможения — 227527Па, температура набегающего потока — 153.04К, температура торможения – 344.44К.

Изобарная теплоёмкость принималась равной Ср=1005 Дж/кг/К, показатель адиабаты γ=1.4. Скорость звука среды a = 248м/с. Скорость набегающего потока U = M * a = 620м/с. Динамическая вязкость среды принималась равной 18.27мкПа*с. Число Прандтля Pr =0.7.

Высота уступа 0.01125м, расстояние от уступа до входного сечения 0.1016м, до выходного сечения 0.3048м, расстояние до верхней границы расчётной области 0.1475м. Диапазон характерных для течения чисел Рейнольдса: $7*10^3 \div 5*10^6$ [5].

Строилась трёхблочная двумерная сетка. Каждый из блоков представлял собой прямоугольник; первый блок примыкал к уступу своей левой стороной

и насчитывал 240x40 ячеек, второй располагался над уступом (104x112 ячеек), третий располагался над первым блоком и замыкал расчётную область (240x112 ячеек). На всех твёрдых поверхностях задавалось граничное условие прилипания, k и ω аппроксимировались при помощи пристеночных функций.

Результаты моделирования сравниваются с опытными данными по обтеканию обратного уступа той же геометрии, представленными в работе [5], а также с расчётными данными, полученными в кодах PARC,WIND и ANSYSFluidDynamics. Расчёт проводился до 0.06с, что соответствовало установившемуся режиму течения.

На рис. 10 представлены графики распределения давления за уступом в расчёте, в эксперименте и в расчётах других авторов. Давление отнесено к статическому давлению перед уступом, а горизонтальная координата отсчитывается от стенки уступа в дюймах.

Как видно из графика, модель, реализованная в pisoCentralFoam, приводит к результатам, наиболее близким к полученным с помощью ANSYS [6] и в эксперименте, указанном в [6]. В то же время положение кривой, полученной с помощью разработанной модели, несколько отличается от экспериментальных данных из источника [5] и результатов кодов WIND и PARC [7], а предсказываемое давление в отрывной зоне несколько завышено.

Следует отметить, что распределение давления, которое приводится в руководстве пользователя ANSYS [6], в отчёте [5] найдено не было.

Для наглядной оценки адекватности воспроизведения пространственных характеристик среды при сверхзвуковом обтекании со скачками уплотнения выполнено сравнение с имеющимися экспериментальными и расчётными данными. На рис. 11 представлена картина течения, полученная с помощью разработанной модели, реализованной pisoCentralFoam, и сравнение с экспериментальным положением скачка уплотнения [5] и расчётным положением скачка уплотнения, полученным в коде PARC [7]. Как видно из рисунка, положение скачка уплотнения лежит достаточно близко как к экспериментальным, так и к расчётным данным.

Результат проведённого исследования сеточной сходимости представлен на рис. 12. Схема имеет второй порядок точности по пространству и решение сходится к точному; при измельчении сетки в области отрыва по каждому направлению в 4 и более раз погрешность не превышает 1%.

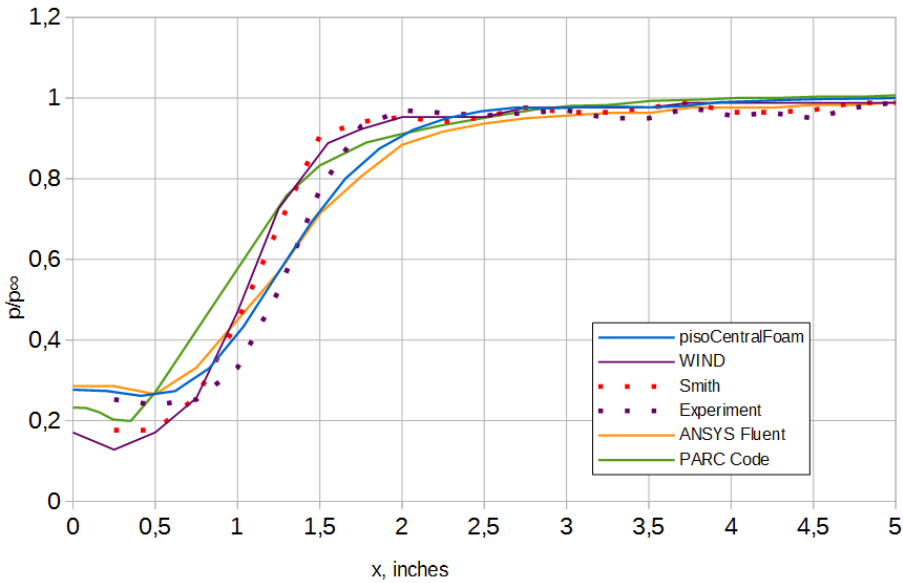


Рис. 10. Сравнение распределения давления за обратным уступом

Fig. 10. Comparison of the pressure distribution behind backward step

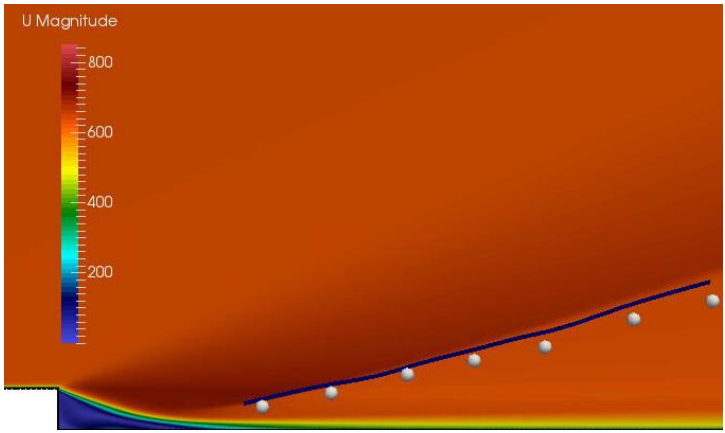


Рис. 11. Сравнение положений скачка уплотнения, вызванного присоединением потока. Серыми точками показано экспериментальное положение скачка уплотнения, синий линией — результат расчёта кодом PARC[7]

Fig. 11. Comparison of the shock position caused by the flow reattachment. Grey dots show the experimental position of the shock wave, the blue line - the result of the calculation with PARC code [7]

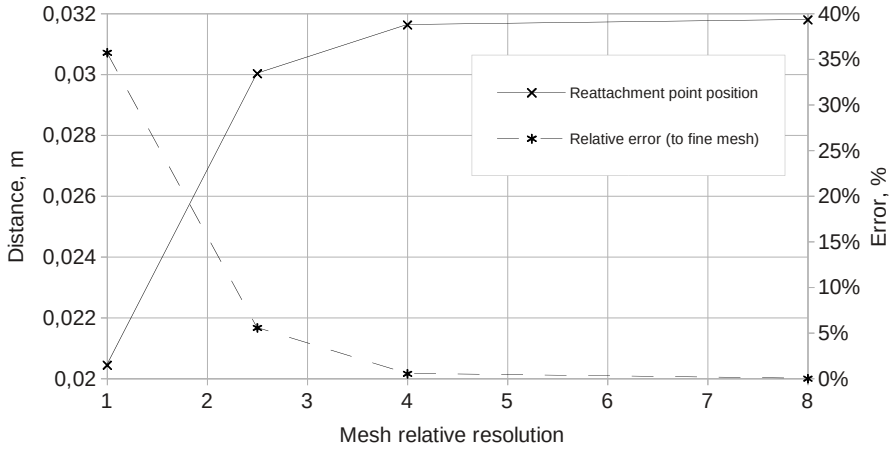


Рис. 12. Сеточная сходимость разработанной численной схемы на примере случая обтекания обратного уступа

Fig. 12. Mesh convergence of the developed numerical scheme by the example of the case of a flow over backward step

6.4 Обтекание прямого уступа сверхзвуковым потоком

Рассматривается сверхзвуковое течение идеального газа в канале с резким сужением (рис. 13). Данная задача является классическим тестом методов моделирования сверхзвуковых течений (см. [8]). В начальный момент времени по всему пространству канала скорость, давление и температура распределены равномерно. Граничные условия следующие.

- На уступе (заштрихован) – условие непротекания для скорости, условие адиабатичности для температуры (нулевая нормальная производная), условие непроницаемости для давления (нулевая нормальная производная).
- На верхней горизонтальной и нижней (вдоль отрезка NX1) горизонтальной границах – условие проскальзывания для скорости (нормальная скорость равна 0, нулевая нормальная производная для тангенциальной), условие адиабатичности для температуры, условие непроницаемости для давления.
- На входе в расчётную область (левая вертикальная граница) – фиксированные значения скорости (3м/с), давления (1 Па), температуры (1 К).
- На выходе из расчётной области (правая вертикальная граница) – нулевые нормальные производные для скорости, давления и температуры.

Физические свойства газа подобраны так, чтобы в начальный момент времени число Маха потока в горизонтальном направлении равнялось 3, а показатель адиабаты (Cp/Cv) – 1.4.

- Молярная масса: 11640.3 г/моль.
- Адиабатная теплоёмкость 2.5 Дж/кг/К.
- Число Прандтля 1.

Параметры разбиения приведены в табл. 3. Расчёт проводился до момента времени t=4c. Сравнивалось положение «ножки» $\boldsymbol{\lambda}$-скачка и наличие/отсутствие неустойчивости Кельвина-Гельмгольца, которая наблюдалась во многих работах при воспроизведении данного случая. При корректной дискретизации в момент времени t=4c положение ножки $\boldsymbol{\lambda}$-скачка должно приходиться на передний край уступа (X=60см).

Данный случай использовался для тестирования масштабируемости модели и исследования сеточной сходимости по времени и пространству. Результаты сеточной сходимости приводятся только для средней, улучшенной и мелкой сеток. Параметры грубой сетки приведены для справки.

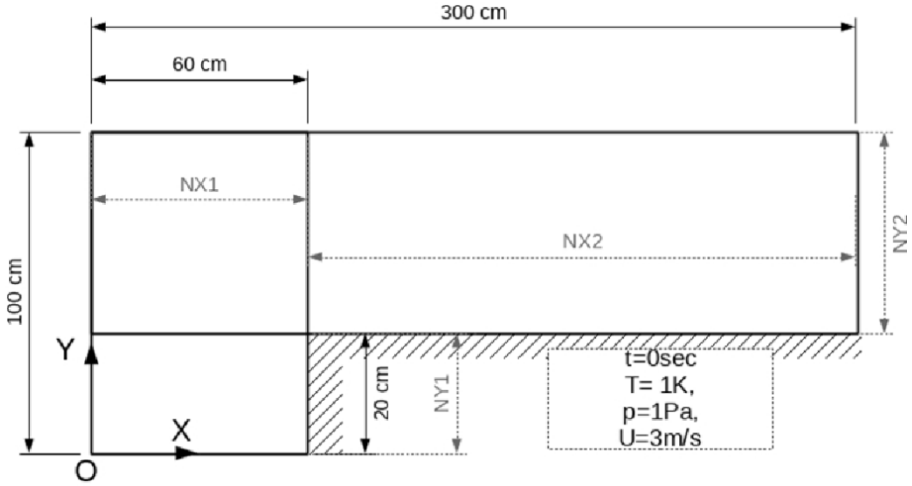


Рис. 13. Схема расчётной области и разбиения на блоки.

Fig. 13. Geometry of computational domain and settings for block mesh for the case of supersonic flow over forward step.

Табл. 3. Параметры разбиения расчётной области

Tab. 3. The parameters of the block mesh of the computational domain

№ п.п.	Отрезок	Число отрезков разбиения			
		грубая сетка	средняя сетка	улучшенная сетка	мелкая сетка
1	NX1	96	192	384	768
2	NX2	384	768	1536	3072
3	NY1	32	64	128	256
4	NY2	128	256	512	1024
		Число ячеек в расчётной сетке			
		60 тыс	250 тыс	1 млн	4 млн

Из расчётов видно, что вне зависимости от степени измельчения расчётной сетки, положение «ножки» $\boldsymbol{\lambda}$— скачка соответствует положению уступа (рис. 14). При этом в случае использования схемы первого порядка по времени, неустойчивость Кельвина-Гельмгольца начинает проявляться только на самой мелкой сетке (4 млн ячеек), в то время как использование схемы второго порядка аппроксимации по времени позволяет воспроизводить этот эффект и на заметно более грубой сетке (1 млн ячеек) — рис. 15. Исследование масштабируемости (рис. 16) решателя показало удовлетворительное ускорение на сетке 1 млн ячеек и сверхлинейное ускорение для сетки 4 млн. ячеек. Сверхлинейное ускорение связано с невозможностью размещения всех данных численной модели (4 млн ячеек) в кэше процессора в однопроцессорном режиме расчета, что привело к завышенному относительному ускорению.

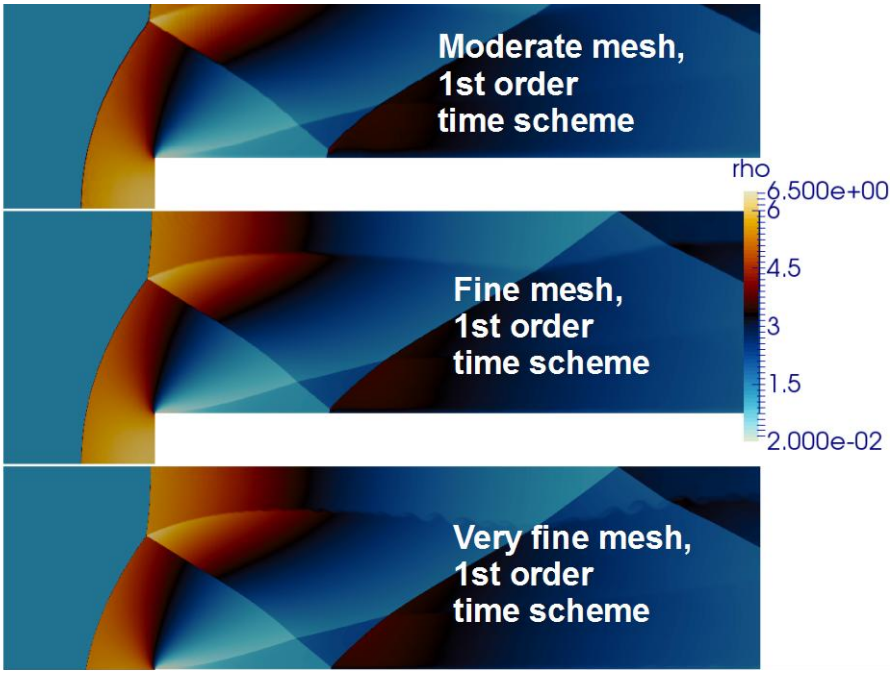


Рис. 14. Сравнение картин течения (поле плотности) для «средней» сетки, улучшенной сетки и мелкой сетки в момент времени t=4с. 1-й порядок аппроксимации по времени и 2-й порядок аппроксимации по пространству.

Fig. 14. A comparison of flow visualisation (field density) for the moderate grid, fine grid and very fine grid at time t = 4s. First order approximation with respect to time and the second order approximation in space were used.

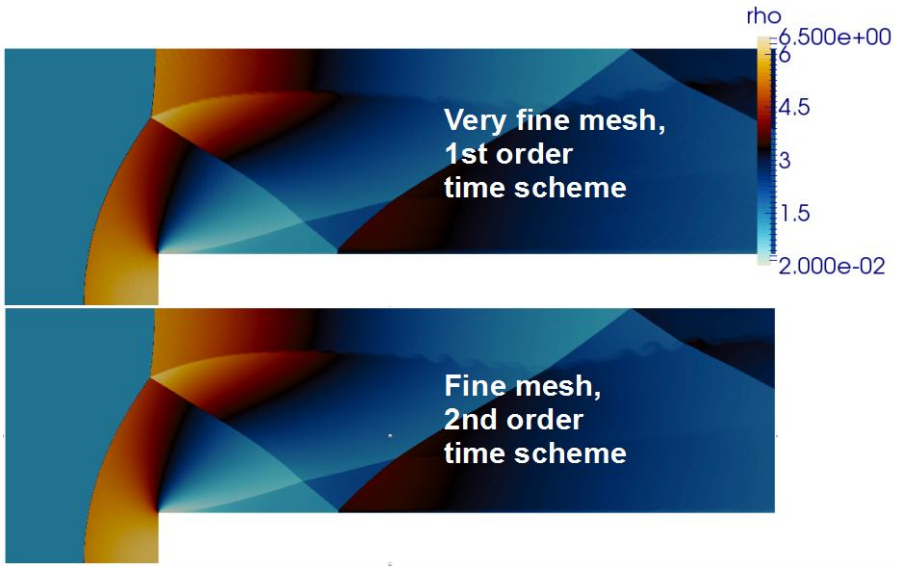


Рис. 15. Сравнение картин течения (плотности) для случая, посчитанного с 1-м порядком аппроксимации по времени на самой мелкой сетке и случая, посчитанного со 2-м порядком аппроксимации по времени на более грубой сетке

Fig. 15. Comparison of the flow visulisation (density) for the case calculated with the 1st order approximation in time on the finest mesh and the case calculated with the 2nd order approximation with respect to time on a coarser mesh

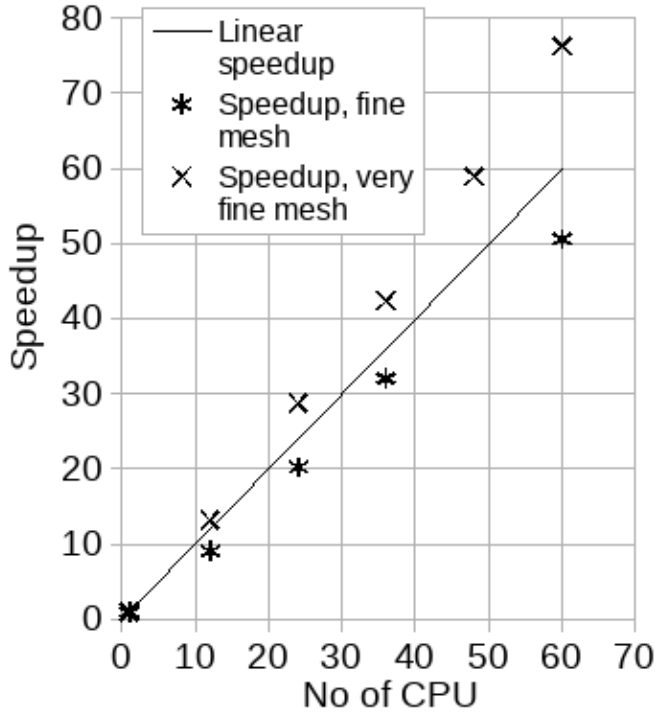


Рис. 16. Сравнение масштабируемости разработанного решателя для сеток 1 млн ячеек и 4 млн ячеек

Fig. 16. Scalability of the developed solver for 1 million cells and 4 million cells grids

6.5 Течение в сверхзвуковом сопле при наличии прямого скачка уплотнения в закритической части

Рассматривалась задача о течении в простейшем одномерном сверхзвуковом сопле, геометрия которого задавалась комбинацией двух усеченных конусов. Начальные данные соответствовали расчетному случаю из [6]. Результаты сравнивались с приближённым аналитическим решением, основанным на законах изоэнтропического течения идеального газа и теории прямых скачков уплотнений [4,6], и с расчетом в ANSYSFluidDynamics.

Схема сопла и структура установившегося течения изображены на рис.17. Отношение площадей на входе и на выходе к критическому сечению принималось равным 3, длина сопла равной 2м (для удобства оперирования с обезразмеренной координатой скачка).

Граничные условия определялись давлениями на входе и на выходе, которые принимались равными 300 и 175 кПа соответственно, на стенках ставилось ГУ проскальзывания. Течение принималось идеальным.

Поскольку существующее аналитическое решение справедливо лишь для одномерного случая (идентичные параметры течения по всему поперечному сечению сопла), была выбрана одномерная расчетная сетка (по одной ячейке в направлениях OY, OZ; ось OX расположена по оси симметрии сопла). Количество ячеек по X - 100.

Таким образом, постановка задачи была максимально приближена к формулировке задачи, соответствующей аналитическому решению.

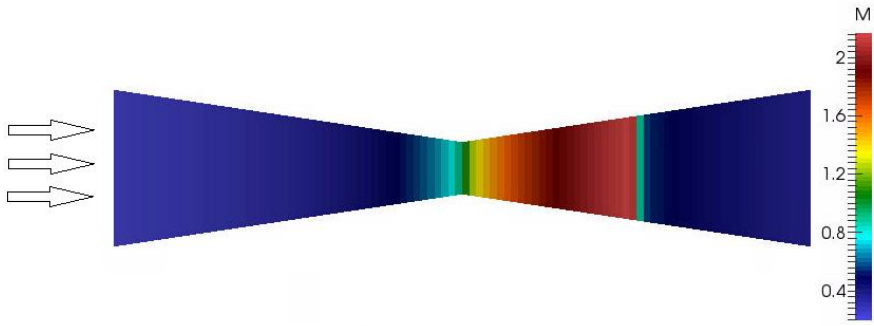


Рис. 17. Картина установившегося течения в сопле

Fig. 17. Visualization of the stationary supersonic flow in the nozzle

Сравнение распределения числа Маха по длине сопла с аналитическим решением представлено на рис. 18. Графики практически полностью совпадают, исключая небольшие расхождений в области скачка уплотнения, которые могут быть объяснены схемной диффузией.

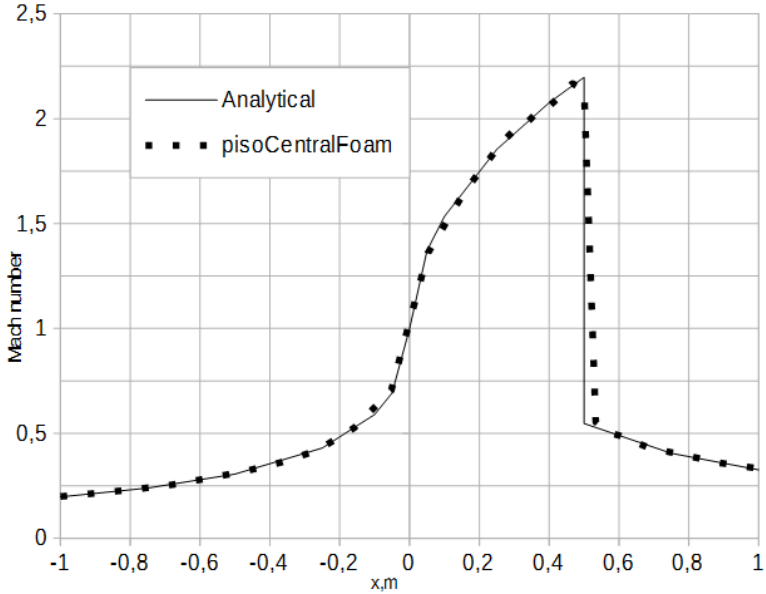


Рис. 18. Сравнение распределения числа Маха по длине сопла.

Fig. 18. Comparison of the distribution of the Mach number along the length of the nozzle.

6.6 Дозвуковое течение ламинарного вязкого потока в канале круглого сечения (течение Пуазейля)

С помощью данного расчётного случая проверяется корректность воспроизведения диффузионных слагаемых в уравнении сохранения импульса (тензора напряжений) при малых числах Маха. Поскольку для данного случая известно аналитическое решение, то можно количественно оценить разницу между точным и приближённым решениями. Для постановки задачи принимается, что число Re = 200, вязкость вычисляется из физических данных, задаваемых при постановке начальных и краевых условий.

Принимается, что граничные условия соответствуют нормальным условиям: скорость на входе $U_{вх}$=0,68369м/с; давление на выходе $p_{вых}$=101325Па; температура — 25 ^{0}C; газ — воздух. Профиль скорости на входе в исследуемую область равномерный.

В соответствии с указанными параметрами среды число Прандтля Pr=0.73, динамическая вязкость $\mu=1.85\cdot10^{-5}$ Па·с, теплоемкость Cp=1007 Дж/кг/К, молярная масса M=28.96г/моль.

Расчётная область представляет собой сектор цилиндрического канала с длиной, существенно большей диаметра канала для получения ламинарного профиля на выходе.

Для получения заданного значения критерия Re, диаметр расчётной области выбирается равным 4.6мм. Длина полагалась равной 161мм. Для получения равномерной сетки расчётная область разбивается на 23 отрезка по радиусу и 1610 отрезков по длине.

Результаты сравнения численного решения, полученного с помощью настоящей модели на выходе из расчётной области, с аналитическим решением (см., например, [4,6]) для ламинарного профиля представлены на рис. 19. Число Маха составляло 0.002. Расчёт вёлся с шагом по времени 30-40 мкс, что соответствует акустическому критерию Куранта порядка 1300.

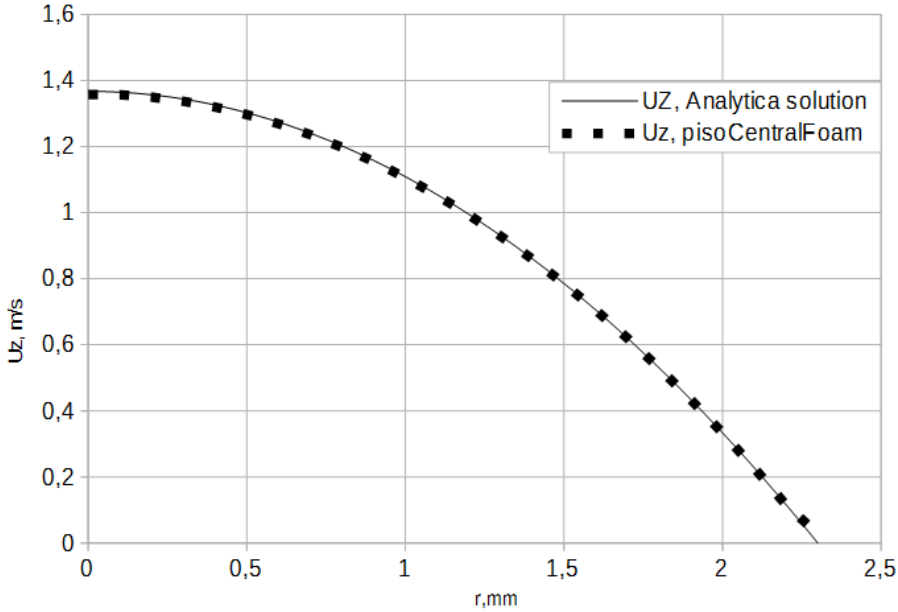


Рис. 19. Сравнение аналитического и расчётного распределений Uz по радиусу канала круглого сечения расчета.

Fig. 19. Comparison of analytical and calculated distributions of Uz along radius of channel with circular cross section

6.7 Обтекание цилиндра в ламинарном режиме

С помощью данного теста исследуется пригодность реализованной модели для моделирования дозвуковых течений для такого широко известно случая, как течение вокруг плохо обтекаемых тел. Исследование проводится для двух случаев — ламинарное течение и турбулентное течение. Исследование последнего случая особенно важно в свете поставленной в работе цели обеспечения использования уже имеющейся в OpenFOAM библиотеки моделей турбулентности без её изменения.

За основу берутся результаты расчётов, полученные в работе [9]. Сравнение проводится для двух значений числа Маха – меньше 0.1 и 0.3, которые соответствуют двум предельно допустимым случаям – “глубокий” дозвук (полностью несжимаемое течение) и сжимаемое течение. Число Re равно 100. В качестве давления и температуры выбираются близкие к нормальным условия – 101325 Па и 300К соответственно, рабочая среда – воздух, молярная масса – 28.9 г/моль.

Таким образом, плотность среды при этих условиях будет равна 1.17404 кг/м3 Изобарная теплоемкость принимается равной 1004 Дж/кг/К, следовательно, показатель адиабаты равен 1.4 Скорость звука среды – $a=\sqrt{(\gamma RT)}$= 347.6 м/с Динамическая вязкость среды взята равной 18.5 мкПа-с Число Прандтля Pr = 0.73. Диаметр цилиндр определяется по заданной скорости на входе и числу Re.

Приняв U = 10 м/с, что соответствует числу Маха 0.029, получаем значение диаметра цилиндра 0.000157м (0.157мм). Приняв U = 100 м/с, что соответствует числу Маха 0.29, получаем значение диаметра цилиндра 0.0157мм.

В качестве критерия проверки правильности результатов расчёта выступал коэффициент сопротивления. В табл. 4 приведено сравнение коэффициентов сопротивления, полученных при помощи гибридного метода с другими численными и экспериментальными исследованиями, приведёнными в [9]. Следует также добавить, что частота срыва вихрей и амплитуда колебаний коэффициента лобового сопротивления цилиндра также находятся в хорошем совпадении с известными данными.

Табл. 4. Сравнение коэффициента сопротивления цилиндра полученных разными методами

Table 4. Comparison of the cylinder drag coefficient obtained by different methods

	pisoCentralFoam	ACL 2008-4	Sharman 05	Mene-01	Kang (2003)	Ding 07
Cd	1.37	1.365	1.33	1.37	1.33	1.356

6.8 Обтекание цилиндра турбулентным потоком

Моделируется обтекание одиночного цилиндра (рис. 20), результаты сопоставляются с исследованием [10]. Помимо сопоставления с экспериментом, было выполнено сравнение с расчетом по несжимаемой и по сжимаемой дозвуковой моделям, имеющимся в OpenFOAM.

Рабочая среда — воздух, условия — близкие к нормальным (давление 101325 Па и температура 300К), плотность — 1.18 кг/м^3, кинематическая вязкость — 1.5*10^{-5} м^2/с, скорость звука — около 330 м/с. Скорость набегающего потока в экспериментах принималась равной 10 м/с, т.е. число Маха было меньше 0.1

— глубоко дозвуковое течение. В ходе экспериментов измерялись значения силы лобового сопротивления F_d и по её значению рассчитывался коэффициент лобового сопротивления C_d .

На первом этапе моделирования для отладки модели расчёты проводились в несжимаемом приближении. Расчёт проводился до наступления установившегося режима течения. Турбулентность учитывалась с использованием модели k-omegaSST [11]. Были рассмотрены варианты расчётных сеток с низким разрешением вблизи поверхности цилиндра (y^+ ~ 100, расчёты проводились с использованием пристеночных функций), так и сеток с высоким разрешением вблизи поверхности цилиндра (y^+ ~1, пристеночные функции не использовались). По полученным результатам для дальнейшего исследования был выбран второй вариант сетки (рис. 21).

На границах расчётной области задавались следующие граничные условия:

1) на входе — значение скорости, интенсивности кинетической энергии и температуры (для сжимаемых случаев);
2) на выходе — значение давления;
3) на поверхности цилиндра — нулевое значение скорости (условие прилипания), пристеночная функция или нулевое значение для кинетической энергии турбулентности;
4) на верхней и нижней границах расчётной области — условие проскальзывания.

Внутри расчётной области в начальный момент температура и скорость задавались равными температуре и скорости на входе, давление — равным давлению на выходе.

При наступлении установившегося режима течения в расчётной области наблюдался периодический отрыв вихрей от поверхности цилиндра и образование за цилиндром вихревой дорожки (т. н. дорожки Кармана).

Вследствие отрыва вихрей значение коэффициента лобового сопротивления C_d колеблется во времени. Поэтому в качестве результатов рассматривалось усреднённое по времени значение C_dна интервале времени после наступления установившегося режима течения.

Моделирование проводилось для случая обтекания цилиндра с диаметром 12.22 см при числе Рейнольдса (Re), равном $7.3*10^4$ (отсюда скорость набегающего потока U= 9м/с). Интенсивность кинетической энергии турбулентности в эксперименте изменялась путем установки на входе решеток с разной формой и размером ячеек и составляла в рассмотренном случае 0.7% (рис. 22). Результирующее значение C_d в эксперименте составило 1.22.

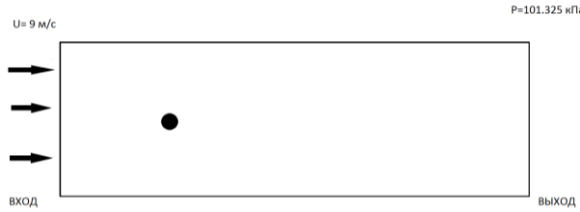


Рис. 20. Расчётнаяобласть

Fig. 20. Geometry and initial settings for the case of flow over cylinder

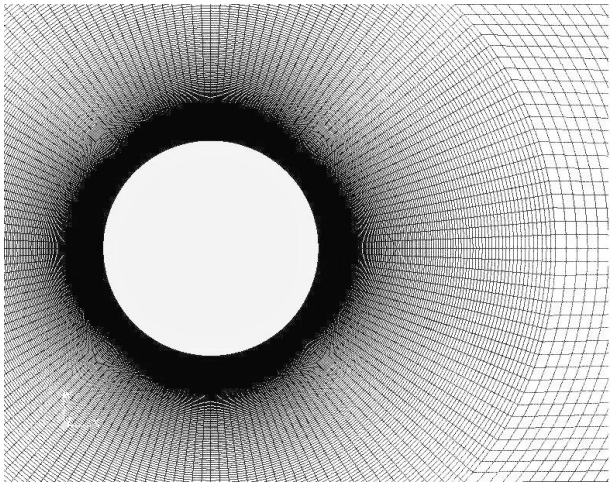


Рис. 21. Расчётнаясетка

Fig. 21. Computational mesh

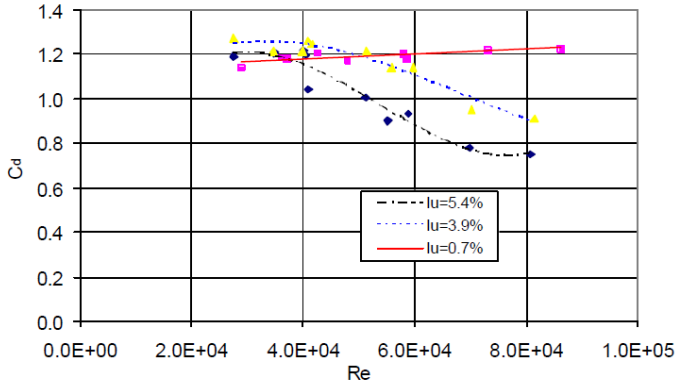


Рис. 22. Экспериментальные данные по обтеканию одиночного цилиндра

Fig. 22. Experimental data (drag coefficient) for the flow around a single cylinder

На втором этапе были проведены расчеты с помощью гибридного метода. Расчеты проводились на сетке с y^+ ~0.8, пристеночные функции не использовались. Результаты сравнения эксперимента, расчёта в несжимаемом солвере pimpleFoam, стандартном сжимаемом решателе rhoPimpleFoam, а также в исследуемом в данной работе решателе pisoCentralFoam приведены в табл. 5.

Табл. 5. Результаты расчётов коэффициента лобового сопротивления цилиндра в турбулентном режиме с помощью различных численных моделей.

Table. 5. The results of calculations of the cylinder's drag coefficient in a turbulent flow through a variety of numerical models.

№	Описание	Используемый солвер	Значение C_d
1	Эксперимент	-	1,22
2	Расчет в несжимаемом решателе (y^+~0.8, I_{inlet}=0%)	pimpleFoam	1,15
3	Расчёт на мелкой сетке в сжимаемом решателе	pisoCentralFoam	1,07
4	Расчёт на мелкой сетке в сжимаемом решателе	rhoPimpleFoam	1,12

6.9 Течение струй газов со смешением

Предложенный в [1] гибридный метод может быть расширен на случай движения многокомпонентной среды совершенных газов. В этом случае к общей системе уравнений из [1] добавляются уравнения переноса (баланса) массы каждой из компонент. С учётом диффузионного приближения [16] уравнение переноса *i*-й компоненты смеси приобретает вид:

$$\frac{\partial \rho Y_i}{\partial t} + \nabla \cdot \left(\vec{U} \rho Y_i \right) = \nabla \cdot (\rho D_i \nabla Y_i)$$

Данный подход был реализован в виде отдельного решателя OpenFOAM и протестирован на простейшей задаче ламинарного смешения двух газов. Для сравнения были взяты результаты расчётов, полученные с помощью схемы AUSM [12].

В данном случае рассматривается смешение двух различных газов текущих в плоском канале (рис. 23). Свойства газа в верхней струе соответствуют азоту (N_2), текущему со скоростью 0.1м/с, нижней струи — водороду (H_2), текущему со скоростью 0.3м/с. Длина расчётной области составляет 1.2м, высота — 0.16м. Давление на выходе — 100кПа, температура обеих струй на входе — 300К. Моделирование производится до достижения стационарного состояния.

Физические свойства сред были взяты следующие:

- Азот N2: молярная масса 28 г/моль, удельная изобарная теплоёмкость 1040 Дж/кг/К, число Прандтля 0.73, динамическая вязкость 17 мкПа-с, коэффициент теплопроводности 0.026 Вт/м/К.
- Водород H2: молярная масса 2 г/моль, удельная изобарная теплоёмкость 15000 Дж/кг/К, число Прандтля 0.73, динамическая вязкость 8.9 мкПа-с, коэффициент теплопроводности 0.172Вт/м/К.

Описание граничных условий и их типы приведены в табл. 6.

Сравнение расчётов с результатами моделирования [12]. Распределения плотности смеси и скорости смеси построены на расстоянии 0.7м от входа в расчётную область и представлены на рис. 24.

Анализ распределения полей плотности и скорости на рис. 24 показывает отсутствие осцилляций при смешении двух компонент движущихся с разной скоростью и близость полученного решения к схеме AUSM. Таким образом, по крайней мере в стационарном приближении схема обеспечивает отсутствие осцилляций для многокомпонентных дозвуковых течений. Перетечки энергии , не обусловленные постановкой задачи также отсутствуют.

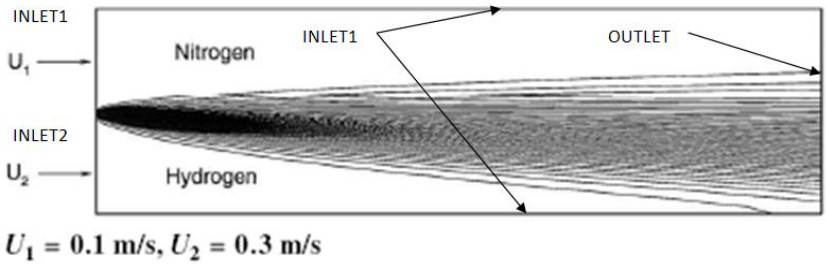


Рис. 23. Схема течения в рассматриваемой задаче о смешении двух газов

Fig. 23. Geometry and settings for the considered problem of flow with mixing of two gases

Таблица 6. Граничные условия

Table 6. Boundary conditions

Поле	Вход азота (Inlet1)	Вход водорода (inlet2)	Выход смеси (outlet)	Стенки (walls)
Давление, Pa	Условие постоянного потока	Условие постоянного потока	Полное давление 100кПа	Условие непроницаемости
Массовая доля, N2	Фиксированное значение, 1	Фиксированное значение, 0	Условие свободного выхода	Условие непроницаемости компоненты
Массовая доля, H2	Фиксированное значение, 0	Фиксированное значение, 1	Условие свободного выхода	Условие непроницаемости компоненты
Скорость среды/компоненты, m/s	Фиксированное значение, 0.1	Фиксированное значение, 0.3	Условие свободного выхода	Условие проскальзывания
Температуры среды/компоненты, K	Фиксированное значение, 300	Фиксированное значение, 300	Условие свободного выхода	Условие адиабатичности

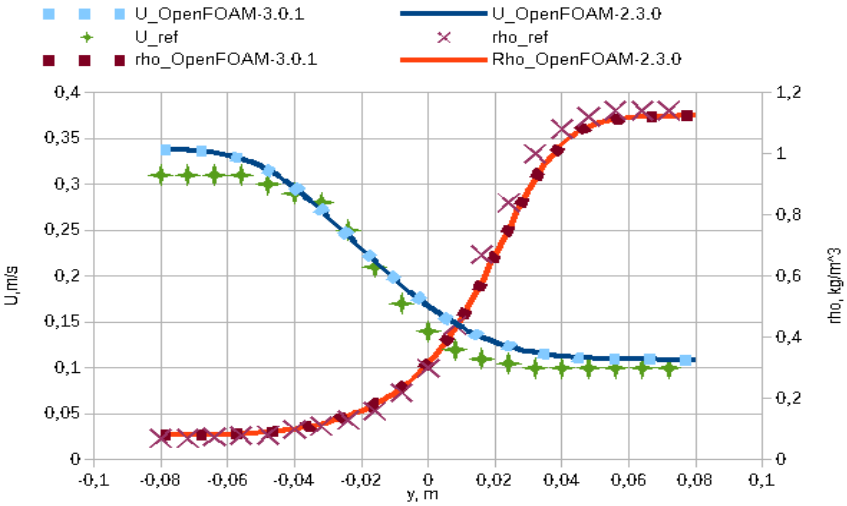


Рис. 24. Сравнение распределения плотности (ось справа) и скорости смеси (ось слева), полученные для задачи смешения струй различными методами и в различных пакетах (U_OpenFOAM-3.0.1 — скорость смеси, гибридный метод, OFv3.0.1, U_OpenFOAM-2.3.0 — скорость смеси, гибридный метод, OFv2.3.0, U_ref — скорость смеси, AUSM, rho_OpenFOAM-3.0.1 — плотность смеси, гибридный метод, OFv3.0.1, rho_OpenFOAM-2.3.0 — плотность смеси, гибридный метод OFv2.3.0, rho_ref — плотность смеси, AUSM)

Fig. 24. Comparison of density distribution (axis on the right) and speed (axis on the left) of mixture for the problem of mixing jets, computed using different methods and packages (U_OpenFOAM-3.0.1 — speed of mixture, hybrid method, OFv3.0.1, U_OpenFOAM-2.3.0 — speed of mixture, hybrid method, OFv2.3.0, U_ref — speed of mixture, AUSM, rho_OpenFOAM-3.0.1 — density of mixture, hybrid method, OFv3.0.1, rho_OpenFOAM-2.3.0 — density of mixture, hybrid method OFv2.3.0, rho_ref — density of mixture, AUSM)

6.10 Моделирование распространения акустических волн

Одним из важных отличий методов, относящихся к классу решения задачи о распаде разрыва (или схожих с ними), от методов из класса проекций является возможность непосредственного учёта распространения акустических волн. Методы второго типа либо «размазывают» решение, теряя информацию, либо создают численные осцилляции.

Тестирование гибридного метода на задачах распространения акустических возмущений позволяет определить степень его пригодности для решения подобных задач, или иными словами - «близость» свойств метода к «исходному» методу Курганова-Тадмора в акустическом диапазоне. Кроме того, тестирование позволит качественно «оценить» дисперсионные и диссипативные свойства схемы [13].

Для тестирования были отобраны два случая с аналитическим решения однородного волнового уравнения, соответствующие монополю и диполю [14].

6.10.1 Моделирование акустических волн, производимых «дышащей» сферой

В первом случае рассматриваются колебания плотности (давления) среды, производимые изменением радиуса сферы по гармоническому закону, рис. 25а . Акустическое давление, возникающее в этом случае вычисляется согласно следующему закону [15]:

$$p(r,t) = \frac{A}{r} e^{-j(\omega t - kr)}$$

где амплитуда вычисляется как $A = \rho_0 c U_0 a e^{-jka}$ при условии совпадения направления излучения с радиус-вектором наблюдателя.

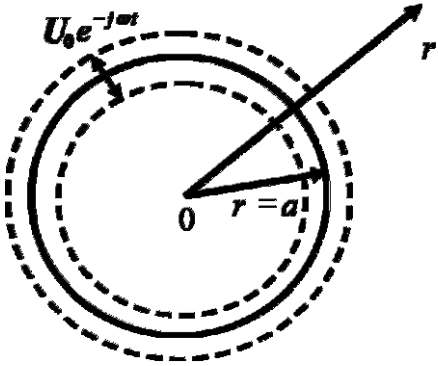


Рис. 25а. Схема излучения акустических волн монополем

Fig. 25а. Diagram of acoustic wave emission by monopole

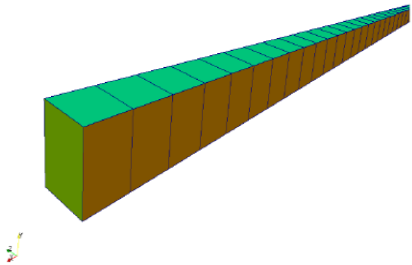


Рис. 25б. Расчётная схема для моделирования монополя.

Fig. 25б.Computational domain of monopole

Для построения расчётной модели, была сформирована область, включающая в себя сегмент поверхности сферы. На «узкой» стороне сегмента задавалась скорость движения поверхности сферы, на противоположной стороне — условие свободного распространения волны. На остальных гранях — условие симметрии, рис. 25б. Далее в расчётной области выбиралась точка в которой сравнивалась динамика пульсации давления с аналитическим выражением.

Константы среды и параметры движения выбирались следующие:

$$U = 5\ \text{m/s};\ \rho_0 = 1.2922\ \text{m/s};\ c = 330.7\ \text{m/s};\ a = 0.1\ \text{m};$$
$$\omega = 1000\ 1/\text{s};$$

Необходимо понимать, что аналитическое решение было получено для распространения акустических волн в стоячей среде (однородное волновое уравнение), в то время как численное решение получено для уравнений Эйлера. Это означает, что между двумя решениями должен присутствовать сдвиг по времени и, значит, для их корректного сравнения требуется совместить оба временны*х* ряда данных. Различие между аналитической и расчётной временными зависимостями обусловлено первым периодом колебаний, на котором численная модель вынуждена преодолевать «начальную инерцию» среды.

Было определено, что диссипативные и дисперсионные свойства схемы становятся пренебрежимо малыми при использовании более чем 20 ячеек на длину волны и акустическом числе Куранта меньшем 1/2. В этих условиях решение стремится к аналитическому, рис. 26.

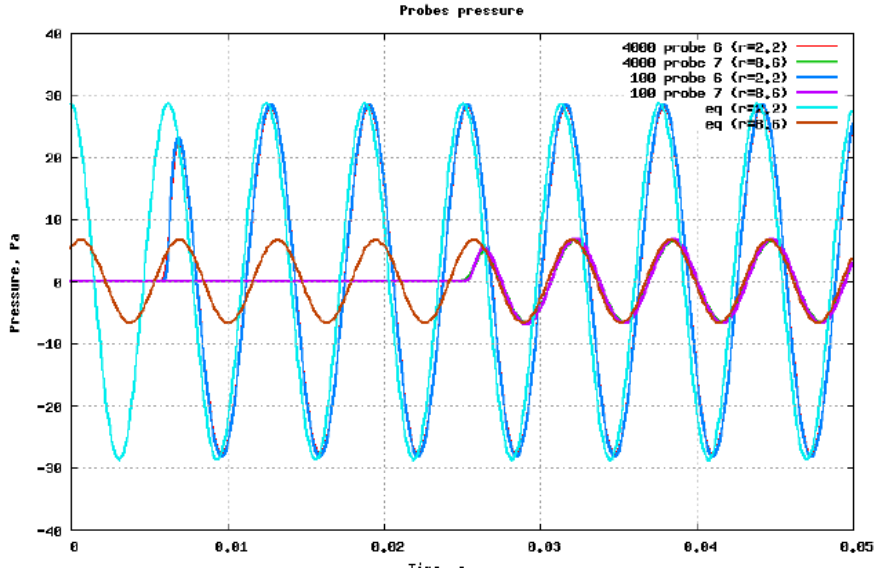


Рис. 26. Сравнение аналитического и численного решения в различных точках и с различным сеточным разрешением для случая пульсирующей сферы.

Fig. 26. Comparison of analytical and numerical solutions at different points and for different mesh resolution for the case of pulsating sphere

6.10.2 Моделирование акустических волн, производимых колеблющейся сферой

Второй тест является расширением предыдущего на трёхмерный случай (рис 27а). Расчётная область представляет собой «дольку» шара из которой вырезан шар малого диаметра.

На поверхности последней задан гармонический закон колебаний вдоль выбранной оси, на внешней поверхности — условие свободного распространения волны. Результаты сравнения представлены на рис. 28, качественная картина распространения волн дана на рис. 27б. Как видно из рисунка, присутствует расхождение по фазе между численным и аналитическим решением, которое как и в первом случае объясняется начальными условиями и соответствующим им начальным возмущениям.

Параметры расчётной сетки (разрешение) были подобраны на основе предыдущего теста. Хорошее совпадение с аналитическим решением при задании соответствующего сеточного разрешения, выбранного на основе выводов из предыдущего раздела, позволяет говорить о принципиальной возможности прогнозирования погрешности численной схемы в зависимости от шага по пространству и времени. Данное свойство является критическим при использовании что в промышленных приложениях.

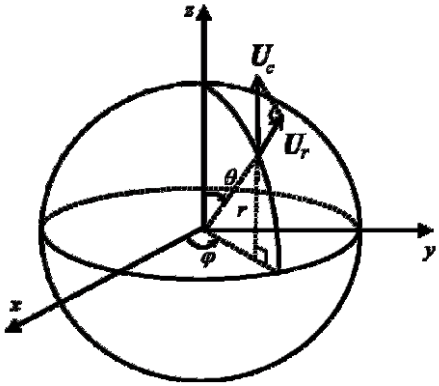


Рис. 27а. Схема излучения акустических волн диполем - «дрожащей» сферой

Fig. 27a.Diagram of acoustic wave emission by dipole («trembling» sphere)

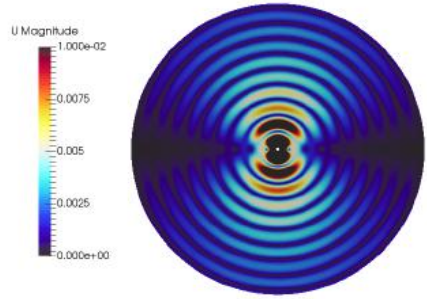


Рис. 27б. Расчётная область и результат для моделирования диполя — дрожащей сферы.

Fig. 27б.Computational domain and result for modelling of the trembling sphere

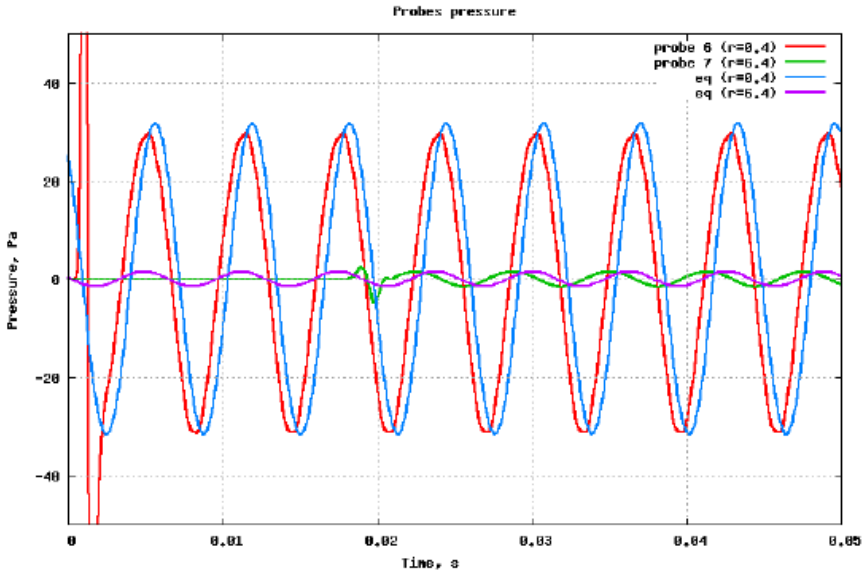


Рис. 28. Сравнение аналитического и численного решения в различных точках для случая дрожащей сферы.

Fig. 28. Comparison of analytical and numerical solutions at different points for the case of trembling sphere

6.11 Истечение струи газа из сверхзвукового сопла

Объектом исследования в данном расчётном случае является плоское двумерное сверхзвуковое сопло [17,18], истечение через которое детально исследовалось как экспериментальным, так и расчётным способом в NASA, рис. 29а и 29б. Коэффициент расширения сопла (отношение площади выходного сечения к критическому) составляет 1.797, проектное отношение давления – 8.78. Эксперименты показали, что при отношении давлений меньше проектного, поток перерасширен и подвержен неустойчивым срывам пограничного слоя, сопровождаемым образованием ударных волн.

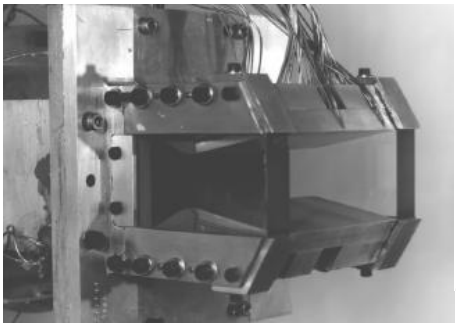


Рисунок 29а. Сопло NASA

Fig. 29a.NASAnozzle

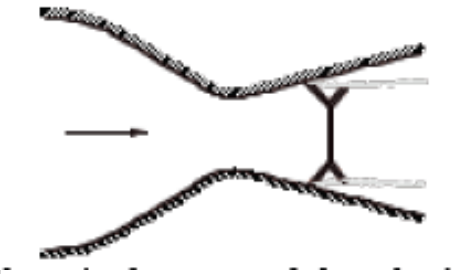


Рисунок 29б. Схематическое изображение перерасширенного сопла с отрывом

Fig. 29b. Schematic representation of over-expanded nozzle with separation

Исследуемой средой является сухой воздух при температуре T=294.44K. Физические свойства следующие – число Прандтля 0.7, динамическая вязкость **μ=**18.27 мкПа-с, удельная изобарная теплоёмкость Ср=1005 Дж/кг/К, молярная масса **28.96кг/кмоль, коэффициент теплопроводности λ=0.024 Вт/м/К.**

Были построены сетки трёх типов:

- Двумерная четырёхугольная сетка, порядка 200 тысяч элементов.
- Двумерная треугольная сетка, порядка 800 тысяч элементов.
- Трёхмерная тетраэдральная сетка, порядка 5500 тысяч элементов.

Расчётная область включала в себя верхнюю часть сопла (рассматривалось симметричное течение) и объём на выходе. Сеточное расширение выбиралось равномерным. Турбулентные свойства течения воспроизводились с помощью **k-ω-SST [11] модели и пристеночных функций,** величина y+ не превышала 3. В качестве граничных условий задавались полное давление и температура на входе в расчётную область и полное давление на выходе. Для моделирования выбран режим с отношением давлений 5.02.

Для учёта смены режима на выходной границе применялось смешанное граничное условие – если число Маха в прилегающих ячейках было равно или превышало 1, то задавалось условие свободного выхода, иначе – полное давление среды на бесконечности. Расположение и обозначение граничных условий показано на рис. 30, описание граничных условий дано в табл. 7.

Качественная картина течения показана на рис. 31 для двух различных типов сетки (четырёхугольная и треугольная). Сравнение расчётных данных [17] с гибридным методом показано на рис. 32, сравнение расчётных и экспериментальных данных ([18]) показано на рис. 33.

Сопоставление экспериментальных и расчётных данных показывает их хорошую согласованность для сеток трёх разных типов, что позволяет говорить о наличии сеточной сходимости. Положительным аспектом является «схожесть» картин течения, полученных на четырёхугольной и треугольной сетках с примерно одинаковой длиной рёбер, позволяя надеяться на независимость получаемого результата от топологии.

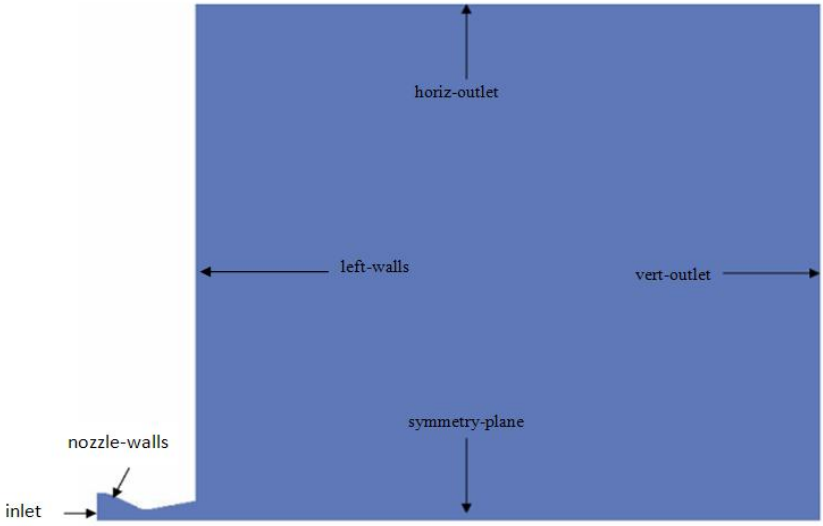


Рис. 30. Схема расположения граничных условий и их наименования в модели истечения сверхзвуковой струи в сопле NASA

Fig. 30. Geometry, settings and boundary conditions in the model of flow expansion in a supersonic jet in a NASA nozzle

Таблица 7. Список граничных условий в задаче истечения газа из сверхзвукового сопла

Table 7. List of boundary conditions for the problem of expansion of gas from a supersonic nozzle

Граница/ Поле	P, Pa	U, m/s	T, K	k	omega
inlet	Полное давление, 101325Па	Условие свободного входа	294.44	Интенсивность турбулентности	Турбулентная длина смешения
nozzle-walls	Условие непроницаемости	Условие прилипания	Условие адиабатичности	Пристеночная функция	Пристеночная функция
symmetry-plane	Условие симметрии				
vert-outlet	Условие свободного выхода или полное давление	Условие свободного выхода	Условие свободного выхода	Условие свободного выхода	Условие свободного выхода
horiz-outlet	Условие свободного выхода или полное давление	Условие свободного выхода	Условие свободного выхода	Условие свободного выхода	Условие свободного выхода
left-walls	Условие непроницаемости	Условие прилипания	Условие адиабатичности	Пристеночная функция	Пристеночная функция

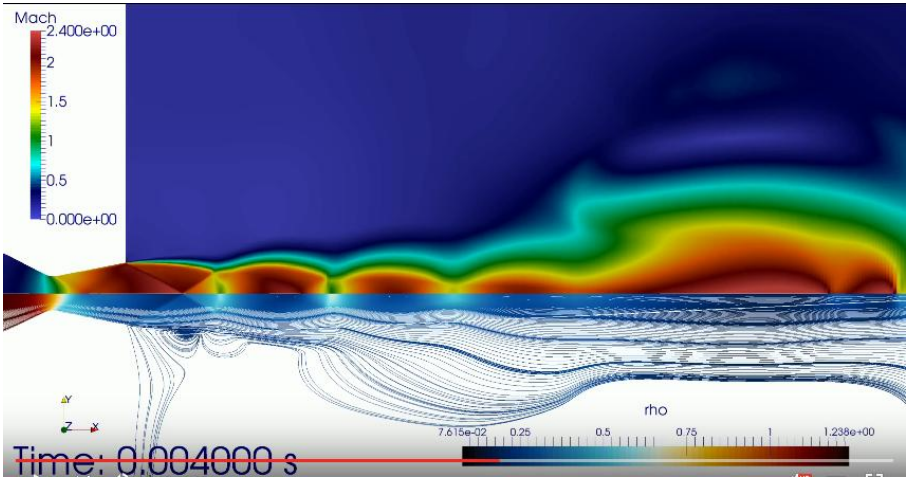


Рис. 31а. Мгновенное поле числа Маха и линии тока, раскрашенные по полю плотности при истечении струи газа из сопла NASA, гексаэдральная сетка

Fig. 31a. Instantaneous field of Mach number and stream lines colored according to field of density during jet expansion from a NASA nozzle, hexahedral mesh

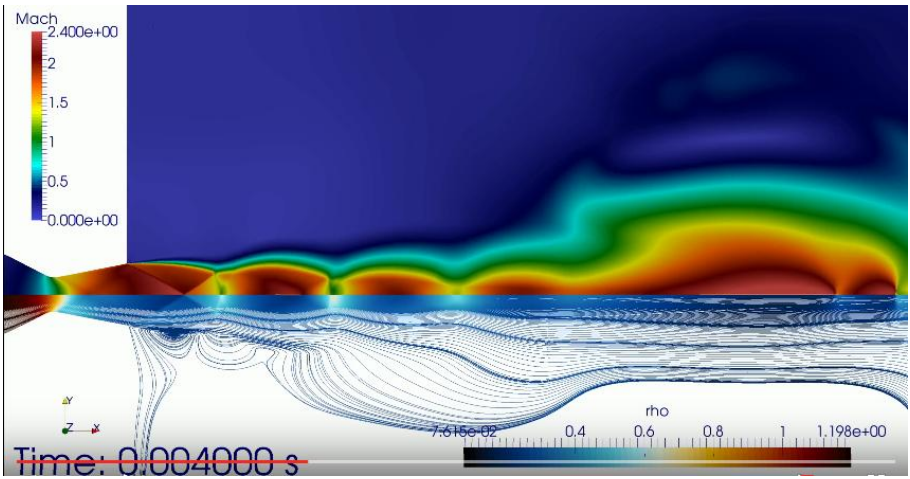


Рис. 31б. Мгновенное поле числа Маха и линии тока, раскрашенные по полю плотности при истечении струи газа из сопла NASA,тетраэдральная сетка

Fig. 31б. Instantaneous field of Mach number and stream lines colored according to field of density during jet expansion from a NASA nozzle, tetrahedral mesh

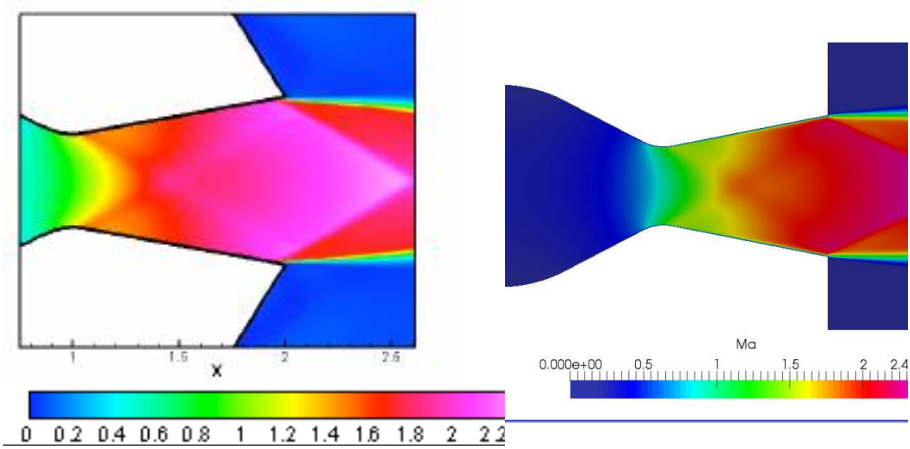


Рис. 32. Сравнение визуализации числа Маха в сопле, полученной численным методом в [17] (слева), и в настоящей работе (справа)

Fig. 32. Comparison of visualizations of Mach number in a nozzle, calculated by the numerical method in [17] (left) and in this paper (right)

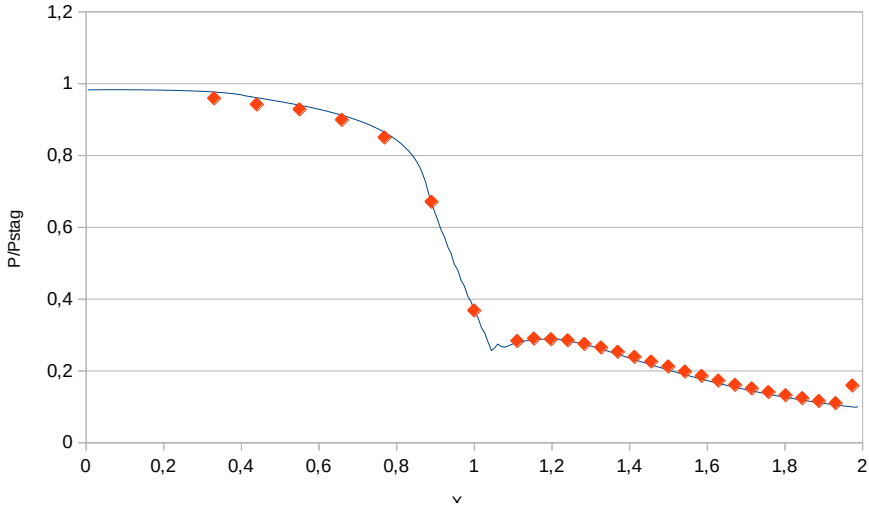


Рисунок 33. Сравнение экспериментального распределения давления по стене сопла (оранжевые ромбы) и расчётного (синяя линяя).

Fig 33. Comparison of experimental (orange diamonds) and calculated (blue line) distribution of pressure on the nozzle wall

6.12 Истечение квазиравновесной расширяющейся струи плазмы в вакуум

Одним из важных приложений, в которых гибридный метод может оказаться эффективным, является моделирование механики плазмы. Этот тип течений характеризуется большими градиентами плотности, скорости, давления и температуры как в пространстве, так и во времени (пульсации). Впрочем, даже в случае стационарных задач эти течения характеризуются экстремально широкими диапазонами изменения термодинамических параметров — в сотни и тысячи раз, что несомненно является «вызовом» для численных методов.

Для тестирования гибридного метода был взяты результаты эксперимента и соответствующих расчётов [19,20], выполненных средствами коммерческого пакета Fluent. Данный эксперимент проводился в университете г. Эйндховен и позволяет оценить применимость приближения сплошной среды для указанного класса задач.

Опуская подробности формирования высокотемпературной плазмы, расчётная область представляет собой цилиндрический канал малого сечения, который расширяется в цилиндрический канал широкого сечения, заканчивающийся цилиндрической щелью, так что в разрезе стенки широкой части образуют «уступ» (рис. 34).

В канал малого сечения поступает горячая плазма Ar при M = 1, которая, переходя в канал большого сечения, расширяется до давления 20-100Па и, огибая уступ, проходит к выходу. Для сравнения с расчётом из эксперимента доступны данные по распределению скоростей и температуры на оси канала. Диаметр последнего участка перед расширением составляет 6мм, его длина - 10 мм, из которых последние 5 мм приходятся на конический раструб с углом раствора 45^0, так что диаметр канала перед областью свободного течения - 16мм Согласно постановке задачи [19] известны температура газа плазмы (9283К, или 0.8эВ), число Маха (M=1) и расход — 3 SLM. Исходя из этих параметров вычисляется скорость потока на входе и давление.

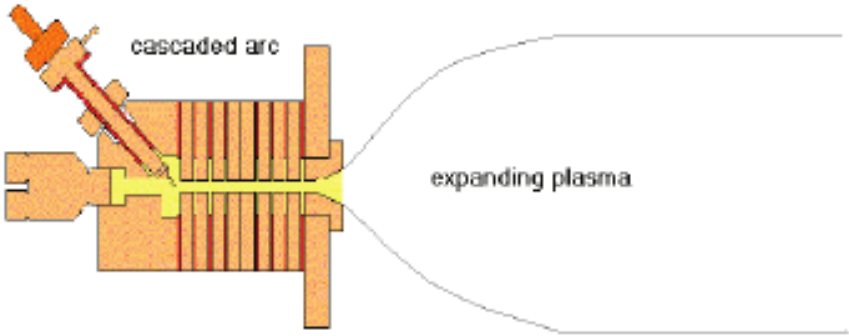


Рис. 34. Схематическое описание установки для генерации плазмы.

Fig. 34. Schematic representation of the plasma generation device

Качественная картина течения приведена на рис. 35. Сравнение экспериментальных и расчётных картин течения показано на рисунке 36. Данная задача решалась в стационарной постановке, что позволило сэкономить время вычислений в несколько раз даже для такого относительно сложного случая с изменением температуры и давления в 10 и 100 раз. Результаты расчётов показывают в первую очередь хорошее совпадение с математической моделью, заложенной в коммерческий пакет Fluent. При этом расхождение с экспериментальными данными можно объяснить не только некорректностью выбора предположения о сплошности среды, но и граничными условиями, неопределённость которых связана с погрешностью экспериментальных данных. Последняя же составляет порядка 10% по заявлению самих авторов эксперимента.

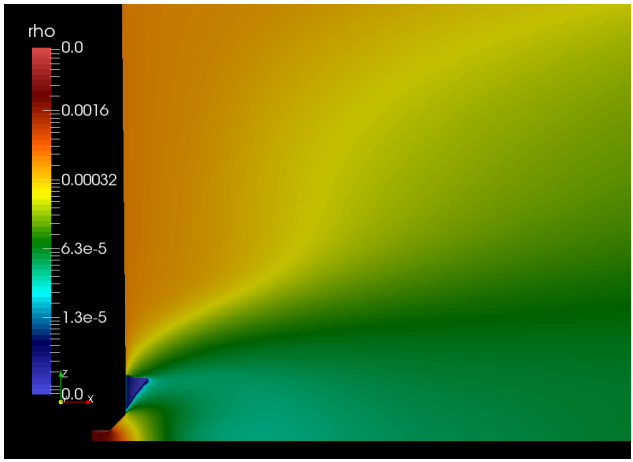


Рис. 35. Поле плотности (в логарифмическом масштабе) плазмы Ar при стационарном расширении в вакуум 20Па

Fig. 35. Field of density (in log scale) of Ar plasma during stationary expansion into 20Pa vacuum

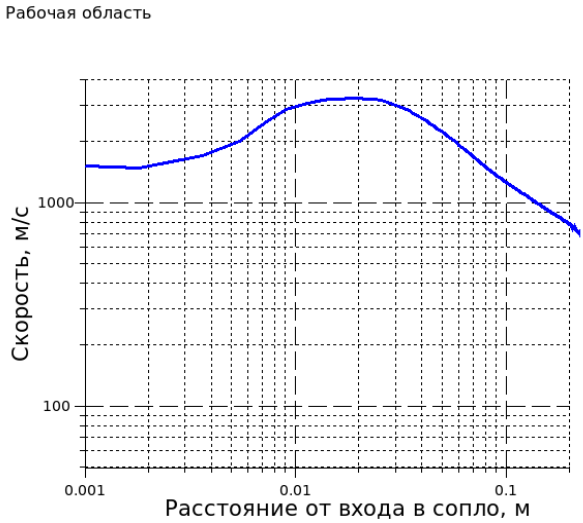


Рис. 36а. Распределение осевой скорости плазмы, полученное расчётным способом с помощью гибридного метода. Построеновлогарифмическоммасштабе.

Fig. 36a. Distribution of axial speed of plasma, computed using hybrid method. Plotted in log scale.

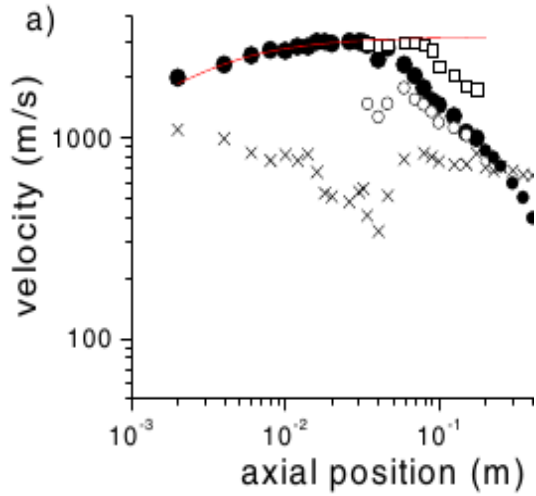


Рис. 36б. Распределение осевой скорости плазмы, полученное экспериментальным способом в работе [19]. Построеновлогарифмическоммасштабе.

Fig. 36б. Distribution of axial speed of plasma, measured in experiment in [19]. Plotted in log scale.

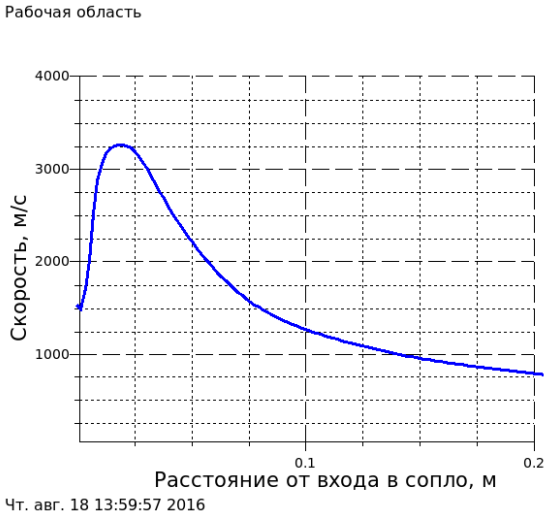


Рис. 36в. Распределение осевой скорости плазмы, полученное расчётным способом с помощью гибридного метода. Построеновлинейноммасштабе.

Fig. 36в. Distribution of axial speed of plasma, computed using hybrid method. Plotted in linear scale.

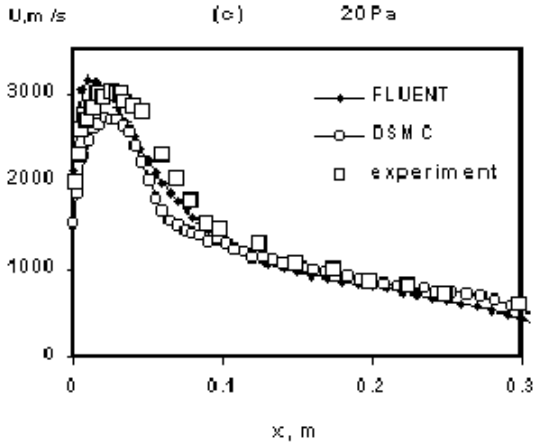


Рис. 36г. Сравнение распределений осевой скорости плазмы, полученных экспериментальным способом в работе [19] и расчётно-теоретическими средствами в работе [20]. Построеновлинейноммасштабе

Fig. 36г. Comparison of axial plasma velocity between experiment [19] and numerical simulation [20]. Plottedinlinearscale.

6.13 Моделирование течения в высокоскоростном компрессоре

Гибридный метод был также доработан для моделирования устройств с подвижными частями. Использование стандартного алгоритма PISO позволило выполнить процедуру интегрирования уравнений на подвижной сетке аналогично имеющимся в OpenFOAM моделям[21]. Реализованный метод был валидирован на стандартном тесте ERCOFTAC, см. [22, 23]. На рис. 37, представлены результаты сравнения разработанного метода, стандартной модели OpenFOAM и эксперимента для обезразмеренного давления в радиальном зазоре компрессора (рис. 37б).

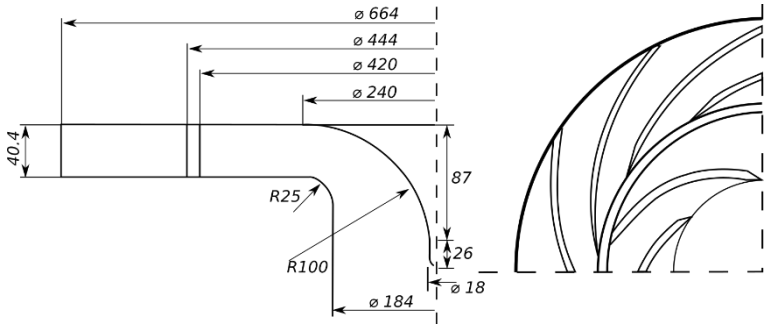


Рис. 37а. Схема компрессора ERCOFTAC

Fig. 37a. Diagram of an ERCOFTAC compressor

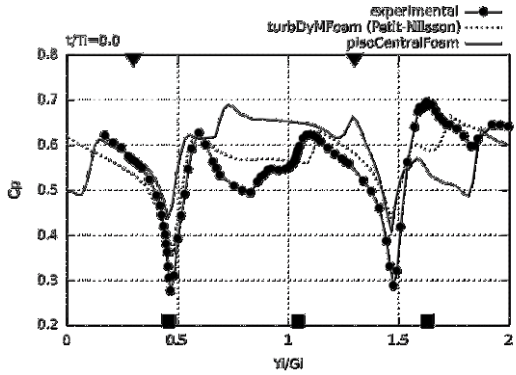


Рис. 37б. Сравнение профилей давления в зазоре компрессора ERCOFTAC, полученных расчётным и экспериментальным способом

Fig. 37b. Comparison of computed and experimental pressure profile in an ERCOFTAC compressor opening

Далее с использованием гибридного метода и алгоритма POD были получены и исследованы характерные моды и частоты потока в типичном высокоскоростном микрокомпрессоре [24], конструкция которого показана на рис. 38. В настоящее время статистические методы анализа, схожие с POD, являются аналогами модальных методов анализа конструкций, применяемых для выявления собственных форм и частот колебаний конструкций. Совместное использование таких статистических методов анализа для конструкций и проточных частей позволит избежать возникновения резонансных явлений и связанных с ними разрушений.

С другой стороны, эти методы могут использоваться для тестирования численных схем — если для заданного течения заранее известны моды и соответствующие им частоты, то их отсутствие может вполне сигнализировать о недостаточном качестве численной схемы.

Например, анализ мод течения в модели компрессора показал наличие характерных частот, соответствующих лопаточным частотам ротора и статора — моды №0, №5 и №7 или оборотным частотам, моды №1 и №3 (рис. 39). Визуализация соответствующих мод показана на рис. 40 для мод №0 и №3. Статистический анализ показывает, что даже на относительно грубой сетке основные частоты, характерные для данного типа машин, — оборотная, лопаточная импеллера и лопаточная диффузора могут быть разрешены с использованием гибридного метода.

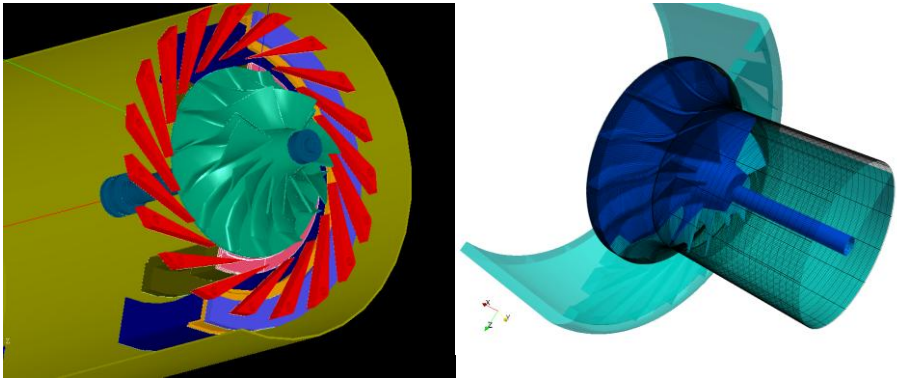


Рис. 38. Схематическое изображение проточной части рассматриваемого компрессора (слева) и построенной блочно-структурированной сетки (справа).

Fig. 38. Schematic presentation of flow channel in the considered compressor (left) and consequent block-structured mesh (right)

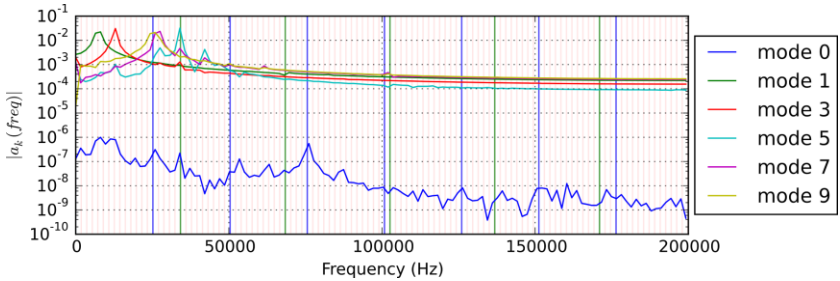


Рис. 39. Характерные моды течения в микрокомпрессоре. Синими вертикальными линиями отмечены частоты относящиеся к ротору, зелёными — относящиеся к статору. Частотавращения 108000об/мин

Fig. 39. Characteristic modes of flow in a microcompressor. Frequencies related to rotor are marked with blue vertical lines, frequencies related to stator with green vertical lines. Rotation frequency is 108000rpm.

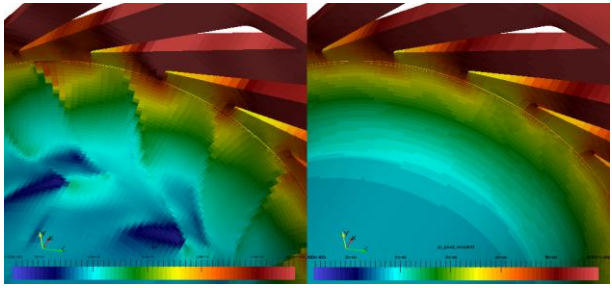


Рис. 40а. Мгновенное (слева) поле давления в компрессоре и его 0-я мода (справа) в области импеллера

Fig. 40a. Instantaneous field of pressure in the compressor (left) and its zeroth mode (right) in the impeller area

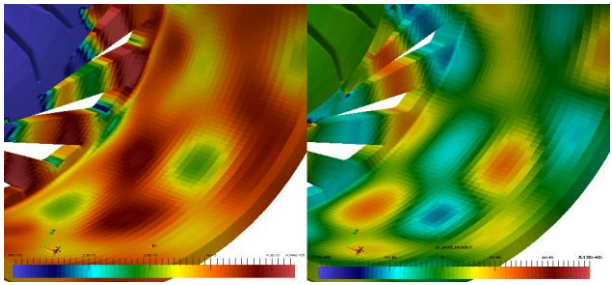


Рис. 40б. Мгновенное (слева) поле давления в компрессоре и его 3-я мода (справа) в области диффузора

Fig. 40b. Instantaneous field of pressure in the compressor (left) and its third mode (right) in the diffusor area

Тестирование модели на задаче ERCOFTAC позволяет говорить о качественно удовлетворительном совпадении результатов расчётов и эксперимента. При этом наблюдающиеся различия могут быть связаны с рядом причин, которые требуют рассмотрения в отдельном исследовании - «трёхмерность» течения, влияние «модельного» числа Маха на расчёт, выбор модели турбулентности и пр.

Также наглядно продемонстрирована возможность использования методов численного моделирования газодинамики высокоскоростных потоков и методики POD для обработки больших данных, что позволяет говорить о возможности внедрения последней как инструмента для модального анализа в вычислительной механике жидкостей.

6.14 Модель гидродинамики водокольцевого насоса

Ещё одним важным направлением, заслуживающим интереса, является возможность применения гибридного метода для моделирования двухфазных течений в гомогенном сжимаемом приближении. Такие модели могут быть полезными для первичной оценки интегральных характеристик устройств со смешением потоков сред с большим отношением плотностей (например, вода и воздух). Кроме того, сжимаемые модели позволяют оценить пульсации давления и следовательно, уровень шума, что также является актуальной инженерной задачей в настоящее время. В случае течения гомогенной двухфазной смеси система уравнений баланса массы, импульса и энергии смеси дополняется уравнением переноса массы одной из компонент, например воды (Liq), выраженной в массовой доли этой фазы Y_{Liq}:

$$\frac{\partial \rho Y_{Liq}}{\partial t} + \nabla \cdot \left(\vec{U} \rho Y_{Liq} \right) = 0$$

алгебраическим уравнением сохранения массовых долей всех фаз:

$$Y_{Liq} + Y_{Gas} = 1$$

и алгебраическим выражением для плотности смеси:

$$\rho = \left(\frac{Y_{Liq}}{\rho_{Liq}} + \frac{Y_{Gas}}{\rho_{Gas}} \right)^{-1},$$

а также выражением для изоэнтропийной скорости звука среды (смеси):

$$a = \sqrt{\gamma \frac{\partial p}{\partial \rho}} = \sqrt{\frac{\gamma}{\rho^2} \left(\frac{Y_{Liq}}{\rho_{Liq}^2} \frac{\partial \rho_{Liq}}{\partial p} + \frac{Y_{Gas}}{\rho_{Gas}^2} \frac{\partial \rho_{Gas}}{\partial p} \right)^{-1}}$$

Приведённая функция является нелинейной относительно давления, что создаёт дополнительные трудности, поскольку даже при относительно небольших скоростях (порядка 10м/с) течение может становиться «около-» или даже «сверхзвуковым», приводя к появлению скачков плотности.

Важным приложением таких гомогенных моделей может являться моделирование водокольцевых насосов [26], которые используются в энергетике для создания разрежения высокой степени.

Принцип работы насоса (рис. 41) относительно прост и базируется на двух законах — законе сохранения массы и законе сохранения импульса. Ротор машины размещён с эксцентриситетом относительно статорной части, имеющей цилиндрическую форму. При вращении ротора жидкость в рабочей части за счёт центробежных сил «разбрасывается» к периферии, образуя между валом ротора и межфазной поверхностью жидкий кольцевой канал переменного сечения. При проталкивании прокачиваемой среды (газа) лопастями ротора через расширяющуюся часть жидкого кольцевого канала, происходит расширение среды и как следствие — создаётся разрежение на всасе. Затем кольцевой канал сужается и проталкивание газа через него приводит к росту давления на выходе из насоса.

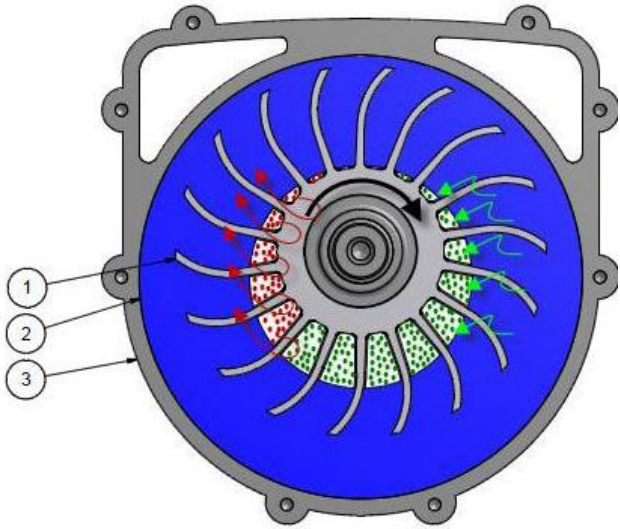


Рис. 41. Схема работы и устройства рабочей части водокольцевого насоса. Синим цветом показана жидкость создающая сужающийся-расширяющийся канал, белым цветом — пространство для прохождения прокачиваемого газа, зелёными точками — входящий поток среды, красными точками — уходящий поток среды. 1 — лопасти ротора, приводящие в движение жидкость 2, создающую канал переменного сечения между поверхностью ротора и межфазной границей. 3 — статорнаячастьнасоса.

Fig. 41. Diagram of operation and flow structure in a liquid ring pump. Liquid forming the converging-diverging channel is shown with blue, passage for pumped gas with white, incoming flow of gas with green points, outgoing flow of gas with red points. 1 — rotor blades which move the liquid 2, which forms variable cross-section channel between the rotor surface and phase boundary. 3 — stator part of the pump.

В работе в качестве модели для тестирования кода была выбрана конструкция близкая к реальной, переданная Dr.-Ing. Jörn Beilke [25]. Для соединения подвижных и неподвижных частей модели использовались поверхности интерполяции данных. Для этого между соответствующими частями создавался зазор, который выбирался либо исходя из конструкторской документации, либо из соображений снижения времени расчёта (чем тоньше слой, тем больше время счёта). В начальный момент времени расчётная область была «залита» жидкостью согласно её предполагаемому положению при работе на номинальной мощности. Скорость вращения вала увеличивалась ступенчатой функцией от 0 до 200 рад/с, давление на всасе снижалось с 100кПа до 60кПа. В результате расчёта были получены распределения полей давления, скорости и объёмной и массовых долей в водокольцевом насосе (рис. 41). Сделана оценка подачи при скорости вала 200 рад/с, перепаде давлений 40 кПа — 5 $м^3$/ч. Сравнивая данную оценку с экспериментальной величиной для перепада 40 кПа — 16$м^3$/ч при скорости вращения вала 298 рад/с (см [25]), можно сделать вывод о качественно правильном воспроизведении явлений с помощью данной модели, поскольку:

- переход с частоты вращения 298рад/с до 200рад/с при сохранении перепада должен снизить подачу по крайней мере на 1/3;
- величина зазора между вращающимся ротором и подводящими/отводящими патрубками значительно меньше чем заданная в модели (в несколько раз), что очевидно сказывается на увеличении модельных протечек.

Качественный анализ течения (рис. 42) показывает сжатие-расширение среды в водокольцевом зазоре, наличие трансзвуковых зон в областях с объёмной долей воздуха около 50%.

Исходя из предварительного анализа результатов расчётов можно утверждать применимость модели для решения задач подобного класса.

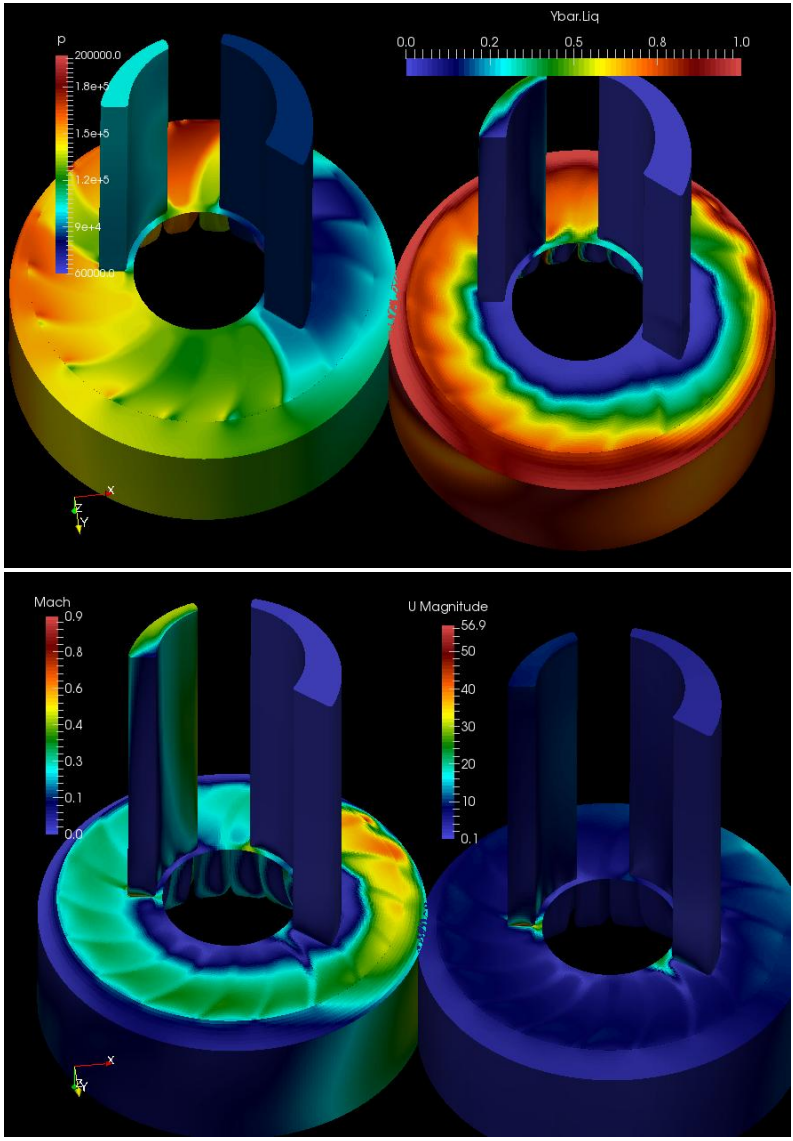


Рис. 42. Качественная картина течения в водокольцевом насосе. Вверху слева: поле статического давления. Вверху справа: поле объёмной доли жидкости. Внизуслева: эффективное число Маха. Внизусправа:полемодуляскорости.

Fig. 42. Qualitative analysis of flow in a liquid ring pump. Top left: field of static pressure. Top right: field of volumetric fraction of liquid. Bottom left: effective Mach number. Bottom right: magnitude of speed.

7. Заключение

1. Сформирован список задач для оценки адекватности гибридного метода моделирования сжимаемых течений. Список включает в себя задачи с аналитическим решением или экспериментальными данными и относящиеся к следующим прикладным направлениям: сжимаемые транс- и сверхзвуковые течения, распространение акустических волн, несжимаемые течения, течения газа и плазмы с высоким отношением давления (5), струйные течения, многокомпонентные и двухфазные течения.

2. Тестирование на задачах с известным аналитическим решением показало, что по крайней мере в областях сверхзвукового или околозвукового течения свойства гибридного метода слабо отличаются от метода Курганова-Тадмора, взятого за основу. В области несжимаемых течений поведение гибридного метода полностью соответствует стандартным схемам типа PISO/SIMPLE.

3. Метод позволяет решать задачи с распространением акустические колебаний, разрешать сжимаемые двухфазные или многокомпонентные течения, исследовать движение плазмы. Продемонстрирована возможность использования метода для решения задач с подвижной расчётной сеткой (в неинерциальной системе отсчёта).

Список литературы

[1]. M. Kraposhin, A. Bovtrikova, S. Strijhak. Adaptation of Kurganov-Tadmor Numerical Scheme for Applying in Combination with the PISO Method in Numerical Simulation of Flows in a Wide Range of Mach Numbers. Procedia Computer Science, 66:43–52, 2015
[2]. OpenFOAM: http://openfoam.org/
[3]. J.D. Anderson, Jr. Modern Compressible Flow: With Historical Perspective. New York: McGraw-Hill, third edition, 2003
[4]. F.M. White. Fluid Mechanics. McGraw-Hill Book Co., New York, NY, third Edition, 1994
[5]. H.E. Smith. The Flow Field and Heat Transfer Downstream of a Rearward Facing Step in Supersonic Flow. Technical report ARL 67-0056. Aerospace Research Laboratories, Ohio, 1967 (Mar.)
[6]. ANSYS Fluid Dynamics Verification Manual, Release 15.0, 2013
[7]. G.D. Garrard, W.J. Phares. Calibration of the PARC Program for Propulsion-Type flows. AEDC-TR-90-7, July, 1990
[8]. М.П. Галанин, Е.Б. Савенков. Методы численного анализа математических моделей. М.: ИздательствоМГТУим. Н.Э. Баумана, 2010
[9]. C. Liang. High-order accurate simulation of low-Mach laminar flow past two side-by-side cylinders with Spectral Difference method. Report ACL 2008-4 Aerospace Computing Laboratory, Aeronautics and Astronautics , Stanford University , May 2008

[10]. X. Liu. Wind loads on multiple cylinders arranged in tandem with effects of turbulence and surface roughness. Master thesis, Department of Civil and Environmental Engineering, Louisiana State University, 2003

[11]. F.R. Menter, M. Kuntz, R. Langtry. Ten Years of Industrial Experience with the SST Turbulence Model. Turbulence, Heat and Mass Transfer 4: Proceedings of the Fourth International Symposium on Turbulence, Heat and Mass Transfer, Antalya, Turkey, 12-17 October, 2003. Publisher: 2003 Begell House, Inc.

[12]. J.R. Edwards, M. Ling. Low-Diffusion Flux-Splitting Methods for Flows at All Speeds. AIAA Journal 1998

[13]. R.A.C. Germanos, L.F. de Souza. Analysis of Dispersion Errors in Acoustic Wave Simulations. Thermal Engineering, Vol. 5 - No 01 - July 2006

[14]. Y.-H. Kim. Sound Propagation. An Impedance Based Approach. John Wiley Sons, first edition, 2010

[15]. L.E. Kinsler. Fundamentals of acoustics. Wiley, New York, 2000

[16]. Л.Г. Лойцянский. Механика жидкости и газа. М.: Дрофа, 2003

[17]. K.S. Abdol-Hamid et al. Numerical Investigation of Flow in an Over-expanded Nozzle with Porous Surfaces. 41st AIAA/ASME/SAE/ASEE Joint Propulsion Conference and Exhibit, 2005

[18]. S.C. Asbury, C.A. Hunter. Static Performance of a Fixed-Geometry Exhaust Nozzle Incorporating Porous Cavities for Shock-Boundary Layer Interaction Control. NASA Langley Research Cente, 1999

[19]. R. Engeln, S. Mazouffre, P. Vankan, D.C. Schram, N. Sadeghi. Flow dynamics and invasion by background gas of a supersonically expanding thermal plasma. Plasma Sources Sci. Technol. 10 (2001) 595–605

[20]. S.E. Selezneva, M.I. Boulos, M.C.M. van de Sanden, R. Engeln, D.C. Schram. Stationary supersonic plasma expansion: continuum fluid mechanics versus direct simulation Monte Carlo method. Journal of Physics D: Applied Physics, Volume 35, Number 12, http://dx.doi.org/10.1088/0022-3727/35/12/312

[21]. H. Jasak, Z. Tukovic. Dynamic mesh handling in OpenFOAM applied to fluid-structure interaction simulations. V European Conference on Computational Fluid Dynamics, ECCOMAS CFD 2010, Lisbon, Portugal,14-17 June 2010

[22]. O. Petit, H. Nilson, M. Page, M. Beaudoin. The ERCOFTAC Centrifugal Pump OpenFOAM Case-Study. In Proceedings of the 3rd IAHR International Meeting of the Workgroup on Cavitation and Dynamic Problem in Hydraulic Machinery and Systems, Brno, Czech Republic, 2009

[23]. J.F. Combès. Test Case U3: Centrifugal Pump with a Vaned Diffuser. ERCOFTAC Seminar and Workshop on Turbomachinery Flow Prediction VII, Aussois, jan 4-7, 1999

[24]. K. Wittig. Konstruktion einer Gastubine fuer Modellflugzeuge und Dokumentation der Auslegungsrechnungen. Muenchen, 24 September 1993

[25]. Strömungssimulation Flüssigkeitsringpumpe. Projekt 1. Ingenieurburo beilke, 28.09.2015, Dresden

[26]. H. Ding, Y. Jiang, H. Wu, J. Wang. Two Phase Flow Simulation of Water Ring Vacuum Pump Using VOF Model. ASME/JSME/KSME 2015 Joint Fluids Engineering Conference, Volume 1: Symposia, Seoul, South Korea, July 26–31, 2015

[27]. M. Kraposhin, C. Brouzet, T. Dauxois, E. Ermanyuk, S. Joubaud, I. Sibgatullin. Direct numerical simulation of internal gravity wave attractor in trapezoidal domain with oscillating vertical wall. Proceedings of ISP RAS, 26(5):117–142, 2014. DOI: 10.15514/ISPRAS-2014-26(5)-6

[28]. В.А. Васильев, М.В. Крапошин, А.Ю. Ницкий, А.В. Юскин. Применение HPC-технологий для решения пространственных задач мультифизики. Вычислительныеметодыипрограммирование, 12(1):160–169, 2011

Study of capabilities of hybrid scheme for advection terms approximation in mathematical models of compressible flows

M.V. Kraposhin <m.kraposhin@ispras.ru>
Institute for System Programming of Russian Academy of Sciences, Russia, Moscow, Solzhenitsyna str., 25

Annotation. The hybrid method for approximation of advective terms is proposed in order to resolve flows in the wide Mach numbers region. This hybrid method is based on the Kurganov-Tadmor (KT) scheme and projection method PISO (Pressure Implicit with Splitting Operators). To construct this method Kurganov-Tadmor scheme for convective fluxes was formulated in implicit manner together with introduced blending function which switches between compressible regime (KT) and incompressible regime (PISO) depending on local characteristics of the flow. Such hybrid scheme gives next benefits: a) implicit treatment of diffusive terms allows to remove time step restrictions imposed by this kind of processes when approximated with explicit scheme; b) implicit formulation of convective terms together with mixing between PISO and KT produces better stability relied only on the flow Courant number, removing acoustic Courant number restrictions at low Mach number flows; c) however, acoustic flows can can also be reproduced — in this case, local acoustic Courant number must be decreased to values less the 1 in the whole computational domain; d) utilization of KT scheme as the basis for approximation of convection terms allows to achieve non-oscillating solution for both acoustic and compressible cases (Mach number larger then 0.3). In order to study hybrid method properties a set of cases with analytical solution or experimental data for different classes of flows was considered: a) compressible flows — propagation of the wave in straight channel (Sod's Problem), supersonic flow over flat wedge, supersonic flow over backward step, flow over forward step with supersonic velocities, flow in supersonic converging-diverging nozzle with shock wave; b) incompressible flows — subsonic flow of laminar viscous fluid in the channel with circle cross section, flow around cylinder in laminar and turbulent regimes, mixing of two gases in 2D flat channel; c) industrial and academic verification tests — superisonic flow of air in NASA nozzle for pressure ratio 5, expansion of stationary equilibrium hot plasma in vacuum; d) qualitative assessment of the hybrid method adequacy for industrial cases — numerical simulation of flows in high speed micro-compressor, simulation of two-phase flow in liquid ring pump. All materials are available for public access through GitHub project https://github.com/unicfdlab.

Keywords: mathematical models, numerical simulation, numerical schemes, compressible flows, acoustics, computational fluid dynamics, open source software.

DOI: 10.15514/ISPRAS-2016-28(3)-16

For citation: Kraposhin M.V. [Study of capabilities of hybrid scheme for advection terms approximation in mathematical models of compressible flows]. *Trudy ISP RAN / Proc. ISP RAS*, vol. 28, issue 3, 2016, pp. 267-326 (in Russian). DOI: 10.15514/ISPRAS-2016-28(3)-16

References

[1]. M. Kraposhin, A. Bovtrikova, S. Strijhak. Adaptation of Kurganov-Tadmor Numerical Scheme for Applying in Combination with the PISO Method in Numerical Simulation of Flows in a Wide Range of Mach Numbers. Procedia Computer Science, 66:43–52, 2015

[2]. OpenFOAM: http://openfoam.org/

[3]. J.D. Anderson, Jr. Modern Compressible Flow: With Historical Perspective. New York: McGraw-Hill, third edition, 2003

[4]. F.M. White. Fluid Mechanics. McGraw-Hill Book Co., New York, NY, third Edition, 1994

[5]. H.E. Smith. The Flow Field and Heat Transfer Downstream of a Rearward Facing Step in Supersonic Flow. Technical report ARL 67-0056. Aerospace Research Laboratories, Ohio, 1967 (Mar.)

[6]. ANSYS Fluid Dynamics Verification Manual, Release 15.0, 2013

[7]. G.D. Garrard, W.J. Phares. Calibration of the PARC Program for Propulsion-Type flows. AEDC-TR-90-7, July, 1990

[8]. M.P.Galanin, E.B. Savenkov. [Methods of numerical analysis of mathematical models]. BMSTU, Moscow, Russia, 2010 (in Russian)

[9]. C. Liang. High-order accurate simulation of low-Mach laminar flow past two side-by-side cylinders with Spectral Difference method. Report ACL 2008-4 Aerospace Computing Laboratory, Aeronautics and Astronautics , Stanford University , May 2008

[10]. X. Liu. Wind loads on multiple cylinders arranged in tandem with effects of turbulence and surface roughness. Master thesis, Department of Civil and Environmental Engineering, Louisiana State University, 2003

[11]. F.R. Menter, M. Kuntz, R. Langtry. Ten Years of Industrial Experience with the SST Turbulence Model. Turbulence, Heat and Mass Transfer 4: Proceedings of the Fourth International Symposium on Turbulence, Heat and Mass Transfer, Antalya, Turkey, 12-17 October, 2003. Publisher: 2003 Begell House, Inc.

[12]. J.R. Edwards, M. Ling. Low-Diffusion Flux-Splitting Methods for Flows at All Speeds. AIAA Journal 1998

[13]. R.A.C. Germanos, L.F. de Souza. Analysis of Dispersion Errors in Acoustic Wave Simulations. Thermal Engineering, Vol. 5 - No 01 - July 2006

[14]. Y.-H. Kim. Sound Propagation. An Impedance Based Approach. John Wiley Sons, first edition, 2010

[15]. L.E. Kinsler. Fundamentals of acoustics. Wiley, New York, 2000

[16]. L.G. Loitsiansky. [Fluid and Gas Mechanics]. Drofa, Moscow, Russia, 2003 (in Russian)

[17]. K.S. Abdol-Hamid et al. Numerical Investigation of Flow in an Over-expanded Nozzle with Porous Surfaces. 41st AIAA/ASME/SAE/ASEE Joint Propulsion Conference and Exhibit, 2005

[18]. S.C. Asbury, C.A. Hunter. Static Performance of a Fixed-Geometry Exhaust Nozzle Incorporating Porous Cavities for Shock-Boundary Layer Interaction Control. NASA Langley Research Cente, 1999

[19]. R. Engeln, S. Mazouffre, P. Vankan, D.C. Schram, N. Sadeghi. Flow dynamics and invasion by background gas of a supersonically expanding thermal plasma. Plasma Sources Sci. Technol. 10 (2001) 595–605

[20]. S.E. Selezneva, M.I. Boulos, M.C.M. van de Sanden, R. Engeln, D.C. Schram. Stationary supersonic plasma expansion: continuum fluid mechanics versus direct simulation Monte Carlo method. Journal of Physics D: Applied Physics, Volume 35, Number 12, http://dx.doi.org/10.1088/0022-3727/35/12/312

[21]. H. Jasak, Z. Tukovic. Dynamic mesh handling in OpenFOAM applied to fluid-structure interaction simulations. V European Conference on Computational Fluid Dynamics, ECCOMAS CFD 2010, Lisbon, Portugal,14-17 June 2010

[22]. O. Petit, H. Nilson, M. Page, M. Beaudoin. The ERCOFTAC Centrifugal Pump OpenFOAM Case-Study. In Proceedings of the 3rd IAHR International Meeting of the Workgroup on Cavitation and Dynamic Problem in Hydraulic Machinery and Systems, Brno, Czech Republic, 2009

[23]. J.F. Combès. Test Case U3: Centrifugal Pump with a Vaned Diffuser. ERCOFTAC Seminar and Workshop on Turbomachinery Flow Prediction VII, Aussois, jan 4-7, 1999

[24]. K. Wittig. Konstruktion einer Gastubine fuer Modellflugzeuge und Dokumentation der Auslegungsrechnungen. Muenchen, 24 September 1993

[25]. Strömungssimulation Flüssigkeitsringpumpe. Projekt 1. Ingenieurburo beilke, 28.09.2015, Dresden

[26]. H. Ding, Y. Jiang, H. Wu, J. Wang. Two Phase Flow Simulation of Water Ring Vacuum Pump Using VOF Model. ASME/JSME/KSME 2015 Joint Fluids Engineering Conference, Volume 1: Symposia, Seoul, South Korea, July 26–31, 2015

[27]. M. Kraposhin, C. Brouzet, T. Dauxois, E. Ermanyuk, S. Joubaud, I. Sibgatullin. [Direct numerical simulation of internal gravity wave attractor in trapezoidal domain with oscillating vertical wall]. *Trudy ISP RAN/Proc. ISP RAS*, 26(5):117–142, 2014 (in Russian). DOI: 10.15514/ISPRAS-2014-26(5)-6

[28]. V.A. Vasiliev, M.V. Kraposhin, A.Yu. Nitzkiy, A.V. Yuskin. [Application of HPC-technologies to solving spatial multiphysics problems]. Vychislitel'nye metody i programmirovanie [Numerical Methods and Programming], 12(1):160–169, 2011 (in Russian)