DOI: https://doi.org/10.32836/2521-6643-2019-2-58-6
UDC 004.738:004.94

O. N. Martyniuk, PhD of Technical Sciences,
Associate Professor of the Computer Intellectual
Systems and Networks Departament,
Odessa National Polytechnic University
T. Ahmesh, Master, Graduate Student
of the Computer Intellectual Systems
and Networks Departament,
Odessa National Polytechnic University
O. V. Drozd, Doctor of Tehnical Sciences,
Professor of the Computer Intellectual
Systems and Networks Departament,
Odessa National Polytechnic University
O. N. Molotkov, PhD of Tehnical Sciences,
Associate Professor of the Computer Sciences
and Software Engineering Departament,
University of Customs and Finance
K. V. Smoktii, PhD in Economics, Associate
Professor of the Information Technologies
Departament, Vasyl' Stus Donetsk
National University

SINGLE-LEVEL METHOD OF BEHAVIORAL ONLINE TESTING OF DISTRIBUTED INFORMATION SYSTEMS

The article considers a single-level method of behavioral online testing with recognition of the reference behavior of DIS. The method features are the evolutionary search for reference behavior in the flow of DIS functioning, parallelization of operations of check recognition and evolutionary development of check populations. The method allows to reduce the complexity and time of check, to make it compatible with the real functioning of the DIS.

Key words: *behavioral online testing; recognizing experiments; Petri nets recognized behavior; identificators; check evolution system.*

Тестування та діагностика складних розподілених інформаційних систем (РІС), що забезпечують їм необхідний рівень надійності функціонування та працездатності, часто повинні виконуватися на системному, інформаційному та функціональному рівнях, які представляють поведінку РІС.

Ключові слова: *поведінковий робочий контроль; експерименти розпізнавання; розпізнавання поведінки мереж Петрі; ідентифікатори; контрольні еволюційні системи.*

Рассматривается одноуровневый метод поведенческого рабочего контроля с распознаванием эталонного поведения РИС. Особенностями метода являются эволюционный поиск эталонного поведения в потоке функционирования РИС, распараллеливание операций контрольного распознавания и эволюционного развития контрольных популяций. Метод позволяет сократить сложность и время контроля, сделать его совместимым с реальным функционированием РИС.

Ключевые слова: *поведенческий рабочий контроль; эксперименты распознавания; распознавания поведения сетей Петри; идентификаторы; контрольные эволюционные системы.*

Problem formulation. Complicating computer networks, the emergence of new distributed information technologies and systems (DIS) [1], the growth of their speed, information and computing power [2], accompanied by the expansion of implementation areas and increased criticality [3] and the rapid development of DIS reliability tools [4]. In such means, automated systems of technical diagnostics occupy the most important place [5], including, among other things, subsystems of online and offline testing and diagnosis [6].

Analysis of recept researches and publications. To date, a number of effective models and methods of online and offline testing and diagnosis have been developed, based on the use of structural, informational, hardware, and functional features of existing and promising DIS and their components [8–10].

However, this use is seriously complicated by the trends of increasing distribution and intellectualization of DIS [1], a sharp increase in the degree of component integration and concealment of the internal structure [8] of the VLSI, FPGA, microprocessors [11]. As a result, the methods of online and offline testing and diagnosis of real DIS are shifted to the level of system, information, functional specifications [12], and their computational becomes NP-complex [13], especially when the appearance of agent properties – autonomy, mobility, intellectuality, co-operativeness [14] for components of DIS. As a result, systemic, decompositional, behavioral, information-functional [15–17], as well as intelligent neural and evolutionary-genetic [18; 19] methods are developed intensively. In most cases, integrated methods [20] and their based technologies of online and offline testing and diagnosis are used to achieve greater effect.

The decompositional approach, in particular, makes significant use of information about the distributed spatial structure of DIS [17], but in most cases the verification does not involve information about the temporary multi-level structure of DIS, of course. appearing in the process of their end-to-end design. In this case, the testing and diagnostic process can be simplified by presenting a special build for the hierarchies of nested checks, which for offline testing and diagnosis is fairly regular [21] (usually going from basic simple checks to complex checks). However, in behavioural testing, which involves solving the problem of background (passive) recognition of reference behaviour in the functioning of the object being tested, such a regular assembly of checks is not possible, the assembly is done in random, subordinate to the basic functioning of the mode, and repeatedly changes direction, both from simple checks to complex (ascending) and from complex to simple (downward). End-to-end multi-level organization of behavioural testing, both online and online, in addition to this imposes new conditions of hierarchical compatibility and inheritance (or not destruction) of multi-level checks (their broadcasts down and encapsulations up).

In this regard, the multi-level decompositional bi-directional assembly of checks in the recognition of reference behavior in behavioral online and offline testing and diagnosis of DIS is of interest.

Purpose of the article. The purpose of the article is to develop an evolutionary method of behavioural online testing based on a behavioural check model of DIS based on Petri nets by determining the evolutionary mechanisms of identified check behaviour.

Main material. The behavioural work of DIS presented in this paper is based on the recognition of the reference behaviour of DIS and its components in accordance with the conditions determined by the formal model of behavioural online and offline testing [22]:

$$cS=(W^{\wedge}, Pr, Ci, Cp, Sg_{ta}, Cc_{t}). \tag{1}$$

The model is based on the extended interval-probability properties of the Petri network and represents special objects – $W^{\wedge}$ – registered behavior of the object being inspected,, checked reference properties *Pr*, reference behavioral identifiers *Ci*, check primitives *Cp* and fragments *Cf*, as well as special operations – identification (as finding), identification as coincidence (further coincidence), determinization and relationships – compatibility/incompatibility, ordering, uncertainty/indifference for them in signature Sg_{ta}, some check strategy Cc_{t}. Special objects form a carrier set, and special operations and relationships, replenished with multiple and vector operations, form signatures respectively operations and relationships of the algebraic system – a formal model of behavioral DIS work control, which determines the general conditions of its conduct in a particular method. Consequently, the initial carrier set, the signatures of operations and the relationship of the formal model of work control should be formed at the preprocessor stage of behavioral work control Consequently, the initial carrier set, the signatures of operations and the relationship of the formal model of online testing should be formed at the preprocessor stage of behavioral online testing.

The background mode of behavioral check recognition allows only the observation of DIS (registration of input and output events in the respective input and output alphabets), and not control the flow of its all events. Therefore, the background mode should perform a full check analysis of current event recognition on the current carrier set at the main stage, with the possibility of using the results of previous experience. Moreover, the formation of recognized behavior, as noted above, should be performed in a random, subordinate mode to the basic functioning of DIS, both on the basis of assembling simple checks in complex, and on the basis of disassembly of complex checks in simple, with multiple changes in this direction. These features of recognition analysis lead to the formation of many variants of current checks and assume their current ranking by the criteria of completeness, length and multiple of online testing, as well as the distribution of these options for execution. Thus, in the construction of the recognition process there is both a random component of the input-weekend event flow, and a purposeful ranking according to the specified criteria. These properties determine the feasibility of evolutionary optimization by special evolutional check system *Ce* [19] in the method of forming recognized behavior $Cc_t=Ce$:

$$Ce = (W^{\wedge}, Ex, Lx, Sg, Cf^{f}). \tag{2}$$

In the system *Ce* $W^{\wedge}$– recorded population behavior, *Ex* – check primitives-individuals of evolution, *Lx* - binding primitives-individuals of evolution, *Sg* – signature of evolutionary operations of mutation and crossover, evolutionary function of fitness and choice, Cf^{f} – final a set of check fragments, that meet the goals and requirements for completeness, length and multiplicity.

One-level behavioral online testing method. In accordance with the general strategy of the check analysis, the procedure of localized single-level agent-based behavioral online testing DIS evolutionarily builds checks of properties *Pr* in registered behavior $W^{\wedge}$ based on reference behavioral identifiers *Ci*, control primitives *Cp* and fragments *Cf* [22] using special operations and relationships of signature Sg_{ta} during the main functioning of DIS. The result of the check establishes the correspondence of the component Petri subnets (PSN) – the reference models of the form *S(f)* for the checked DIS components and the actual models of the form $S(f)^{\wedge}$ for these components [22].

The preparatory stage of the method based on the formal *cS* behavioral check model forms the initial carrier set – behavioral identifiers *Ci* and reference initial check fragments $Cf_0=\cup_{i\in I}cf_{0i}$, the latter defined as appropriately indivisible at the domain level. For the model *cS*, it is possible to talk about check primitives *Cp*, which may form a subset of the initial check fragments $Cp\subseteq Cf_0$.

The main stage of the method during the main operation of some verifiable joint venture $S(f)^{\wedge}$ for fragments of the current fixed behavior $W^{\wedge}=\cup_{j\in J}w_j^{\wedge}$, in particular, the initial $W_0^{\wedge}=\cup_{j\in J}w_{0j}^{\wedge}$ and the current set of modified check fragments $Cf=\cup_{i\in I}cf_i$ ($Cf_0=\varnothing$) based on the model of behavioral online testing *cS* performs an evolutionary search for the next identifiers from *Ci*, initial check fragments from Cf_0, in particular, check primitives from *Cp*. Formation of a new current set of modified check fragments $Cf'=\cup_{i\in I}cf_i'$, as a result of markings found by identifiers from *Ci* positions/transitions, as well as markings of initial check fragments from Cf_0, in particular, *Cp*.

Specifically, test relationships compatibility/incompatibility check into the check fragments from the current set of *Cf* for positions/transitions, where identifiers from *Ci*, initial check fragments from Cf_0 and check primitives from *Cp* are found, when performing an operation of identification α. Further, operations coincidence β and determinization γ are performed [21]. As result, a revamped set *Cf'* of modified current check fragments is being formed in the current previously recorded behavior $W^{\wedge}=\cup_{j\in J}w_j^{\wedge}$. However, during these search-identification α, coincidence β and determinization γ in the formation of the updated set of *Cf'* in the current fixed behavior of the $W^{\wedge}=\cup_{j\in J}w_j^{\wedge}$, which contains check new fragments from the updated set of *Cf'*, this behavior $W^{\wedge}$ can be updated again/replenished itself to $W^{\wedge'}=\cup_{j\in J}w_j^{\wedge'}$. This is the due to continued basic functioning of DIS, with $W^{\wedge}\leq W^{\wedge'}\leftrightarrow(\forall w_j^{\wedge}\in W^{\wedge}(\exists w_{0j}^{\wedge'}\in W^{\wedge}(w_{0j}^{\wedge'}=w''w_j^{\wedge}w'''\&w'', w'''\in(X\times Y)^*\cup e)))\&(|W^{\wedge}|\leq\ \leq|W^{\wedge'}|)$. This circumstances necessitate another evolutionary step of search-identification α, coincidence β and determinization γ for $W^{\wedge'}$.

As noted, behavioral online testing of separate component of DIS – Petri net (PN) *S(f)* – is performed on the base of a pseudo-random purposeful search from an evolutionary system *Ce* [20] with a modified check signature of operations and functions (2).

Behavioral check is applied signature of evolutionary component operations and functions on the reference positions/transmissions, containing μ – partial binary mutation operation based on some, in particular, pseudo-random, substitution/expansion of own recognized behavior from *Cf* some other compatible (adjacent in reference positions) external (infectious) for *S(f)* – behavior from *Cf'*; κ - partial binary multipoint operation of crossover, based on some replacement/extension of own behavior from *Cf* compatible (adjacent in reference positions) behavior from *Cf'*; ϕ – partial binary operation of immunity, based on a search for an earlier, preserved mutational experience; φ – three-core fitness-function for new check fragments – results of mutation and crossover operations; σ – three-core function of choice of operands for mutation and crossover operations.

In the evolutionary operations of the crossover κ and, to a lesser extent, mutations μ are used behavioral control and recognition operations, namely identification α, coincidence β, determinization γ.

The functions of fitness φ and, above all, the choice σ, taking into account the necessary criteria of completeness and complexity of check, can have control parameters, external in relation to the check evolution, which are determined independently of it.

The method of single-level behavioral online testing on based of evolutionnary search implements behavioral check functions and evolutionary functions, allows parallelization, in particular, in the "wave" search in width – multi-process (multi-thread) and multi-agent.

There are two stages in the method - preparatory and basic:

Preparatory stage

1. At the preparatory stage (with respect to the online testing) for the reference PN *S(f)* (or some PSN from PN *S(f)*) simultaneous and independent of each other zero 1st and 2nd processes (or threads) are launched, which:

a) forms the 1st preparatory process (or thread) to highlight the verifiable properties *Pr* for PN *S(f)*, defined as *F* and *S* functions, as well as a priori-set indivisible reference fragments of behavior $\varnothing \subseteq Cf^{Pr}$;

b) reshapes at the end of the allocation of properties *Pr* – 1st process (new – at the end of the first process from the previous item a)), the 2nd, 3rd, ..., i-th preparatory processes (or streams), to synthesize identificators *Ci* of support positions and transitions during the parallel evolutionary, partial or complete construction of a special automata Rabin-Scott *H(S(f)*, as a special graph of multiple, achievable markings for PN *S(f)* based on its behavior in the characters of the input (manager) *X* and output (observed) *Y* alphabets;

c) reworks at the end of the synthesis of *Ci* – 1st preparatory process (or thread) (new process – at the end of previous processes from the item (b)) to synthesize check 0-primitives from *Cp* (as the inclusion of the received identifiers *Ci*, associated with incidental identifiable support positions and transitions), 2nd, 3rd, ... i-th preparatory processes (or threads) for the synthesis of check primitives from *Cp* during the parallel evolutionary application of the coincidence-crossover operation for sets of pairs of checked properties *Pr* (item 1*a*) and identifiers *Ci* (item 1*b*), associated with common, incidentally for its, identifiable support positions and transitions;

d) independent of the steps 1*a* – 1*c* and simultaneous with them – 1th preparatory process (or thread) takes (registers) a priori pre-set initial behavior $W_0^{\wedge}$ of verifiable model $S(f)^{\wedge}$ – initial unrecognized check fragments of behavior, prior to the beginning of the check, and $\varnothing \subseteq W_0^{\wedge}$, at zero step, the current behavior $W^{\wedge}$ is accepted equal to the initial $W^{\wedge} = W_0$.

Main stage.

2. At the current moment of event time in a reed reconditioning iteration of procedures by 1th main process (or thread) (preparatory from Step 1d) records the updated behavior of the $W^{\wedge} \Leftarrow W' \cup W^{\wedge}$ verifiable model $S(f)^{\wedge}$, where W' is a new recorded behavior for the period of the last previous steps of the procedure, as preparatory 1a-1c (at the first performance of the item 2, if there have not been any, $W' = \varnothing$), and performed on previous iterations of the 3-5 item procedure, followed by a rollback to item 2.

3. At the current set of check fragments of the functioning of model $S(f)$, registered before p.3, and the current set of its already confirmed positions $P^{\wedge}$ and transitions $\mathrm{T}^{\wedge}$ (support in $W^{\wedge}$), named (identified) in $W^{\wedge}$ to form the current check fragments Cf, the 2nd parallel process (or thread) begins to perform parallel (branching), background for the main functioning of DIS, eventfully initiated from each observed input-output event, 3rd, 4th, ..., i-th evolutionary processes (or threads) of recognition (with immediate transmission by appropriate processes (or threads) of item 3a–3c successful results to item 4):

a) recognition (identification ($\alpha \in Sg_1$) of current identifiers $Ci^{\wedge'}$ for new, as yet unconfirmed for this moment (i.e. not yet supported in W) current positions $P^{\wedge'}$ and transitions $T^{\wedge'}$, as well as the corresponding naming of these recognized positions $P^{\wedge'}$ and transitions $T^{\wedge'}$, as a result of this new $Cf^{\wedge'}$ and updated current check fragments $Cf^{\wedge} \leftarrow Cf^{\wedge} \cup Cf^{\wedge'}$, as an updated current markup of input behavior $W^{\wedge} \leftarrow W^{\wedge'} \cup W^{\wedge}$;

b) recognition of related (incidental) with the new recognized positions $P^{\wedge'}$ and the transitions $T^{\wedge'}$ of the current unconfirmed check primitives from $Cp^{\wedge'}$ and the verifiable properties from $Pr^{\wedge'}$, the definition of coverage of a all set of reference primitives Cp and properties Pr by the recognized primitives and properties, both this step and all steps to this moment, switching to item 8, if all processes (or threads) are finished or coverage is fully;

c) recognition of the previously confirmed current transporting (connecting) paths $Cl^{\wedge}$ from the updated current $Cf^{\wedge}$ and $W^{\wedge}$ to the unconfirmed (not yet supported in the updated current $W^{\wedge}$) positions and transitions, namely: confirmed (already reference) updated current steps-primitives from $Cp^{\wedge}$ and updated current shortest paths from $Cl^{\wedge}$.

4. For the current set of updated check fragments $Cp^{\wedge}$, the next i+1th parallel process begins to execute eventfully initiated (as registration), parallel (branching), background for the main functioning of DIS, all sorts at this moment, evolutionary i+2-th, i+3-th, ..., i+j-th-processes (or streams) for conversion opera-

tions from the signature $Sg_1=\{\alpha, \beta, \gamma\}$ (the recognition-identification α of supported positions and transitions, coincidence β of supported positions and transitions, determinization γ after coincidence β) in several iterations, determined for α, β and in general probabilistic for γ, to the moment, where their application does not give new fragments to the $W^\wedge$.

5. For still unfulfilled processes (or threads) from $i+j$-th in items 3, 4 of the regular $i+j+1$-th parallel process begins to perform eventfully initiated (as registration), parallel (branching), background for basic functioning of DIS, all sorts of at this moment the processes (or threads) of waiting for the consistent completion of the $i+j$-th processes (or threads) with their results at the exit of item 4, if the conversions α, β, γ not empty and performed by any processes (or threads) of paragraph 4, then for them the $i+j+1$-th process (or thread) performs an immediate translation to item 2.

6. In the $i+j+2$-th (reformed 1-st) process (or thread) determines the final achieved current coverage of set reference check primitives *Cp*.

7. $i+j+2$-th (reformed 1-st) parallel process (or thread) pre-stops work (all current processes (or threads) are completed at this point), wait appearance of a new input-output event within the first specified time period and goes to item 2, when it appears.

8. $i+j+2$-th (reformed 1-st) parallel process (or thread) pre-stops behavioral periodic check, when covering a all set of reference check primitives *Cp* with a reset of recognition results, confirming of compliance $S(f)^\wedge$ and reference $S(f)$ models, after the second specified period of time, possibly zero, again goes to item 2.

Evolutionary pseudo-random targeted search in most processes (or threads) allows to achieve good results in most cases significantly faster, than the upper estimate of *NP*-computational component, for $S(f)$, complexity of deterministic method.

Estimating the dimension of the method. Larger than dimensions for PN $S(f)$ are ratings for the Rabin-Scott automata $h(S(f))$, which serves to build and analyze identifiers *Ci* of positions and transitions.

For a simple (not multiple) Rabin-Scott automata $h(S(f))$, the number of vertices is no more, than $mn(n-1)$, the upper limit of the total number of fields for vertices in $h(S(f))$ is $2mn(n-1)$, the number of arcs does not exceed $((lm)^n - \mathbf{2}$, the upper estimate of the total number of fields for arcs in $h(S(f))$ is $2((lm)^n - 1)$. Here's the $n=|P|+|T|$, $m=|X|$, $l=|Y|$. The upper estimate of the total number of fields in $h(S(f))$ is $2(mn(n-1)+(lm)^n-1)$. The longest simple identifier also has a length of $n-1$.

For the Rabin-Scott multiple automata $h(S(f))$, the number of vertices is the same, as for the simple one, i.e. the number of vertices is no more than $mn(n-1)$, the upper limit of the total number of fields for arcs in $h(S(f))$ is $2mn(n-1)$, the number of arcs is greater, but does not exceed $(lm2^{n-1})^n - 1$, the upper estimate of the total number of fields for tops in $h(S(f))$ is $2((lm2^{n-1})^n - 1)$. The top estimate of the total number of fields in $h(S(f))$ is $2(mn(n-1)+ +(lm2^{n-1})^n - 1)$. The longest multiple identifier has a total length, also equal to $n - 1$.

The empirical experience of using Rabin-Scott automata $h(S(f))$ for real DIS component models has shown, that simple identifiers are sufficient in most cases. Still, analysis tasks, based on Rabin-Scott, are *NP*-complex.

Comprehensive upper estimate of the total number of elementary operations in behavioral online testing for PN $S(f)$ with simple identifiers is:

$$c \le ((lm)^n 2^{n-1})(2n+3) - 3+n(2m+1). \tag{3}$$

Empirical experience of applying evolutionary methods in recognising experiments for PN $S(f)$ showed an average decrease in the given estimate in the statistical majority of cases compared to deterministic methods in $log2(2nm)$ times.

The average empirical estimate of the number of elementary operations in behavioral online testing for PN $S(f)$ with simple identifiers is:

$$c \le (((lm)^n 2^{n-1})(2n+3) - 3+n(2m+1)+m(n-1)(2n+1))/log2(2nm). \tag{4}$$

Experimental tests of basic behavioral check procedures implementing the developed component method were conducted for medium-complexity DIS facilities and components. For selected objects and error class, comparing experiments for procedures based on deterministic and evolutionary methods for DIS “Smart Home” confirmed a near 90 % decrease in the computational complexity of check (down to 5,000 conditional transitions) and timer (down to 0.1 hours of time), when using an I7-based computer.

These results are in line with analytical estimates. The reduction in check is achieved on the special components of the “Smart Home” DIS, which has partial certainty of functions.

Conclusions and further researches directions.

These method, estimates and results of the experiment show that behavioral online testing tasks are achievable by using the proposed method for simple and medium-sized models with complexity of individual DIS components. This fact makes it possible to develop decompositional methods on its basis.

Thus, the component method of behavioral online testing of DIS in an expanded class of component errors with the features of component evolutionary development of the population of control fragments reduces computational complexity behavioral work control on $log_2(2nm)$ times in comparison with the automata deterministic method of behavioral online testing.

List of sources used:

1. *Coulouris G., Dollimore J., Kindberg T., Blair G.* Distributed Systems: Concepts and Design. 5th ed. Boston: Addison-Wesley, 2011. 1067 p.

2. *Kharchenko V., Gorbenko A., Sklyar V., Phillips C.* Green Computing and Communications in Critical Application Domains: Challenges and Solutions // IX International Conference of Digital Technologies. Zhilina, Slovak Republic, 2013. P. 191–197.

3. *Hahanov V., Litvinova E., Chumachenko S.* Cyber Physical Computing for IoT-driven Services // Springer. 2017. 279 p.

4. *Drozd O., Antoshchuk S., Martinuk A., Drozd J.* Increase in reliability of on-line testing methods using natural time redundancy // Proc. IEEE East-West Design & Test Symposium, Moscow, Russia, 2010. P. 223–229.

5. *Romankevich V. A.* Self-testing of multiprocessor systems with regular diagnostic connections // Automation and Remote Control. 2017. Vol. 78, Iss. 2. P. 289–299.

6. *Drozd O., Antoshchuk S., Rucinski A., Martinuk A.* Parity prediction method for on-line testing a Barrel-shifter // Proc. IEEE East-West Design & Test Symposium, Lviv, Ukraine, 2008. P. 208–213.

7. *Hahanov V., Litvinova E., Obrizan V., Gharibi W.* Embedded method of SoC diagnosis // Elektronika ir Elektrotechnika. 2008. No 8. P. 3–8.

8. *Biswas A., Mukhopadhyay S., Patra A.* Formal Approach to On-Line Monitoring of Digital VLSI Circuits: Theory, Design and Implementation // Journal of Electronic Testing: Theory and Applications. 2005, October. Vol. 20. P. 503–537.

9. *Petty Mikel D., Kim Jungyoon, Barbosa Salvador E., and Pyun Jai-Jeong* Software Frameworks for Model Composition // Modelling and Simulation in Engineering. 2014. Vol. 2014. 18 p. URL:https://www.researchgate.net/publication/270629314_Software_Frameworks_for_Model_Composition

10. *Gomes L., Fernandes J. M.* Behavioral modeling for embedded systems and technologies: applications for design and implementation // InformatIon scIence reference, Hershey, New York, 2010. 466 p. URL:http://tierra.aslab.upm.es/documents/books/Gomes-BehavioralModeling.pdf

11. *Drozd O., Drozd M., Martynyuk O., Kuznietsov M.* Improving of a Circuit Check ability and Trustworthiness of Data Processing Results in LUT-based FPGA Components of Safety-Related Systems // CEUR Workshop Proceedings. 2017. Vol. 1844. P. 654–661.

12. *Gomes L., Fernandes J. M.* Behavioral Modeling for Embedded Systems and Technologies: Applications for Design and Implementation, 2010. 494 p.

13. *Grindal M.* Handling Combinatorial Explosion in Software Testing // Printed by LiU-Tryck, Linköping, 2007. 89 p.

14. *Stuart J. R., Norvig P.* Artificial Intelligence: a Modern Approach // Prentice-Hall, Inc. A Simon & Schuster Company Englewood Cliffs, New Jersey, 2010. 1095 p.

15. *Kudryavtsev V. B., Grunskii I. S., Kozlovskii V. A.* Analysis and synthesis of abstract automata //Journal of Mathematical Sciences. 2010. Vol. 169. Iss. 4. 2010. P. 481–532.

16. *Schamai W.* Model-Based Verification of Dynamic System Behavior against Requirements Method, Language, and Tool // Linköping University SE-581 83 Linköping, Sweden Linköping, 2013. 257 p.

17. *Мартынюк А. Н., Тамим Ахмеш, Мартынюк Д. А., Дрозд А. В.* Поведенческий рабочий контроль сетевых компьютерных систем // Электротех-
нические и компьютерные системы. 2018. № 28 (104). С. 201–207.

18. *Skobtsov Yu. A., Skobtsov V. Yu.* Evolutionary test generation methods for digital devices // Design of Digital Systems and Devices [eds.: M. Adamski et al.]. Berlin: Springer Verlag, 2011. P. 331–361. (Lecture Notes in Electrical Engineering. Vol. 79).

19. *Martynyuk O., Sugak A., Martynyuk D., Drozd O.* Evolutionary Network Model of Testing of the Distributed Information Systems // Proc. 9th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, Bucharest, Romania, 2017.P. 888–893.

20. *Classen A., Cordy M., Schobbens P.-Y. et al.* Featured Transition Systems: Foundations for Verifying Variability-Intensive Systems and Their Application to LTL Model Checking // TSE, 39(8):1069–1089, Aug. 2013.

21. *Sugak H., Martynyuk O., Drozd A.* Models of the Mutation and Immunity in Test Behavioral Evolution // Proc. 8th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, Warsaw, Poland, 2015. P. 790–795.

22. *Martynyuk O., Drozd O., Tamim Ahmesh et al.* Hierarchical Model of Behavior On-line Testing for Distributed Information Systems // The 10th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications. Vol. 2. 18–20September 2019, Metz, France. P. 724–729.

References:

1. Coulouris G., Dollimore J., Kindberg T. and Blair G. (2011), "Distributed Systems: Concepts and Design", 5th ed. Boston: Addison-Wesley, 1067 p. [USA].

2. Kharchenko V., Gorbenko A., Sklyar V. and Phillips C. (2013), “Green Computing and Communications in Critical Application Domains: Challenges and Solutions”, IX International Conference of Digital Technologies, Zhilina, pp. 191–197 [Slovak Republic].

3. Hahanov V., Litvinova E. and Chumachenko S. (2017), “Cyber Physical Computing for IoT-driven Services”, journal Springer, 279 p. [USA].

4. Drozd O., Antoshchuk S., Martinuk A. and Drozd J. (2010), “Increase in reliability of on-line testing methods using natural time redundancy”, journal Proc. IEEE East-West Design & Test Symposium,Moscow, pp. 223–229 [Russia].

5. Romankevich V. A. (2017), “Self-testing of multiprocessor systems with regular diagnostic connections”, journal Automation and Remote Control, vol. 78, iss. 2, pp. 289–299 [Russia].

6. Drozd O., Antoshchuk S., Rucinski A. and Martinuk A. (2008), “Parity prediction method for on-line testing A Barrel-shifter”, Proc. IEEE East-West Design & Test Symposium, Lviv, pp. 208–213 [Ukraine], DOI: 10.1109/EWDTS.2008.5580162

7. Hahanov V., Litvinova E., Obrizan V. and Gharibi W. (2008), “Embedded method of SoC diagnosis”, journal Elektronika ir Elektrotechnika, vol. 8, pp. 3–8 [Russia].

8. Biswas A., Mukhopadhyay S. and Patra A. (2005, October), “Formal Approach to On-Line Monitoring of Digital VLSI Circuits: Theory, Design and Implementation” Journal of Electronic Testing: Theory and Applications, vol. 20, pp. 503–537.

9. Petty Mikel D., Kim Jungyoon, Barbosa Salvador E., and Pyun Jai-Jeong (2014), “Software Frameworks for Model Composition”, journal Modelling and Simulation in Engineering, vol. 2014, 18 p.,available at:https://www.researchgate.net/publication/270629314_Software_Frameworks_for_Model_Composition

10. Gomes L. and Fernandes J. M. (2010), Behavioral modeling for embedded systems and technologies: applications for design and implementation, in Information science reference, Hershey - New York, 466 p. [USA], available at: http://tierra.aslab.upm.es/documents/books/Gomes-BehavioralModeling.pdf

11. Drozd O., Drozd M., Martynyuk O. and Kuznietsov M. (2017), “Improving of a Circuit Check ability and Trustworthiness of Data Processing Results in LUT-based FPGA Components of Safety-Related Systems”, journal CEUR Workshop Proceedings, vol. 1844, pp. 654–661.

12. Gomes L. and Fernandes J. M. (2010), Behavioral Modeling for Embedded Systems and Technologies: Applications for Design and Implementation, press IGI Global, 494 p.

13. Grindal M. (2007), Handling Combinatorial Explosion in Software Testing, Printed by LiU-Tryck, Linköping, 89 p. [Sweden].

14. Stuart J. R. and Norvig P. (2010.), Artificial Intelligence: a Modern Approach, Prentice-Hall, Inc. A Simon & Schuster Company Englewood Cliffs, New Jersey, 1095 p. [USA].

15. Kudryavtsev V. B., Grunskii I. S. and Kozlovskii V. A. (2010), Analysis and synthesis of abstract automata, Journal of Mathematical Sciences, vol. 169, iss. 4, pp. 481–532.

16. Schamai W. (2013), Model-Based Verification of Dynamic System Behavior against Requirements Method, Language, and Tool, Linköping University SE-581 83, Linköping, 257 p. [Sweden].

17. Martynyuk A. N., Tamim Akhmesh, Martynyuk D. A. and Drozd A. V. (2018), *“Povedencheskiy rabochiy kontrol' setevykh komp'yuternykh sistem”* ["Behavioral worker control of network computersystems”], journal *Elektrotekhnicheskiye i komp'yuternyye sistemy* [Electrical and computer systems], vol.28 (104), pp. 201 – 207 [Ukraine].

18. Skobtsov Yu. A. and Skobtsov V. Yu. (2011), Evolutionary test generation methods for digital devices, Design of Digital Systems and Devices, Berlin, Springer Verlag, pp. 331–361, (Lecture Notes in Electrical Engineering, vol. 79) [Germany].

19. Martynyuk O., Sugak A., Martynyuk D. and Drozd O. (2017), “Evolutionary Network Model of Testing of the Distributed Information Systems”, Proc. 9th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, Bucharest, pp. 888–893. [Romania].

20. Classen A., Cordy M. and Schobbens P.-Y. et al. (2013, Aug.), Featured Transition Systems: Foundations for Verifying Variability-Intensive Systems and Their Application to LTL Model Checking, TSE, 39(8):1069–1089.

21. Sugak H., Martynyuk O. and Drozd A. (2015), “Models of the Mutation and Immunity in Test Behavioral Evolution”, Proc. 8th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, Warsawa, pp. 790–795 [Poland].

22. Martynyuk O., Drozd O., Tamim Ahmesh et al. (2019, September, 18–20), “Hierarchical Model of Behavior On-line Testing for Distributed Information Systems”, The 10th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, Metz, vol. 2, pp. 724–729 [France].