UDC 004.738:004.94

Oleksandr N. Martynyuk[1], Candidate of Technical Sciences, Associate Professor of the Department Computer Intellectual Systems and Networks, E-mail: anmartynyuk@ukr.net, Scopus ID: 57103391900, ORCID: 0000-0003-1461-2000
Tamem Ahmesh[1], post-graduate student of the Department Computer Intellectual Systems and Networks, E-mail: tamim.nor@yahoo.com, ORCID: 0000-0001-7342-4339
Bui Van Thuong[1], post-graduate student of the Department Computer Intellectual Systems and Networks, E-mail: govarava@gmail.com, ORCID: 0000-0002-6176-5432
Oleksandr V. Drozd[1], Doctor of Technical Sciences, Professor of the Department Computer Intellectual Systems and Networks, E-mail: drozd@ukr.net, Scopus ID: 55388226700, ORCID: 0000-0003-2191-6758
[1]Odessa National Polytechnic University, Avenue Shevchenko, 1, Odessa, Ukraine, 65044

MULTILEVEL BEHAVIORAL TESTING OF DISTRIBUTED INFORMATION SYSTEMS

Annotation. *Operational features of modern distributed information systems significantly increase the requirements for the reliability of their functioning and tighten the time limits for its restoration. High reliability of work is provided by various tools, in particular, offline and online testing, but often limiting the real time of the operation of these systems. Well-known testing methods use hardware accelerators, decomposition and parallelization, artificial intelligence technologies, but their capabilities do not remove the advisability of developing new models and methods, in particular, high-level ones, based on a special analysis of system behavior. The article presents a multi-level method of online behavioral testing of distributed information systems, based on hierarchical Petri nets, single-level model and method, and also hierarchical model of online behavioral testing. The multi-level method uses evolutionary optimization and has the features of parallel multi-level organization of compatible check evolutions of individual levels, which develop according to the principle of a "wave" algorithm in the background relative to the main functioning of the distributed information systems themselves. To determine the complex testing criteria in the check evolutions of the multilevel method, a hierarchical system of compatible functions of fitness and the choice of objects in the level evolutions of the hierarchical model is proposed. The functions offer completeness estimates for positions, transitions and arcs of Petri nets, linear and nonlinear graph behavior structures, the action of functions is extended to check primitives and fragments. The hierarchy of fitness and choice functions made it possible to introduce estimations of completeness of testing used to optimize evolutionary search in a multi-level coverage of necessary checks. Hierarchical decomposition, evolutionary approach and hierarchy of optimizing fitness and choice functions of a multi-level method made it possible to reduce time and increase the flexibility of organizing online behavioral testing, taking into account the features of distributed information systems.*

Keywords*: Distributed Information Systems; Behavioral Testing; Hierarchical Petri net; Check Model; Evolution; Multilevel Check*

Introduction. Currently, modern distributed information systems (DIS) are characterized by rapid complication, large-scale expansion and penetration of the Internet, a sharp increase in the responsibility of tasks. The main place in the analysis and synthesis of DIS begin to occupy the operational formation of distributed general and special structures of tasks, resources and processes in open and corporate environments. Complex dynamic communications, coordination and cooperation in such systems require overcoming the risks of access, uncertainty, functional disability, failures and errors, incorrect and malicious actions.

Basic solutions to these problems include, on the one hand, a set of security measures and information protection systems (authorization / authentication, digital signature, encryption, access / trust, shielding, virtualization), on the other hand, formal online and offline testing to ensure the necessary performance level DIS. The increasingly frequent appearance in DIS of multi-agent properties of autonomy, mobility, intelligence, cooperativeness significantly complicates these decisions and control.

Thus, we can conclude that it is necessary to develop operational and test check, as part of the analysis, design and maintenance of DIS, using many technologies, with a sharp increase in the completeness and accuracy of the check itself, reducing the cost of resources, in particular time. The operational and test check of DIS, as a rule, includes a complex of deterministic, probabilistic, fuzzy, evolutionary methods, in particular, with the analysis of structural, functional, informational, interface properties and mechanisms of DIS. These methods have their own values for the completeness of testing results and resource costs, often NP-complex. However, the noted features of the development of DIS in combination with the diversity and short duration of their dynamic cooperatives tighten the time conditions for monitoring and diagnostics, often limiting them to the real-time mode.

Traditional solutions to temporary problems use hardware accelerators, decomposition and parallelization of models and methods. But this does not exclude the possibility of further development of high-level formal models and methods of working and test check, in particular, on the basis of testing of reference behavioral properties, agreed at different levels of system, structural and functional design and confirmed during operation of the implementation of design decisions, using effective intellectual check coverage search methods.

DOI:10.15276/hait.04.2019.6 ISSN 2663-0176 (Print)
ISSN 2663-7731 (Online)

Therefore, it is of interest to study methods for verifying the behavioral properties of distributed hierarchical models, in particular, on the basis of evolutionary search. The relevance of this study is due to the need to improve existing testing methods, which can be based on an evolutionary multi-level search for check coverage. In particular, the mutual inter-level coordination of fitness functions and the choice of check evolutions adjacent in the hierarchical model is of interest. As a result, greater completeness is achieved when constructing a compatible composition of adjacent evolutionary check populations and, as a result, reducing the time of behavioral testing as it approaches the real time of DIS functioning.

Literature review. The study of the concepts, models and methods of analysis and synthesis of modern DIS is devoted to an ever-increasing number of works [1]. This is due to the rapidly expanding scope of DIS applications, the expansion of their scale, distribution and sharable, a sharp structural and combinatorial complication of problems solved by DIS, which are increasingly characterized by NP-complexity [2], and a significant increase in the criticality of their application [3]. As a result, promising DIS are characterized by a complex specialized composition of the properties and mechanisms of many technologies and platforms, such as the Web, multi-agent [4], cluster, GRID, cloud, Internet of things [5-6], and intelligent [7-8]. So, the growth of DIS intelligence using fuzzy, evolutionary, and neural methods, their speed, bordering on the real time of the functioning of objects in the domain, are manifested to the greatest extent. The autonomy and mobility of DIS components is usually inherent in subject specialization. But the spatial and temporal structure of interactions, both internal and external, including with objects of the local and global network, is becoming more and more quickly reconfigurable, expanding, multi-level [9-10]. These DIS properties acquire the character of common ones and indicate that DIS also receive signs of dynamic systems, which quickly form special structures of distributed and shared tasks, resources and processes in deployment infrastructure environments.

A higher level of dynamic, situational communications, coordination and cooperation in such systems exacerbates the risks of access, uncertainty, functional failure, failures and errors, incorrect and malicious actions [11-13]. As a rule, these risks are reduced and, in some cases, eliminated by a set of security measures and information security systems, for example, through authorization / authentication, digital signature, encryption, attributes and access / trust rights, multi-level shielding, subject-logical virtualization. Regardless of this, a higher level of reliability of DIS functioning can be additionally provided by means of their formal working and test check [14-17].

All modern methods of deterministic, probabilistic, fuzzy, evolutionary check of their structural, functional and informational properties, mechanisms and their inherent error classes are characterized by validity and completeness of testing and resource costs in their class [18-20]. At the same time, the trends of rapid growth of uncertainty, intelligence, operational reconfigurability, and the diversity of DIS interactions in situational compositions, both in distribution and separation within fuzzy boundaries, and in the tasks assigned to them, persist and even accelerate [21]. This circumstance limits the use of most known methods to analysis of medium-complexity DIS [22]. Relevant analysis, synthesis and maintenance of DIS are often relevant, often NP-complex [23] with the combined use of different methods [24-27], in particular, intellectual [28-33]. In particular, the methods of early design systemic and structural-functional testing [34-35], including formal multilevel behavioral testing [36], special check [37-39], which largely determine the correctness of projects and the reliability of their subsequent implementations, are found to be mandatory.

The relevance of the work is due to the need to improve existing methods of multidimensional behavioral evolutionary testing of DIS based on hierarchical Petri nets, which have the features of inter-level interaction and the influence of related check evolutions and inter-level integration and optimization of fitness and choice functions defined for these evolutions. Inter-level integration and optimization of fitness functions and the choice of level check evolutions used in behavioral testing of DIS allows increasing the completeness of testing and reducing the testing time as it approaches the real time of RIS functioning.

The purpose of this article. The aim of this research is to increase the completeness of testing and reduce the time of behavioral testing of DIS projects and their implementations by developing a multi-level evolutionary testing method by improving inter-level integration and optimization of fitness and choice function systems.

To achieve the research goal for the multi-level evolutionary method of behavioral testing, based on the hierarchical testing model [36], the completeness and structure of the criteria for completeness for the evolutionary testing of the elements of the hierarchical Petri net (IPN) formed by parallel processes of label movement during the functioning and testing of IPN are defined as follows:

– elementary two-component completeness criteria for positions/transitions, elementary arcs connecting positions and transitions;

– total completeness criteria for simple graph structures – chains, trees, hammocks, networks, cycles;
– total completeness criteria for graphs of single-level Petri nets (PN)
– two-level multicomponent completeness criteria for two-level hierarchies of IPN graphs;
– total completeness criteria for identifiers, check primitives and fragments.

The properties of hierarchical model and multilevel interactions

The multilevel method of behavioral online testing of DIS is based on:
– single-level model of behavioral online testing with the signature of level check operations of identification, identification, determination, relations of compatibility, incompatibility, uncertainty and extended precedence for behavioral check fragments;
– single-level method of behavioral of DIS components;
–hierarchical model of behavioral online testing icS [36] with an older check model, substitutable lower check models, a signature of inter-level mappings, quasi-order relations and compatibility, representing inter-level synchronization of recognized check fragments.

To solve these problems, on the basis, in-first, of the autonomous single-level model $cS=(W^{\wedge}, Pr, Ci, Cp, Sg_{cS}, Cc_t)$ with the signature of the level check operations $Sg_{cS}=\{\alpha, \beta, \gamma\}$ – respectively identification, coincidence, determinization, relationship $Rel_{Cs} = \{\sigma, \eta, \tau, \langle \}$ – respectively compatibility, incompatibility, uncertainty, extended precedence for behavioral check fragments *Cf*, in-second, of single-level method of behavioral online testing of the components of DIS, in-third, the hierarchical model of behavioral online testing

$$icS=(cS, \cup_{i\in I} cS_i^c, \cup_{j\in J} cS_j^t, Sg_{iS}),$$

[39] with highest check model *cS*, replacing lowest substitutes check models $\cup_{i\in I} cS_i^c$, $\cup_{j\in J} cS_j^t$, signature of inter-level mapping $Sg_{iS} = \{\chi^{\to p}, \chi^{p\to}, \nu^{\to t}, \nu^{t\to}\}$, relations of quasi-order

$$\psi^+=\psi\cup(\cup_{i\in I}\psi_i)\cup(\cup_{j\in J}\psi_j),$$

and compatibility

$$\xi^+=\xi\cup(\cup_{i\in I}\xi_i)\cup(\cup_{j\in J}\xi_j),$$

representing inter-level synchronization of recognized check fragments from *Cf*, $\cup_{i\in I} Cf_i^p$, $\cup_{j\in J} Cf_j^t$, a multi-level method of behavioral online testing of DIS is proposed. The method evolutionarily builds the check fragments for behavior of highest Petri Net (PN) *S(f)*, based on reference behavioral identifiers *Ci*, check primitives *Cp* and fragments *Cf*, complementing them with compatible, synchronized (vertically translated) checks of behavior fragments with lowest identifiers $\cup_{i\in I} Ci_i^p$, $\cup_{j\in J} Ci_j^t$, check primitives $\cup_{i\in I} Cp_i^p$, $\cup_{j\in J} Cp_j^t$ and fragments $\cup_{i\in I} Cf_i^p$, $\cup_{j\in J} Cf_j^t$ for lowest Petri subnets (PSN) from $(\cup_{i\in I}S(f)_i^p)\cup(\cup_{j\in J}S(f)_j^t)$, detailing macro-positions and macro-transitions from *S(f)*.

The result of the test establishes the conformity of the system highest PN and its component lowest PSN – for reference models of the form *S(f)* and $(\cup_{i\in I}S(f)_i^p)\cup(\cup_{j\in J}S(f)_j^t)$, that is, the full lowest reference hierarchical PN (HPN)

$$S(f)^+=S(f)\backslash((\cup_{i\in I}p_i)\cup(\cup_{j\in J}t_j))\cup$$
$$\cup(\cup_{i\in I}S(f)_i^p)\cup(\cup_{j\in J}S(f)_j^t),$$

of reference DIS or its components and for actual models of the form $S(f)^{\wedge}$ and $(\cup_{i\in I}S(f)_i^{p\wedge})\cup(\cup_{j\in J}S(f)_j^{t\wedge})$, that is, the full lowest checked HPN

$$S(f)^{+\wedge}=S(f)^{\wedge}\backslash((\cup_{i\in I}p_i^{\wedge})\cup(\cup_{j\in J}t_j^{\wedge}))\cup$$
$$\cup((\cup_{i\in I}S(f)_i^{p\wedge})\cup(\cup_{j\in J}S(f)_j^{t\wedge})),$$

of real DIS or its components.

Preprocessor stage

So, during preprocessor stage for the fragments of the recorded behavior of $W^{\wedge}$ of the basic functioning of some highest PN *S(f)* and the recorded behavior $(\cup_{i\in I}W_i^{p\wedge})\cup(\cup_{j\in J}W_j^t)$ of lowest detailing PSN from $(\cup_{i\in I}S(f)_i^p)\cup(\cup_{j\in J}S(f)_j^t)$ based on the hierarchical model of behavioral online testing *icS*, the method performs a multilevel evolutionary parallel search for sets of identifiers accordingly *Ci* and $(\cup_{i\in I}Ci_i^p)$, $(\cup_{j\in J}Ci_j^t)$, check primitives accordingly *Cp* and $(\cup_{i\in I}Cp_i^p)$, $(\cup_{j\in J}Cp_j^t)$. As identifiers and primitives are found, the method forms the set of modified check fragments accordingly $Cf'=\cup_{i\in I}cf_i'$ and $(\cup_{i\in I}Cf_i^{p\prime})$, $(\cup_{j\in J}Cf_j^{t\prime})$. Search for identifiers and primitives, as well as fragment formation, include the application of operations of identification α, coincidence β, determinization γ to behavior $W^{\wedge}=\cup_{i\in I}w_i$ and $(\cup_{i\in I}W_i^{p\wedge})$, $(\cup_{j\in J}W_j^{t\wedge})$.

For the full lowest reference HPN $S(f)^+$ these sets are presented accordingly as:

$$Ci^+=Ci\cup(\cup_{i\in I}Ci_i^p)\cup(\cup_{j\in J}Ci_j^t),$$
$$Cp^+=Cp\cup(\cup_{i\in I}Cp_i^p)\cup(\cup_{j\in J}Cp_j^t),$$
$$Cf^+=Cf\cup(\cup_{i\in I}Cf_i^{p\prime})\cup(\cup_{j\in J}Cf_j^{t\prime}).$$

Significantly, subsets of identifiers $((\cup_{i\in I}ci_{pi})\cup(\cup_{j\in J}ci_{tj}))$, primitives $((\cup_{i\in I}cp_{pi})\cup \cup(\cup_{j\in J}cp_{tj}))$, fragments $((\cup_{i\in I}cf_{pi})\cup(\cup_{j\in J}cf_{tj}))$, that characterize highest behaviors around macro-positions and macro-transitions $((\cup_{i\in I}p_i)\cup(\cup_{j\in J}t_j))$ of highest PN $S(f)$, is not deducted from the respective formed sets Ci^+, Cp^+, Cf^+, full lowest HPN $S(f)^+$.

Inter-level relations of quasi-order ψ and compatibility ξ are checked and provided for found identifiers from Ci and $(\cup_{i\in I}Ci_i^p)$, $(\cup_{j\in J}Ci_j^t)$, check primitives from Cp and $(\cup_{i\in I}Cp_i^p)$, $(\cup_{j\in J}Cp_j^t)$, fragments from Cf and $(\cup_{i\in I}Cf_i^{p\,\prime})\cup(\cup_{j\in J}Cf_j^{t\,\prime})$, at the same time as the perform of level operations identification α, identity β, and determination γ.

Evolution stage

In hierarchy of interacting evolutionary systems

$$Ce^+=(Ce\cup(\cup_{i\in I}Ce(f)_i^p)\cup(\cup_{j\in J}Ce(f)_j^t)),$$

of a two-level construction of checks, the evolutionary system of the highest level for the system PN $S(f)$ is represented by the five

$$Ce = (W\wedge, Cp, Cl, Sg, W^f),$$

[40], the sets of lowest level evolution systems for components PSN$(\cup_{i\in I}S(f)_i^p)\cup(\cup_{j\in J}S(f)_j^t)$ are represented by $(\cup_{i\in I}Ce(f)_i^p)\cup(\cup_{j\in J}Ce(f)_j^t)$ with fives similar to the highest one, like

$$Ce_i^p = (W_i^p\wedge, Cp_i^p, Cl_i^p, Sg_i^p, W_i^{pf}),$$
$$Ce_j^t = (W_j^{t\wedge}, Cp_j^t, Cl_j^t, Sg_j^t, W_j^{tf}).$$

The hierarchical model of the behavioral online testing icS transmits hierarchies of interacting evolutionary systems Ce^+ to the required translation (detail down and generalizations up), using the hierarchical mappings

$$Sg_{iS}= \{\chi^{\rightarrow p}, \chi^{p\rightarrow}, \nu^{\rightarrow t}, \nu^{t\rightarrow}\},$$

and synchronizing relationship of quasi-order ψ^+ and compatibility ξ^+. At the basic level, the $S(f)^+$ is a transfer for subject entities from the senior PN $S(f)$ and the junior PSN $\cup_{i\in I}S(f)_i^p$, $\cup_{j\in J}S(f)_j^t$ (their alphabets of positions, transitions, chips, inputs and output reactions, subject terms, events, actions, functions and their relationships) as part of $S(f)^+$. At the behavioral check level as part of the hierarchical model of behavioral online testing icS, the transmission is performed for translation in hierarchical mapping of check entities (identifiers Ci^+, primitives Cp^+ and fragments Cf^+ with their non-destructive.

Additionally, in the interaction between the highest Ce and lowest evolutions $\cup_{i\in I}Ce(f)_i^p$, $\cup_{j\in J}Ce(f)_j^t$, both the intrinsic and the new, the inter-level parametric functional dependencies φ^+ and σ^+ are adopted, respectively, for fitness functions and choice functions, highest φ, σ and lowest $\cup_{i\in I}\varphi_i^p$, $\cup_{j\in J}\varphi_j$, $\cup_{i\in I}\sigma_i^p$, $\cup_{j\in J}\sigma_j^t$, having a similar appearance:

$$\varphi^+ = \varphi(S(f)) - ((\Sigma_{i\in I}\varphi_{pi}{}^p)+(\Sigma_{j\in J}\varphi_{pj}{}^t))+((\Sigma_{i\in I}\varphi_{S(f)i}{}^p)+ +(\Sigma_{j\in J}\varphi_{S(f)j}{}^t)),$$
$$\sigma^+ = \sigma(S(f)) - ((\Sigma_{i\in I}\sigma_{pi}{}^p)+(\Sigma_{j\in J}\sigma_{pj}{}^t))+((\Sigma_{i\in I}\sigma_{S(f)i}{}^p)+ +(\Sigma_{j\in J}\sigma_{S(f)j}{}^t)).$$

In the functions φ, σ and $\cup_{i\in I}\varphi_i^p$, $\cup_{j\in J}\varphi_j^t$, $\cup_{i\in I}\sigma_i^p$, $\cup_{j\in J}\sigma_j^t$ the check parameters, external in relation to check evolutions Ce and $\cup_{i\in I}Ce(f)_i^p$, $\cup_{j\in J}Ce(f)_j^t$, can be determined by parameters independent of these evolutions itself.

Check properties determination

In the simplest case the checked properties Pr, $\cup_{i\in I} Pr_i^p$, $\cup_{j\in J} Pr_j^t$ are defined in the formals model of behavioral online testing cS, icS, as primitive fragments of reference behavior - weighted, in general, bipartite multiple-arcs, that look like (p, i, t) or (t, i, p), where i is multiple, of relationship the adjacent F, $\cup_{i\in I} F_i^p$, $\cup_{j\in J} F_j^t$ and compliance S, $\cup_{i\in I} S_i^p$, $\cup_{j\in J} S_j^t$ for conditions X, $\cup_{i\in I} X_i^p$, $\cup_{j\in J} X_j^t$, reactions Y, $\cup_{i\in I} Y_i^p$, $\cup_{j\in J} Y_j^t$, interval In, $\cup_{i\in I} In_i^p$, $\cup_{j\in J} In_j^t$, probabilities Pb, $\cup_{i\in I} Pb_i^p$, $\cup_{j\in J} Pb_j^t$, that mark positions P, $\cup_{i\in I} P_i^p$, $\cup_{j\in J} P_j^t$ and transitions T, $\cup_{i\in I} T_i^p$, $\cup_{j\in J} T_j^t$, incidental these arcs. When checking for each arc (p, i, t) or (t, i, p) of reference behavior, position p and transition t are defined as supporting, identified. The reference checked properties Pr, $\cup_{i\in I} Pr_i^p$, $\cup_{j\in J} Pr_j^t$ determine in advance in relation to performance of online testing.

In a more general case, the verifiable properties Pr, $\cup_{i\in I} Pr_i^p$, $\cup_{j\in J} Pr_j^t$ of the check model may be indivisible (set by external conditions), weighted sequences – linear fragments (chains) Frl, $\cup_{i\in I}Frl_i^{p\,\prime}$, $\cup_{j\in J}Frl_j^{t\,\prime}$ of reference behavior, that look like

$$Frl=((p_1, i_1, t_1),(t_1, i_2, p_2),\ldots, (p_{j-1}, i_j, t_j),\ldots),$$

or

$$Frl=((t_1, i_1, p_1),(p_2, i_2, t_2),\ldots, (t_{j-1}, i_j, p_j),\ldots),$$

in terms of sequences, consisting of multiple-arcs, that look like (p, i, t) or (t, i, p) according to corresponding of their internal adjacent F and incidents of common positions p_j or transitions t_j. In this case, when checking for each chain of reference verifiable properties, the supporting position and transition are its initial (start) p_1 or t_1 and the final (finish) p_k or t_k position and/or transition, where - k the length of the chain.

In the most general case, the checked properties Pr, $\cup_{i\in I} Pr_i^p$, $\cup_{j\in J} Pr_j^t$ of the check model cS, icS may be indivisible (set by external conditions), weighted, non-linear fragments (trees, hammocks, networks, cycles and their compositions) Frn, $\cup_{i\in I}Frn_i^p$, $\cup_{j\in J}Frn_j^t$ of reference behaviours in terms of related bipartite position-transition sequences, according to the corresponding of their internal and external (between different sequences) reference adjacent F to them incidental node positions/transitions p_i, t_i (including, start and finish p_1, t_1). When checking some node positions/transitions p_i, t_i (particular, start and finish p_1, t_1) of reference non-linear verifiable properties Frn, $\cup_{i\in I}Frn_i^p$, $\cup_{j\in J}Frn_j^t$ must be support, that is, previously recognized identified, using identifiers Ci, $\cup_{i\in I} Ci_i^p$, $\cup_{j\in J} Ci_j^t$.

In the check analysis, the verifiable properties Pr, $\cup_{i\in I} Pr_i^p$, $\cup_{j\in J} Pr_j^t$ taking into account the supporting positions/transitions p_i, t_i, weighted conditions, events, actions and functions, in particular, the input conditions (in alphabet X) and their structures, as well as output functions (in alphabet Y) and their structures. That is, verifiable properties should be understood as dynamic behavior structures.

Weight characteristics of fitness and selection

Each position p_i or transition t_i has two sets of incidental incoming arcs $In=\cup_{i\in I}\, in_i$ and outgoing arcs $Out=\cup_{i\in I}\, out_i$, possibly multiples k_{in}, k_{out}, as well as the corresponding two sets of inputs and outputs also taking into account the multiple of the arcs. The position's or transition's own high boundary weight can be measured by the sum of numbers of incoming arcs $\Sigma In_k=\Sigma_{i\in I}(in_i\times k_{ini})$ and numbers of outgoing arcs $\Sigma Out_k=\Sigma_{i\in I}(out_i\times k_{outi})$, based on their multiple k_{in}, k_{out}, or orderly deuce of numbers of incoming and outgoing arcs $(\Sigma In_k, \Sigma Out_k)$, that are also based on their multiplicity.

As noted, in the PN $S(f)$ positions p_i and transitions t_i are bound by arcs in structures - one-step elementary arcs with incidental positions/transition and multi-step chains, trees, hammocks, networks, loops and their combinations. Positions p_i and transitions t_i in such structures are nodes. Then some of the registered behavior of W or Cf (in case registration, check with identification) is activated substructures of executed processes in the PN $S(f)$. In these structures, there are still unrecognized positions and transitions, clearly identified at the current moment positions p_i and transitions t_i. For identified positions p_i and transitions t_i, their activated inputs and outputs, incidents recognized arcs, chains, fragments are marked.

In this regard, the current achieved weight of coverage of some position p_i or transition t_i is their numbers of activated incoming and outgoing arcs of $in_i\times k_{ini}$ and $out_i\times k_{outi}$, taking into account their multiple k_{ini}, k_{outi}, their sum

$$wt_{pi}=wt_{pi}(in_i\times k_{ini})+wt_{pi}(out_i\times k_{outi}),$$
$$wt_{ti}=wt_{ti}(in_i\times k_{ini})+wt_{ti}(out_i\times k_{outi}),$$

or an orderly deuce

$$\underline{wt}_{pi}=(wt_{pi}(in_i\times k_{ini}),\ wt_{pi}(out_i\times k_{outi})),$$
$$\underline{wt}_{ti}=(wt_{ti}(in_i\times k_{ini}),\ wt_{ti}(out_i\times k_{outi})),$$

if you need a separate analysis of inputs and exits.

Accordingly, for the elementary verifiable properties Pra, $\cup_{i\in I}\, Pra_i^p$, $\cup_{j\in J}\, Pra_j^t$ for each of the weighted arcs with the incidental double pair “position-transition” (p_i, t_i) or “transition-position” (t_i, p_i) can enter the first characteristic – weight Wt of check cover.

Thus, using the evolutionary functions of fitness φ and the choice σ the weight of the check coating will be determined for a pair of weights - weight of fitness and choice $wt(p_i, t_i)$, $wt(t_i, p_i)$, corresponding to the pair “position-transition” (p_i, t_i) or “transition-position” (t_i, p_i), which are incidentally selected arc i, as the sum (scalar) of these weights

$$wt(p_i,t_i)=wt_{pi}(out_i\times k_{ini})+wt_{ti}(in_i\times k_{ini}),$$

or

$$wt(t_i,p_i)=wt_{ti}(out_i\times k_{ini})+wt_{pi}(in_i\times k_{ini}),$$

or deuce (vector)

$$wt^{\rightarrow}(p_i,t_i)=(wt_{pi}(out_i\times k_{outi}),wt_{ti}(in_i\times k_{ini})),$$

or

$$wt^{\rightarrow}(t_i,p_i)=(wt_{ti}(out_i\times k_{outi}),wt_{pi}(in_i\times k_{ini})),$$

multiplied by the $k_{outi}=k_{ini}$ of the arc i (the number of moved labels, when the arc i is triggered).

Not the elementary verifiable properties Pr, $\cup_{i\in I}\, Pr_i^p$, $\cup_{j\in J}\, Pr_j^t$ (for each of the weighted linear or non-linear fragments) have the weight of coverage for the evolutionary functions of fitness φ and the choice σ, determined by the sum of weights of fitness φ or choice σ functions of all arcs, included in the appropriate weighted linear Frl, $\cup_{i\in I}Frl_i^p$, $\cup_{j\in J}Frl_j^t$ or not linear Frn, $\cup_{i\in I}Frn_i^p$, $\cup_{j\in J}Frn_j^t$ fragments. So for the chain:

$$Frl=((p_1,\ i_1,\ t_1),(t_1,\ i_2,\ p_2),\dots,\ (p_{j-1},\ i_j,\ t_j),\dots,$$
$$(p_{n-1},i_{n-1},t_{n-1}),(t_{n-1},i_n,p_n),)$$

the weight looks like:

$$\begin{gathered}
wt(Frl)=(wt^p{}_1(out_1\times k_{out1})+wt^t{}_1(in_1\times k_{in1}))+\\
+(wt^t{}_1(out_1\times k_{out1})+wt^p{}_2(in_2\times k_{in2}))+\dots+\\
+(wt^p{}_i(out_i\times k_{outi})+\ +wt^t{}_i(in_i\times k_{ini}))+\dots+\\
+(wt^p{}_n(out_n\times k_{outn})+wt^t{}_{n-1}(in_{n-1}\times k_{inn-1}))+\\
+(wt^t{}_{n-1}(out_{n-1}\times k_{outn-1})+\ +wt^p{}_n(in_n\times k_{inn}))=\\
=\Sigma_{i=1}{}^{n-1}(wt^p{}_i(out_i\times k_{outi})+wt^t{}_i(in_i\times k_{ini}))+\\
+(wt^t{}_i(out_i\times k_{outi})+wt^p{}_{i+1}(in_i\times k_{ini+1})).
\end{gathered}$$

Thus, the weights $wt^p{}_i$ and $wt^t{}_i$ of internal in sequence – linear fragment from *Frl*, $\cup_{i\in I}Frl_i^p$, $\cup_{j\in J}Frl_j^t$ (not starting and not finishing for the sequence) positions p_j and t_j transitions are counted twice in mutually related arcs - respectively, (t_{j-1}, i_{j-1}, p_j) (p_j, i_j, t_j) and $(p_j, i_j, t_j),(t_j, i_{j+1}, p_{j+1})$ - separately for the incoming (to them) arcs and outgoing (from them) arcs into account their multiple. That is,

$$wt_{pi}=wt_{pi}(in_i\times k_{ini})+wt_{pi}(out_i\times k_{outi}),$$
$$wt_{ti}=wt_{ti}(in_i\times k_{ini})+wt_{ti}(out_i\times k_{outi}),$$

what was noted above.

The weight of each node position p_j and transition t_j of some non-linear fragment from *Frn*, $\cup_{i\in I}Frn_i^p$, $\cup_{j\in J}Frn_j^t$, common in incidentality for set of mutually related line parts

$$frl_1=((t_{11}, i_{11}, p_{11}),(p_{12}, i_{12}, t_{12}),\ldots,$$
$$(t_{1j-1}, i_{1j}, p_{1j}),\ldots),$$
$$frl_2=((t_{21}, i_{21}, p_{21}),(p_{22}, i_{22}, t_{22}),\ldots,$$
$$(t_{2j-1}, i_{2j}, p_{2j}),\ldots),\ldots,$$
$$frl_j=((t_{m1}, i_{m1}, p_{m1}),(p_{m2}, i_{m2}, t_{m2}),\ldots,$$
$$(t_{mj-1}, i_{mj}, p_{mj}),\ldots),$$

of this non-linear fragment is also considered to be multiples *m* of these line parts, incidental to this node position p_j and transition t_j, and on the multiplicity ki_{11}, ki_{21},…, ki_{m1} of initial arcs i_{11}, i_{21},…, i_{m1} of these linear parts frl_1, frl_2,…, frl_j directly incidental to node position p_j and transition t_j,.

This weight gain, taking into account inputs and outputs, allows to accentuate precisely the arcs, line claims and non line fragments, as elements of behavioral (dynamic) verifiable properties. The best indicator is a higher weight of check coverage.

The second characteristic of each of the elementary and not elementary verifiable properties may be its total length, conventionally measured by the number of arcs, contained into it. For the second characteristic, the best indicator is the shorter length of the check property, which conditionally represents the lower cost of testing.

The weight and length of the properties being tested are at odds, but the latter changes, when the properties being tested are recognized by the supporting positions or transitions (start, finish, internal) using identifiers in check primitives or fragments. In this case, the completeness of the primitive is the completeness of the verifiable property, that entered it, and the length is the total length of the verifiable property and identifiers entered into the primitive, with multiple accounting of their intersecting (in the behavior) parts.

A check snippet from a primitive may differ in the inclusion of many primitives – verifiable properties and their respective identifiers. Its fullness and length are determined by the corresponding amounts.

Thus, check primitives and fragments inherit from the verifiable properties the characteristics of the weight and length of the check coating, considered for both individual primitives and fragments, and for their certain aggregate.

The total weight of all the properties in the specified and their total length give absolute boundary characteristics (respectively top for weight and bottom for length) to assess the quality of check.

The weight and length of the current set of fragments recorded and formed at some point in the check can be absolute current characteristics of the weight and length respectively for the achieved quality of check.

Current characteristics of weight and length can be presented in relative form (including percentage) as private from dividing absolute current characteristics of weight and length by absolute boundary characteristics – the top for weight and lower for length.

Shortness is the third characteristic of a given set of all (elementary and not elementary) properties. Shortness is also a characteristic of the totality of the check primitives (or fragments) used in some coverage of the properties being checked.

Applying characteristics to check

Collectively, such used check primitives may be, for example, initially selected to perform check, such as ensuring its required completeness, or, for example, achieved at some point in the current executable check. The shortness determines the number of properties in the first set and primitives in the second set accordingly.

Both elementary and non-elementary (linear and non-linear) verifiable properties can receive in a certain way the weight and length of the individual additional weight factor, which can be set by external factors.

This ratio can relate: first, as indivisible to the entire property being tested; second, as indivisible in replicated form to property components - arcs, positions, transitions; third, in a distilled form of some additive decomposition to its components. The third option is of interest in determining the mapping of weight and length at end-to-end, ascending or downward checks.

With downward check, the ratio for check behavior of the detailing Petri subnet (multiple check fragments) detailing some macro-position or macro-transition may be the weight of fitness functions or the choice of this macro-position or macro-transition, represented in the form of additive decomposition in accordance with the check passages through the macro-position or macro-transition. Macro-position or macro-transition passages are understood as its activated three inputs-outs, respectively, the decomposition of the weight factor is presented for each three and within it for its entry-exit function.

The parts of this decomposition of the check pass weight factor are mapping in dish weights, taking into account the mapping of Petri input of hierarchical Petri network, in particular from the ins and outs of the macro-position or macro-transition to the respective ins and outs of its detailed Petri subnet.

Thus, in the check fragment of the Petri subnet, as in the passage of the macro-position or macro-transition, also emphasizes the three – the input-fragment-output.

With upward check of the external ratio of the detailed macro-position or macro-transition of the highest Petri network, there may be an additive weight of functions fitness or selection, defined as the amount of weight factors for the check fragments of the detailed Petri subnet, taking into account the past's inputs and outputs in the respective ins and outs of the macro-position or macro-transition of the lowest Petri network.

That is, the weight factors of the inputs and outputs of the Petri subnet check fragments are mapped in the corresponding weight factors of the inputs and outputs of the activated macro-position or macro-transition passes – three inputs-functions-outputs – and is summarized in components in case of repetition.

Depending on the accepted properties, three definitions of check 1-primitives are possible: first, as weighted arcs of the relationship between positions and transitions of the input Petri network, for which the start and/or finishing (border) positions/transitions are confirmed using identifiers; Second, as weighted linear fragments are also with identifiers confirmed by start and/or finishing positions/transitions; thirdly, as weighted non-linear fragments are also with identifiers confirmed by start, internal, finishing positions/transitions.

The simplest check 0-primitives are the positions / transition identifiers themselves, which highlight confirmed positions/transitions, that can be start, finishing, and internal (not-border) in identifiers. In turn, the check primitives themselves are the indivisible simplest check fragments of behavior that are dedicated to testing.

The multilevel method of behavioral check

In the multilevel method, which has signs of a "wave", multi-process and evolutionary, the following items are performed:

Stage 1. The initial highest checked PN $S(f)$, the mappings $\chi^{\to p}$, $\chi^{p\to}$ of its selected positions, the mappings $\nu^{\to t}$, $\nu^{t\to}$ of its selected transitions $\nu^{\to}$, and the set of initial lowest checked PSN from $(\cup_{i\in I}S(f)_i^{p})\cup(\cup_{j\in J}S(f)_j^{t})$ are accepted as the current highest checked PN $S(f)^{\wedge'}$, current mapping $\chi^{\to p}$, $\chi^{p\to}$ of its selected positions, current mapping $\nu^{\to t}$, $\nu^{t\to}$ of its selected transitions and current sets of initial junior checked PSN of $(\cup_{i\in I}S(f)_i^{p\wedge'})\cup(\cup_{j\in J}S(f)_j^{t\wedge'})$.

Stage 2. For the current highest verifiable PN $S(f)^{\wedge'}$ (some PSN from the *HSN*, accepted as the senior one) starts and executes a multithreaded process of the one-level behavioral evolutional check method, that is presented at [40].

Step 2.1. For the current mappings $\chi^{\to p}$, $\chi^{p\to}$ and $\nu^{\to t}$, $\nu^{t\to}$ of selected positions $(\cup_{i\in I}p_i)$, transitions $(\cup_{j\in J}t_j)$ of the highest checked PN $S(f)^{\wedge}$ and the set of corresponding lowest ones checked PSN from $(\cup_{i\in I}S(f)_i^{p})\cup(\cup_{j\in J}S(f)_j^{t})$ starts and runs $i+j$ multithreading processes, where $i\in I$, $j\in J$ corresponding to the lowest checked PSN of the single-level behavioral evolutional check method [40].

Step 2.2. With the start of execution of each of the $i+j$ processes in its composition, the corresponding special event synchronization relationships compatibility $\xi^{+}=\xi\cup(\cup_{i\in I}\xi_i)\cup(\cup_{j\in J}\xi_j)$ and quasi-order $\psi^{+}=\psi\cup(\cup_{i\in I}\psi_i)\cup(\cup_{j\in J}\psi_j)$ for time intervals, associated with them alphabets of conditions, actual events for *iS*, actions, actual functions associated with the positions P, P_i^{p}, P_i^{t}, transitions T, T_i^{p}, T_i^{t}, and also corresponding to these conditions and functions of input-output symbols X, X_i^{p}, X_i^{t}, Y, Y_i^{p}, Y_i^{t} for $S(f)^{+}= =S(f)\backslash((\cup_{i\in I}p_i)\cup(\cup_{j\in J}t_j))\cup(\cup_{i\in I}S(f)_i^{p})\cup(\cup_{j\in J}S(f)_j^{t})$, representing with highest and lowest levels of hierarchical synchronization conduct transition to $S(f)^{+}$.

Step 2.3. For the current set of registered check fragments $W^{\wedge}$ and corresponding current recognized check fragments $Cf^{\wedge}$ the fourth parallel process executes event-driven (as they are registered), parallel, all possible at this moment, conversion threads of operations from the signature $Sg_1=\{\alpha, \beta, \gamma\}$ (identifications α of reference positions and transitions, identity β of reference positions and transitions, determining γ after identity β) in several iterations, deterministic for α, β and in the general case, probabilistic for γ, until the moment, when their use does not give new fragments awn from $W^{\wedge}$ for $Cf^{\wedge}$.

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 PN *S(f)*, complexity of deterministic method.

Evaluation of behavioral testing of DIS

Representation of the Petri network *S(f)*, where $|P|=n_p$, $|T|=n_t$, $n= n_p+n_t$, $|X|=m$, $|Y|=l$, in the memory of the monitoring system using list structures requires for the upper limit of the total number of conditional fields:

$$\begin{gathered} c_{iS}^{max}=(n_t(4n_p+3L+4)+n_p(3m+2))+ \\ +(\Sigma_{i\in I}\, n_{t\,i}(4n_{pi}+3L_i+4)+n_{pi}(3m_i+2))+ \\ +(\Sigma_{j\in J}\, n_{tj}(4n_{pj}+3L_j+4)+n_{pj}(3m_j+2)). \end{gathered}$$

The complexity of the check analysis of the PN *S(f)*, including preprocessor and main stages for each of evolutions, is determined by upper bound:

$$\begin{gathered} c_{ciS}^{max}=n_t(4n_p+3L+4)+n_p(3m+2)+ \\ +2n_tn_p(n_t-1)+2(2Lmn_pn_t)^{nt}-3)+(n_t-1)(n_pn_t)!+ \\ +(\Sigma_{i\in I}\, n_t(4n_{pi}+3L_i+4)+n_{pi}(3m_i+2)+ \\ +2n_{ti}n_{pi}(n_{ti}-1)+2(2L_im_in_{pi}n_{ti})^{niti}-3)+(n_{ti}-1)(n_{pi}n_{ti})!)+ \\ +(\Sigma_{j\in J}\, n_{tj}n_{tj}(4n_{pj}+3L_j+4)+n_{pj}(3m_j+2)+ \\ +2n_{tj}n_{pj}(n_{tj}-1)+2(2L_jm_jn_{pj}n_{tj})^{njt\,j}-3)+(n_{tj}-1)(n_{pj}n_{tj})!). \end{gathered}$$

In both formulas, there are three parts, respectively, for PN and PSN. The comparison of the testing programs based on automata-deterministic and Petri-evolutionary methods for dynamic DIS of onboard automated control systems and terminal video surveillance confirmed a decrease in the computational complexity of testing and reducing the time of the check.

Conclusions. The paper presents the results of the development of the method of behavioral testing of distributed information systems based on a special model of behavioral check of extended hierarchical Petri nets and characterized by the features of the evolutionary parallelism of the check analysis. A model is based on determining the compliance of the reference and verifiable extended Petri nets, representing respectively the reference and verifiable DIS. The use of the model made it possible to determine the basic conditions for constructing a check method applied both at the system level and at the component level. The conditions are significant in the organization of the parallel event-evolutionary development of the population of recognition of the behavior of the reference Petri net in the functioning of the tested Petri net. Parallelism is applied to parallel execution of and cross-over operations with event-generated operands-individuals — recognizable reference check primitives and fragments. Parallelization associated with the time decomposition of the task of the check analysis allows polynomially reducing the exponential time complexity of the overall behavioral testing. The proposed estimates of the completeness of check used in the functions of fitness and search can reduce the time of a multi-level evolutionary search for coverage of checks.

Hierarchical decomposition and the evolutionary approach increase the flexibility of organizing behavioral online testing by taking into account the features of DIS. The greatest reduction was achieved on the components of the DIS of special behavior, in particular, with a partial certainty of the functions of the models.

References

1. Coulouris, George, Dollimore, Jean Kindberg & Tim, Blair Gordon. (2011). “Distributed Systems: Concepts and Design, 5th ed.”, *Publ. Boston: Addison-Wesley,* 1067 p.
2. Bakker, A. A., Steen, M. & Tanenbaum, A. S. (2006). “Wide-Area Distribution Network for Free Software”, *ACM Trans. on Internet Technology,* Vol. 6, pp. 259-281.
3. Kharchenko, V., Gorbenko, A., Sklyar, V. & Phillips, C. (2013). “Green Computing and Communications in Critical Application Domains: Challenges and Solutions”, *IX International Conference of Digital Technologies,* Zhilina: Slovak Republic, pp. 191-197.
4. Pomorova, O., Savenko, O., Lysenko, S. & Kryshchuk, A. (2013). “Multi-Agent Based Approach for Botnet Detection in a Corporate Area Network Using Fuzzy Logic”, in: Kwiecien, A., Gaj, P., Stera, P. (eds.). *20th Conference on Computer Networks, CN*, Lwówek Slaski: Poland. *Communications in Computer and Information Science,* Vol. 370, *Publ. Springer-Verlag,* Berlin Heidelberg: pp. 146-156.
5. Hahanov, Vladimir, Litvinova, Eugenia & Chumachenko, Svetlana. (2017). “Cyber Physical Computing for IoT-driven Services”, *Publ. Springer* 279 p.
6. Maevsky, D., Bojko, A., Maevskaya, E., Vinakov, O., & Shapa, L. (2017). “Internet of Things: Hierarchy of Smart Systems,” *The 9th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications*, Bucharest: Romania, pp. 821-827. DOI: 10.1109/IDAACS.2017.8095202.
7. Stuart Russell J., Norvig, Peter. (2010). “Artificial Intelligence: a Modern Approach”, *Prentice-Hall, Inc. A Simon & Schuster Company Englewood Cliffs,* New Jersey: 1095 p.

8. Poli, Riccardo Langdon, William B., McPhee, Nicholas F. & Koza, John R. (2008). "A Field Guide to Genetic Programming", *Creative Commons Attribution-Noncommercial-No Derivative Works 2.0,* 250 p.

9. Atamanyuk, Igor, Kondratenko, Yuriy, & Shebanin, Vyacheslav. (2016). "Calculation Methods of the Prognostication of the Computer Systems State under Different Level of Information Uncertainty", *Proceedings of the 12th International Conference on Information and Communication Technologies in Education, Research, and Industrial Application. Integration, Harmonization and Knowledge Transfer,* 21-24 (June, 2016), Kyiv: Ukraine, Ermolayev, V. et al. (Eds), ICTERI'2016, CEUR-WS, Vol. 1614, pp. 292-307.

10. Opanasenko, V. & Kryvyi, S. (2016). "Synthesis of Multilevel Structures with Multiple Outputs", *CEUR Workshop Proc. of the 10th International Conference of Programming, UkrPROG,* Kyiv: Ukraine, (24 May 2016), Vol. 1631, Code 122904, pp. 32-37.

11. Atamanyuk, I. & Kondratenko, Y. (2015). "Computer's Analysis Method and Reliability Assessment of fault-tolerance Operation of Information Systems", *Intern. Conf. ICTERI-2015, S. Batsakis, et al., Eds.,* Vol. 1356, (May 2015), pp. 507-522.

12. Komar, M., Sachenko, A., Kochan, V. & Ababii, V. (2016). "Improving the Security of Intrusion Detection System", *Proc. of 13th International Conference on Development and Application Systems,* Suceava: Romania, pp. 19-21.

13. Norouzi1 Monire, Parsa Saeed & Mahjur Ali. (2014). "A New Approach for Formal Behavioral Modeling of Protection Services in Antivirus Systems", *International Journal in Foundations of Computer Science & Technology (IJFCST),* Vol. 4, No. 3, (May 2014), pp.77-85. DOI:10.5121/ijfcst.2014.4307.

14. Hahanov, V., Litvinova, E., Obrizan, V. & Gharibi, W. (2008). "Embedded Method of SoC Diagnosis", *Elektronika in Elektrotechn,* No 8, pp. 3-8.

15. Drozd, A., Antoshchuk, S., Rucinski, A. & Martinuk, A. (2008). "Parity Prediction Method for on-line Testing A Barrel-shifter", *Proc. IEEE East-West Design & Test Symposium,* Lviv: Ukraine, pp. 208-213. DOI: 10.1109/EWDTS.2008.5580162.

16. Romankevich, V. A. (2017). "Self-testing of Multiprocessor Systems with Regular Diagnostic Connections", *Automation and Remote Control*. Vol. 78, Issue 2, pp. 289-299.

17. Drozd, A., Antoshchuk, S., Martinuk, A. & Drozd, J. (2010). "Increase in Reliability of on-line Testing Methods using Natural Time Redundancy", *Proc. IEEE East-West Design & Test Symposium,* Moscow: Russia, pp. 223-229.

18. K¨astner, C., von Rhein, A., Erdweg, S., Pusch, J., Apel, S., Rendel, T. & Ostermann, K. (2012). "Toward Variability-aware Testing", *In PFOSD '12,* ACM, New York: NY, USA, pp. 1-8.

19. IEC 61508-1:2010. (2010). "Functional Safety of Electrical/Electronic/Programmable Electronic Safety Related Systems – Part 1: General requirements". Geneva: *International Electrotechnical Commission.*

20. Gomes, Luís Fernandes & João, M. (2010). "Behavioral Modeling for Embedded Systems and Technologies: Applications for Design and Implementation", 494 p. DOI: 10.4018/978-1-60566-750-8.

21. "Automated Software Testing Using Metahurestic Technique Based on An Ant Colony Optimization" Praveen Ranjan Srivastava, Computer Science and Information System Group, Birla Institute of Technology and Science, (BITS), Pilani Rajasthan, India, 333031 praveenrsrivastava@gmail.com; Km Baby, Network System Software Technology Group, Cisco Systems, Bangalore India-560 087 bprajapa@cisco.com.This paper appears in: Electronic System Design (ISED), 2010 International Symposium on Issue Date: 20-22 Dec. 2010 on page(s): 235-240. Location: Bhubaneswar Print. ISBN: 978-1-4244-8979-4 INSPEC. Accession Number: 11835766. Digital Object Identifier: 10.1109/ISED.2010.52.

22. Grindal, Mats. (2007). "Handling Combinatorial Explosion in Software Testing", *Printed by LiU-Tryck,* Linköping: Sweden, 89 p.

23. Brucker, Achim D. & Wolff, Burkhart (2012)."On Theorem Prover-based Testing", *Formal Aspects of Computing*.25(5), pp. 683-721. CiteSeerX 10.1.1.208.3135. doi:10.1007/s00165-012-0222-y.

24. Schamai, Vladimir (2013). "Model-Based Verification of Dynamic System Behavior against Requirements Method, Language, and Tool", *Publ. Linköping University* SE-581 83 Linköping: Sweden, 257 p.

25. Kramer, A. & Legeard, B. (2016). "Model-Based Testing Essentials – Guide to the ISTQB® Certified Model-Based Tester – Foundation Level", *Publ. John Wiley & Sons,* ISBN 978-1119130017.

26. Baier, Christel & Katoen, Joost-Pieter. (2008). "Principles of Model Checking", *The MIT Press Cambridge*, Massachusetts London, England, 994 p.

27. Classen, A., Cordy, M., Schobbens, P.-Y., Heymans, P., Legay, A. & Raskin, J.-F. (2013). "Featured Transition Systems: Foundations for

Verifying Variability-Intensive Systems and Their Application to LTL Model Checking", *TSE*, 39(8): (Aug. 2013), pp. 1069-1089.

28. Wegener, Joachim, Baresel, Andre & Sthamer, Harmen. (2001). "Evolutionary test Environment for Automatic Structural Testing", *Information and Software Technology* 43, pp. 841-854.

29. Skobtsov, Yu. A. & Skobtsov, V. Yu., Shalyto, A. A. (2015). "Test Generation and Simulation for Crosstalk Faults", *Radio Electronics, Computer Science, Control*, St. Peterburg: Russian Federation, No 4, pp. 72-78. DOI 10.15588/1607-3274-2015-4-11.

30. Skobtsov, Yu. A. & Skobtsov, V. Yu. (2011). "Evolutionary test Generation Methods for Digital Devices", *Design of Digital Systems and Devices* / [eds.: M. Adamski et al.]. Lecture Notes in Electrical Engineering, Vol. 79, Berlin: *Publ. Springer-Verlag*, pp. 331-361.

31. Phadoongsidhi, M. & Scindy Saluja K. K. (2005). "Logic Crosstalk Delay Fault Simulation in Sequential Circuits", *International Conference on VLSI Design*, Kolkata, (3-7 January 2005): Proceedings Los Alamitos: IEEE, pp. 820-823.

32. Sugak, A., Martynyuk, O. & 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*, Warsaw: Poland, pp. 790-795.

33. Martynyuk, O., Sugak, A., Martynyuk, D. & 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: Romania, pp. 888-893. DOI**:** 10.1109/IDAACS.2017.8095215.

34. Kudryavtsev, V. B., Grunskii, I. S. & Kozlovskii, V. A. (2010). "Analysis and Synthesis of Abstract Automata", *Journal of Mathematical Sciences*, (September 2010), Vol. 169, Issue 4, pp. 481-532.

35. Zaitsev, D. A. (2013). "Toward the Minimal Universal Petri Net", *IEEE Transactions on Systems, Man, and Cybernetics: Systems*. pp. 1-12.

36. Martynyuk, O., Drozd, O., Ahmesh, Tamim, Bui Van Thuong, Sachenko, A., Mykhailova, H. & Dombrovskyi, M. (2019). "Hierachical 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, (18-20 September 2019), Metz, France, pp. 724-729.

37. Drozd, A., Drozd, M., Martynyuk, O. & Kuznietsov, M. (2017). "Improving of a Circuit Checkability and Trustworthiness of Data Processing Results in LUT-based FPGA Components of Safety-Related Systems", *CEUR Workshop Proceedings*, Vol. 1844, pp. 654-661.

38. Drozd, A., Drozd, J., Antoshchuk, S., Antonyuk, V., Zashcholkin, K., Drozd, M. & Titomir, O. (2017). "Green Experiments with FPGA", in book: *Green IT Engineering: Components, Networks and Systems Implementation*, Kharchenko, V., Kondratenko, Y., Kacprzyk, J., Eds., Vol. 105. Berlin, Heidelberg: *Springer International Publishing*, pp. 219-239. DOI: 10.1007/978-3-319-55595-9_11.

39. Zashcholkin, K. V., Drozd, O. V, Ivanova, O. N. & Sulima, Y. Y. (2019). "Compositional Method of FPGA Program code Integrity Monitoring based on the usage of digital Watermarks", *Scientific Journal Applied Aspects of Information Technology*, Odessa: Ukraine, *Publ. Science and Technical*, Vol. 2, No. 2, pp. 138-152.

Received 31.10.2019
Received after revision 03.11.2019
Accepted 02.12.2019

УДК 004.738:004.94

[1]**Мартинюк, Олександр Миколайович**, канд. техніч. наук, доцент, доцент кафедри комп'ютерних інтелектуальних систем і мереж, E-mail: anmartynyuk@ukr.net, Scopus ID: 57103391900, ORCID: 0000-0003-1461-2000

[1]**Ахмеш**, **Тамем**, аспірант кафедри комп'ютерних інтелектуальних систем і мереж,
E-mail: tamim.nor@yahoo.com, ORCID: 0000-0001-7342-4339

[1]**Тхіонг Буї Ван**, аспірант кафедри комп'ютерних інтелектуальних систем і мереж,
E-mail: govarava@gmail.com, ORCID: 0000-0002-6176-5432

[1]**Дрозд, Олександр Валентинович**, доктор техніч. наук, професор, професор кафедри комп'ютерних інтелектуальних систем і мереж, E-mail: drozd@ukr.net, Scopus ID: 55388226700, ORCID: 0000-0003-2191-6758

[1]Одеський національний політехнічний університет, пр. Шевченка, 1, м. Одеса, Україна, 65044

УДК 004.738:004.94

БАГАТОРІВНЕВИЙ ПОВЕДІНКОВИЙ КОНТРОЛЬ РОЗПОДІЛЕНИХ ІНФОРМАЦІЙНИХ СИСТЕМ

Анотація: *Особливості експлуатації сучасних розподілених інформаційних систем в значній мірі підсилюють вимоги до достовірності їх функціонування і посилюють часові межі її відновлення. Висока вірогідність роботи забезпечується різноманітними засобами, зокрема, робочим і тестовим контролем, але часто обмежується реальним часом роботи цих систем. Відомі методи контролю використовують апаратні прискорювачі, декомпозицію і розпаралелювання, технології штучного інтелекту, але їх можливості не знімають доцільність розвитку нових моделей і методів, зокрема, високорівневих, заснованих на спеціальному аналізі поведінки систем. У статті представлений багаторівневий метод поведінкового робочого контролю розподілених інформаційних систем, що базується на ієрархічних мережах Петрі, однорівневих моделі і методі, а також ієрархічної моделі поведінкового робочого контролю. Багаторівневий метод використовує еволюційну оптимізацію і має особливості паралельної багаторівневої організації сумісних контрольних еволюцій окремих рівнів, які розвиваються за принципом «хвильового» алгоритму в фоновому режимі щодо основного функціонування самих розподілених інформаційних систем. Для визначення комплексних критеріїв контролю багаторівневого методу запропонована ієрархічна система сумісних функцій фітнес і вибору об'єктів в контрольних еволюціях рівнів ієрархічної моделі. У функціях пропонуються оцінки повноти і довжини (часу) перевірки для позицій, переходів і дуг мереж Петрі, лінійних і нелінійних графових структур поведінки, дію функцій розширено на контрольні примітиви і фрагменти. Ієрархія функцій фітнес і вибору дозволила ввести оцінки повноти і довжини робочого контролю, застосовані для оптимізації еволюційного пошуку в багаторівневому покритті необхідних перевірок. Ієрархічна декомпозиція, еволюційний підхід і ієрархія функцій фітнес і вибору багаторівневого методу, що оптимізують, дали можливість скоротити час і підвищити гнучкість організації поведінкового робочого контролю з урахуванням особливостей розподілених інформаційних систем.*

Ключові слова: *розподілені інформаційні системи; поведінковий контроль; ієрархічна мережа Петрі; модель перевірки; еволюція; багаторівневий контроль*

УДК 004.738:004.94

[1]**Мартынюк, Александр Николаевич**, канд. технич. наук, доцент, доцент кафедры компьютерных интеллектуальных систем и сетей, E-mail: anmartynyuk@ukr.net, Scopus ID: 57103391900, ORCID: 0000-0003-1461-2000
[1]**Ахмеш**, **Тамем**, аспирант кафедры компьютерных интеллектуальных систем и сетей, E-mail: lukianov@bsu.by, ORCID: 0000-0001-7342-4339
[1]**Тхионг**, **Буи Ван**, аспирант кафедры компьютерных интеллектуальных систем и сетей, E-mail: govarava@gmail.com, ORCID: 0000-0002-6176-5432
[1]**Дрозд, Александр Валентинович**, д-р технич. наук, профессор, профессор кафедры компьютерных интеллектуальных систем и сетей, E-mail: drozd@ukr.net, Scopus ID: 55388226700, ORCID: 0000-0003-2191-6758
[1] Одесский национальный политехнический университет, пр. Шевченко, 1, г. Одесса, Украина, 65044

МНОГОУРОВНЕВЫЙ ПОВЕДЕНЧЕСКИЙ КОНТРОЛЬ РАСПРЕДЕЛЕННЫХ ИНФОРМАЦИОННЫХ СИСТЕМ

Аннотация. *Особенности эксплуатации современных распределенных информационных систем в значительной степени усиливают требования к достоверности их функционирования и ужесточают временные границы ее восстановления. Высокая достоверность работы обеспечивается разнообразными средствами, в частности, рабочим и тестовым контролем, но часто ограничивается реальным временем работы этих систем. Известные методы контроля используют аппаратные ускорители, декомпозицию и распараллеливание, технологии искусственного интеллекта, но их возможности не снимают целесообразность развития новых моделей и методов, в частности, высокоуровневых, основанных на специальном анализе поведения систем. В статье представлен многоуровневый метод поведенческого рабочего контроля распределенных информационных систем, базирующийся на иерархических сетях Петри, одноуровневых модели и методе, а также иерархической модели поведенческого рабочего контроля. Многоуровневый метод использует эволюционную оптимизацию и обладает особенностями параллельной многоуровневой организации совместимых контрольных эволюций отдельных уровней, которые развиваются по принципу «волнового» алгоритма в фоновом режиме относительно основного функционирования самих распределенных информационных систем. Для определения комплексных критериев контроля многоуровневого метода предложена иерархическая система совместимых функций фитнесс и выбора объектов в контрольных эволюциях уровней иерархической модели. В функциях предлагаются оценки полноты и длины (времени) проверки для позиций, переходов и дуг сетей Петри, линейных и нелинейных графовых структур поведения, действие функций расширено на контрольные примитивы и фрагменты. Иерархия функций фитнесс и выбора позволила ввести оценки полноты и длины рабочего контроля, примененные для оптимизации эволюционного поиска в многоуровневом покрытии необходимых проверок. Иерархическая декомпозиция, эволюционный подход и иерархия* *оптимизирующих функций фитнесс и выбора многоуровневого метода дали возможность сократить время и повысить гибкость организации поведенческого рабочего контроля с учетом особенностей распределенных информационных систем.*

Ключевые слова: *распределенные информационные системы; поведенческое тестирование; иерархическая сеть Петри; модель проверки; эволюция; многоуровневый контроль*

Oleksandr N. Martynyuk, Candidate of Technical Sciences, Associate Professor
Research field: Behavioral testing and diagnosis of computer systems, formal verification and recognizing of digital systems

Tamem Ahmesh, PhD. Student
Research field: testing and diagnosis of computer systems, computer networks

Bui Van Thuong, PhD. Student
Research field: testing and diagnosis of computer systems, “green” computer systems and components

Oleksandr V. Drozd, Doctor of Technical Sciences, Professor
Research field: Testing and diagnosis of computer systems, arithmetical foundations of computer systems, computer systems and components