Platform for Testing and Dynamic Control of Software Code Based on Dimensions*

Sergei A. Petrenko [1] [0000-0003-0644-1731], Alexander A. Petrenko [2],
Krystina A. Makoveichuk[3] [0000-0003-1258-0463], Alexander V. Olifirov [3] [0000-0002-5288-2725],
Yan T. Makoveichuk [3] [0000-0002-8919-7828]

[1] Innopolis University, Kazan, Russia

s.petrenko@rambler.ru

[2] MIREA - Russian Technological University, Moscow, Russia

aa_petrenkoa@guu.ru

[3]V.I. Vernadsky Crimean Federal University, Simferopol, Russia

christin2003@yandex.ru
alex.olifirov@gmail.com

Abstract. A possible platform for test automation and dynamic control of software code based on authoring models and methods of invariants of similarity and dimensions is considered. The development of the mentioned platform allowed organizing dynamic control of the program code of modern applications of digital enterprises with high reliability and reasonable complexity.

Keywords: semantics of computing, technology platform, digital economy, digital enterprise, invariants of similarity, dimensions.

1 The relevance of research

The relevance of developing a platform to automatize the testing and dynamic control of the software code of digital platforms is explained by the need to ensure the required quality and speed of development of the relevant software (Qa&testing), and the imperfection of known methods of testing and verification of the software code in the context of a large-scale digital transformation of leading state and commercial enterprises in Russia.

Currently, the following main solutions are used to test digital enterprise applications: Katalon (https://www.katalon.com/), TestComplete (http://tiny.cc/xyt5cz), ReportPortal (https://reportportal.io), TestRail (https://bit.ly/2kfTbia), Aquality Automation (https://aquality-automation.ru/) [3, 14-15].

The advantages of the first Katalon solution (takes up 2.5% of the world market of automated testing tools) include: own development environment with project templates; developed components for analysis of test results; scripting tools (Record & Play). Disadvantages of Katalon include: lack of native support for Desktop application automation; focus on applications written only in Java and Groovy programming languages; lack of flexibility when distributed runs are needed; lack of web version and user interface overload etc.

The advantages of the second Test Complete, Smartbear Software solution (occupies at least 5% of the world market of automated testing tools) include: support for different types of interfaces; the ability to write scripts in different programming languages; a well-established product support process. Disadvantages of Test Complete include: the inability to store the testing history; limited IDE compared to Visual Studio, Idea Jetbrains, etc.; the need for licensing (not distributed free of charge).

The advantages of the third solution ReportPortal (occupies from 2.5% of the world market of automated testing tools) include: support for most Test Runners; possibility of a joint analysis of results; automatic analysis of test results. Disadvantages of Reportportal include: limited ability to develop new tests; complex structure of stored results.

Advantages of the fourth Test Rail solution, An Idera Inc. (occupies up 7% of the world market of automated testing tools) include: a functionally rich structure for organizing tests; a REST API for integration with other systems; and a JIRA defect tracking subsystem. Disadvantages of Test Rail include the lack of a system for generating automated tests; and the lack of visual analysis of test results.

Advantages of the fifth solution Aquality Automation (occupies up to 4% of the world market of automated testing tools) include: a single user-friendly interface for writing and maintaining tests; fast generation, development and launch of automated tests for various types of products (web, mobile, desktop); advanced algorithms for automatically determining the causes of test failures; formation, maintenance and management of a set of tests for a product and a test data storage system; automatic collection, aggregation and visualization of automated testing results; collection and provision of access to testing artifacts for further analysis; reduction of labor costs for regression testing by 15 times, the time of its implementation - by 5 times; integration with continuous build systems, maintaining a lifecycle for detected test failures; integration with cloud testing platforms.

In practice, testing applications of the digital economy, the listed solutions have shown themselves as follows. ReportPortal is effective for pre-scripted automated tests that require analysis of test execution. TestRail is a convenient test storage tool but does not allow using written tests to automate testing. TestComplete provides a tool to automate the application interface but does not provide an analytic system to enable launches and analysis. Aquality Automation combines solving most of the problems in one interface and allows organizing the process of automated testing of digital economy applications in general.

2 Description of the proposed approach

The proposed QA & Testing platform for automated testing of digital economy applications is based on several well-known and author's models and methods of software testing and analytical program verification [1-2, 4-8].

This platform includes a Web portal (Database, Backend and Frontend), as well as corresponding modules for organizing and conducting automated testing of digital economy applications. Thus, a web application provides an interface for developing tests, grouping them into sets, and visualizing test results. This web application consists of the following components: database; Database; Backend and Frontend component. In this case, the components are packaged into a docker image and distributed further in the form of an image. MySQL database is used as Database (it is possible to use other well-known databases). The purpose of the Database is to store any static information: descriptions of tests, launches and their results. SQL procedures are used to optimize work with Database and speed up query processing.

Backend is a portal with REST API for interacting with other platform components. At the same time, the REST API is developed using technologies: Java, Servlet API, Spring Framework. The Backend component contains methods for processing test results from different test runners (for example, TestNG, JUnit, MSTest, RobotFramework, etc.). The backend also provides interfaces for sending data to the platform, including test results, new tests, etc.

The frontend is responsible for visualizing the information received from the database. The user graphical interface is developed using TypeScript, AngularJS, HTML5 technologies. Using the graphical interface, the user can analyze the results obtained in the form of graphs, tables. Also, the user can flexibly manage data using filters and a search system that provides hints based on historical and analyzed information.

Also, the QA & Testing platform contains several modules for automating application types: web, mobile, desktop. Each such module can use various combinations of free automation tools and programming languages. For example, a module for working with Java is designed to automate testing of web applications and is an add-on to the tool for working with a browser via WebDriver. This module allows you to automate web tests for Chrome, Firefox, Safari, IExplorer, Edge. At the same time, tests can be run under the control of an operating system with JDK version 8 and higher installed on it.

The QA & Testing platform supports integration with the most popular Continuous Integration systems (Jenkins, TeamCity, Azure VSTS, Atlassian Bamboo) to run automated tests. At the same time, the system provides a common interface for launching regardless of which Continuous Integration system the integration is configured with. The launcher provides a set of parameters (data files, browser settings, applications, etc.) that are used to automate testing a product with specified parameters.

2.1 An example of checking the semantics of applications.

Consider an operator of the form:

$$A = B \cdot C + \frac{D}{E} + 1 \,. \tag{1}$$

In terms of dimensions, we obtain three corresponding dimensional equations (2–4):

$$(1) \cdot \ln[A] + (-1) \cdot \ln[B] + (-1) \cdot \ln[C] = 0 \,, \tag{2}$$

$$(1) \cdot \ln[A] + (-1) \cdot \ln[D] + (1) \cdot \ln[E] = 0 \,, \tag{3}$$

$$(1) \cdot \ln[A]^1 = 0 \,. \tag{4}$$

If in (1) we consider a numeric constant as a variable (for example, with the name *CONST_1*) of a certain dimension, then operator (2) will add six variables (*A, B, C, D, E, CONST_1*) to the system of constraints. Equation (2) will also change - it will take the following form:

$$(1) \cdot \ln[A]^1 + (-1) \cdot \ln[CONST\_1]^1 = 0 \,. \tag{5}$$

As a result, variable *A* will remain in the system with a nontrivial value until the completion of the calculations [7].

Correctness assertion. Such a change in the status of numeric constants makes it possible to determine the criterion for the semantic correctness of the Technological platform of some digital enterprise in the following form [8]:

Statement 1. For the semantic correctness of the Technological platform of a digital enterprise, the system of dimensional equations built for it with regard to numerical constants must have at least one solution vector among the set of solution vectors, consisting of all nonzero components.

Proof of the statement (by contradiction). The appearance among the variables corresponding to dimensions, such that is identically equal to zero for any values of other variables, means its dimensionlessness. However, this contradicts the condition for constructing a system of restrictions, namely the introduction of dimensions for all variables and process constants. *The statement is proven.*

For the numerical verification of the criterion, based on the matrix *S* of coefficients, we construct a system of equations for the dimension of the matrix *R*, which has a special form:

$$R = \begin{Vmatrix} 1 & 0 & \dots & 0 & c_{1,1} & \dots & c_{1,n-k} \\ 0 & 1 & \dots & 0 & c_{2,1} & \dots & c_{2,n-k} \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 1 & c_{k,1} & \dots & c_{k,n-k} \end{Vmatrix} . \tag{6}$$

The matrix *R* in this form can be represented by the following formula:

$$R_{k\times n} = E_{k\times k} \mid C_{k\times(n-k)}\,, \tag{7}$$

where *E* is the identity matrix, *k* and n are the number of rows and columns of the original matrix *S*, respectively.

To construct a matrix *R*, it is sufficient to use three types of operations:

1. addition of an arbitrary row of the matrix with a linear combination of other rows;
2. permutation of lines;
3. permutation of columns.

The main course of the process of achieving a form is similar to the Jordan-Gauss method (see, for example, [8]). The difference is as follows:

- double pass of the algorithm: first in the forward (top to bottom), and then in the opposite (bottom to top) direction;
- permutation of columns in cases where a nonzero value in any cell within the first k columns, which is not the first nonzero in a row, cannot be turned into zero due to the absence of other nonzero members in this column [8].

As applied to the solution of the system of dimensional constraints, the matrix *R* is identical to the matrix *S*, except for possibly performed column permutations. That is, there is an equivalent

$$(S\cdot X = 0) \Leftrightarrow (R\cdot T\cdot X = 0)\,, \tag{8}$$

where *T* is a square permutation matrix of dimension *n* × *n* corresponding to the permutations of columns in *S* performed at the stage of constructing *R*. This result is due to the nature of the transformations performed on the matrix *S* in the process of constructing the matrix *R*.

Correctness conditions. Formula (8) allows using not the matrix *S*, but the matrix *R* when checking the semantic correctness. For this, we formulate the following statement:

Statement 2. For the first k values of the solution vector of the system of constraints of the *i*-th component to be identically equal to zero, it is necessary and sufficient that all the elements in the *i*-th row of the matrix *C* in the formula be equal to zero.

Let us prove the necessity of the condition (by contradiction). Let there be at least one nonzero element in the *i*-th row of the matrix *C* (for example, in position j). Then, set equal to zero all (*n* – *k*) last variables except for the (*k* + *j*) th, we obtain the following equality:

$$\sum_{p=1,\,p\neq i}^{k} 0\cdot x_p + x_i + \sum_{q=1,q\neq j}^{n-k} c_{i,q}\cdot 0 + c_{i,j}\cdot x_{k+j} = 0 \Rightarrow \tag{9}$$

$$x_i = -c_{i,j}\cdot x_{k+j}\,, \tag{10}$$

from which it follows that in this case, the variable x_i is not equal to zero. We got a contradiction. *The necessity of the condition is proved.*

Let us prove the sufficiency of the condition. If all elements of the i-th row of matrix C are equal to zero, we obtain the following equality:

$$\sum_{p=1,p\neq i}^{k} 0 \cdot x_p + x_i + \sum_{q=1}^{n-k} 0 \cdot x_{k+q} = 0\,, \tag{11}$$

from which the required identity is obtained directly.

$$x_i \equiv 0\,. \tag{12}$$

The statement is proven.

The variables corresponding to the first k columns of the matrix *R* are basic (independent) in the given system of dimension invariants. The variables corresponding to the remaining columns of the *R* matrix are dependent. Thus, the above statement determines the relationship between the incident of anomalous functioning of the Technological platform of a digital enterprise and the situation when one of the basic variables has the dimension "0". The reason for this relationship is that the situation of dimension "0" is impossible according to the methodology for constructing a system of invariants of dimension.

Assessment of the results obtained. This technique (with complete construction of the matrix *R*) is the basis for the construction of optimized algorithms for testing the criterion. As input data, the algorithm uses a matrix of dimension $k \times n$ with elements from *Z*, the result of the work is a Boolean variable, which has the value "True" if the criterion of semantic correctness is met and "False" otherwise. The intermediate results of the algorithm are:

- matrix *C* of dimension $k \times (n - k)$ with elements from the set of rational numbers *Q*, corresponding to the matrix S in the form (8);
- the value of k_{ERR}, equal to 0, if the criterion of semantic correctness is fulfilled, or the number of the first row of the matrix *C*, consisting only of zero elements if a violation of the criterion is found.

In practice, when using the basic technique, a variation of the algorithm for constructing the matrix *R* is possible, which consists in creating it directly during the analysis of each operator of the computational process under study. The allocation of basic variables and the necessary computational transformations over *R* are performed each time a new line is added to it. The purpose of the modification is to have a matrix of dimensionality constraints already reduced to the form (7) at each analysis step.

This algorithm allows:

- eliminate the computational costs associated with the late (within the framework of the Jordan-Gauss algorithm) permutation of the matrix columns;
- to reduce the number of computational operations during the selection of the identity matrix on the left side of the matrix *R*.

The algorithm requires additional storage of the permutation matrix *T* at the entire stage of the analysis of the Technological platform of the digital enterprise and somewhat slows down the access to the matrix elements. However, the use of efficient data structures makes it possible to reduce the additional costs to a negligible amount. The construction of the matrix *R* allows us to detect an incident of anomalous functioning of the mentioned Technological platform before the end of the entire construction (however, it is not at all necessary that the criterion of semantic correctness is violated at the moment of adding information about a semantically incorrect operator). This fact is an advantage of the modified technique in the case of a large number of erroneous packets (intentionally or unintentionally generated). In this case, the receiving station *L*, having not yet decoded the message completely, can decide to ignore it, thereby freeing up its computational resources. This possibility of the technique can be used when it is included in a layered system of protection against cyber attacks of the "denial of service" class. The possibility of early rejection of the packet does not affect the average statistical computational complexity under normal operating conditions. This is since in such conditions the proportion of abnormal implementations of processes in the network protocol stack should tend to zero [3-8].

3 Conclusions

The lack of proper tools for automated testing and analytical verification of digital economy applications can lead to an organization:

- significant financial losses due to complete failure of the system and shutdown of work due to poor quality, incomplete or untimely testing;
- loss of existing and potential clients due to failure to complete tasks on time;
- loss of business reputation, which has been created over the years and has a value in terms of investment and impact on income [9-13].

Note that manual testing and analytical verification of digital economy applications are not always effective, since there is a risk of missing defects due to the human factor, as well as the need to attract a large number of highly qualified engineers, mathematicians and testers (up to 50 people).

The proposed solution - the QA & Testing platform - for automated testing and analytical verification of digital economy applications provides:

- completeness, reliability and consistency of testing and dynamic control of the program code of digital platforms;
- the ability to dynamically control not only the structure but the semantics of digital economy applications;
- the efficiency of detecting “digital bombs” and destructive program code of digital platforms;
- reducing the cost of ensuring the security and cyber resilience of digital platforms in the long term.

This work was supported by the RFBR grant (No. 18-47-160011 p_a)

References

1. Barabanov A.V., Markov A.S., Tsirlov V.L. Methodological Framework for Analysis and Synthesis of a Set of Secure Software Development Controls, Journal of Theoretical and Applied Information Technology, 2016, vol. 88, No 1, pp. 77-88.
2. Biryukov, D. N., Lomako, A. G. Approach to Building a Cyber Threat Prevention System. Problems of Information Security. Computer systems, Publishing house of Polytechnic University, vol. 2, pp. 13–19, St. Petersburg, Russia, 2013.
3. Petrenko Sergei. Cyber Resilience, ISBN: 978-87-7022-11-60 (Hardback) and 877-022-11-62 (Ebook) © 2019 River Publishers, River Publishers Series in Security and Digital Forensics, 1st ed. 2019, 492 p. 207 illus.
4. Petrenko Sergei. Big Data Technologies for Monitoring of Computer Security: A Case Study of the Russian Federation, ISBN 978-3-319-79035-0 and ISBN 978-3319-79036-7 (eBook), https://doi.org/10.1007/978-3-319-79036-7 © 2018 Springer Nature Switzerland AG, part of Springer Nature, 1st ed. 2018, XXVII, 249 p. 93 illus.
5. Petrenko Sergei. Cyber Security Innovation for the Digital Economy: A Case Study of the Russian Federation, ISBN: 978-87-7022-022-4 (Hardback) and 978-87-7022-021-7 (Ebook). River Publishers, River Publishers Series in Security and Digital Forensics, 1st ed. 2018, 490 p.
6. Sergei Petrenko, Elvira Khismatullina. Cyber-resilience concept for Industry 4.0 digital platforms in the face of growing cybersecurity threats. Software Technology: Methods and Tools, 51st International Conference, TOOLS 2019, Innopolis, Russia, October 15–17, 2019, Proceedings. Editors: Mazzara, M., Bruel, J.-M., Meyer, B., Petrenko, A. (Eds.), eBook ISBN 978-3-030-29852-4, DOI 10.1007/978-3-030-29852-4, Softcover ISBN 978-3-030-29851-7, 420 p. URL: https://www.springer.com/gp/book/9783030298517.
7. Sergei Petrenko, Elvira Khismatullina. Method of improving the Cyber Resilience for Industry 4.0. Digital platforms. Software Technology: Methods and Tools, 51st International Conference, TOOLS 2019, Innopolis, Russia, October 15–17, 2019, Proceedings. Editors: Mazzara, M., Bruel, J.-M., Meyer, B., Petrenko, A. (Eds.), eBook ISBN 978-3-030-29852-4, DOI 10.1007/978-3-030-29852-4, Softcover ISBN 978-3-030-29851-7, 420 p. URL: https://www.springer.com/gp/book/9783030298517.
8. S.A Petrenko; D. E. Vorobieva. Method of Ensuring Cyber Resilience of Digital Platforms Based on Catastrophe Theory. 2019 XXII International Conference on Soft Computing and Measurements (SCM). DOI: 10.1109/SCM.2019.8903658, St. Petersburg, Russia, IEEE, 2019.
9. D. N. Biryukov, Cognitive-functional memory specification for simulation of purposeful behavior of cyber systems. Proc. SPIIRAS. 3(40), pp. 55–76 Russia, 2015.
10. D. N. Biryukov, A. P. Glukhov, S. V. Pilkevich, T. R. Sabirov, Approach to the processing of knowledge in the memory of an intellectual system, Natural and technical sciences, No. 11, pp. 455–466, Russia, 2015.
11. E. D. Vugrin and J. Turgeon, "Advancing Cyber Resilience Analysis with Performance-Based Metrics from Infrastructure Assessment," in Cyber Behavior: Concepts, Methodologies, Tools, and Applications, Hershey, PA, IGI Global, 2014, pp. 2033-2055.
12. D. N. Biryukov, A. G Lomako, Yu. G. Rostovtsev, The appearance of anticipatory systems to prevent the risks of cyber threat realization, Proceedings of SPIIRAS, Issue. 2 (39), pp. 5–25, Russia, 2015. DOI: http://dx.doi.org/10.15622/sp.39.1

13. D. N. Biryukov, Y. G. Rostovtsev, Approach to constructing a consistent theory of synthesis of scenarios of anticipatory behavior in a conflict. Proc. SPIIRAS. 1(38), pp 94–111, Russia, 2015. https://doi.org/10.15622/sp.38.6
14. E. D. Vugrin and J. Turgeon, "Advancing Cyber Resilience Analysis with Performance-Based Metrics from Infrastructure Assessment," in Cyber Behavior: Concepts, Methodologies, Tools, and Applications, Hershey, PA, IGI Global, 2014, pp. 2033-2055. DOI: 10.4018/jsse.2013010105
15. Florin, M. V., & Linkov, I. (Eds.), (2016) IRGC Resource Guide on Resilience. Lausanne: EPFL International Risk Governance Council (IRGC). DOI: 10.5075/epfl-irgc-262527