UDC 004.738:004.94

MODELS AND METHODS OF TEST SYNTHESIS FOR MULTILEVEL NETWORK SYSTEMS, BASED ON PETRI NETS

Al Sharif Waseem,
Martynyuk A. N., *Ph.D. (Engineering), associate professor,*
Nesterenko S. A., *D.Sc. (Engineering), professor*
Odessa National Polytechnic University

Abstract. *The developed models and testing methods of synthesis of tests for components of multilevel network systems, which are based on the identification of and broadcast test component behavior of Petri nets in their composition, have been considered. Synthesis compatibility tests determine acceptable test primitive's component nets in their composition. Implementation method included the development package synthesis tests.*

Keywords*: Petri nets, behavior, identification, implementation, translation, test*

МОДЕЛИ И МЕТОДЫ СИНТЕЗА ТЕСТОВ ДЛЯ МНОГОУРОВНЕВЫХ СЕТЕВЫХ СИСТЕМ НА ОСНОВЕ СЕТЕЙ ПЕТРИ

Аль Шариф Васим,
Мартынюк А. Н., *к.т.н., доцент,*
Нестеренко С. А., *д.т.н., профессор*
Одесский национальный политехнический университет

Анотація. *В роботі розроблені моделі тестування і методи синтезу тестів для компонентів багаторівневих мережних систем, що засновані на ідентифікації, реалізації і трансляції тестової поведінки компонентних мереж Петрі в їх композиціях. Методи синтезу тестів визначають сумісності допустимих тестових примітивів компонентних мереж Петрі в їх композиції. Реалізація методу включила розробку пакету синтезу тестів.*

Ключові слова*: мережа Петрі, поведінка, ідентифікація, реалізація, трансляція, тест*

Introduction. Tool requirements worker and test control and diagnosis of the components of multi-level network systems (MLS) as a way to provide, maintain and restore their health, grow faster than the development itself MLS. Under MLS mean interacting open systems, each of which performs a client and / or server functions, components interact within the MLS for various protocols.

The present level of development of methods of synthesis of tests for MLS components enable verification of systems of high complexity. Development and implementation were simulated, behavioral, functional, structural methods, and hybrid methods based on them. Such combinations, combining the use of automatic models and logic-algebraic models of external behavior, can combine their advantages and Settings os specific MLS.

However, the continuing need for research testing problems due to the increasing use of MLS increasingly complex systems, which present information about the operation of a distributed, usually with a high degree of uncertainty. Relevance of the work due to the need of development of existing and development of new models and methods of synthesis of tests aimed at improving the diagnosis of MLS and increasingly complex, especially in terms of reducing the length of the test and the time of their development. The solution to these problems is to reduce the design and production and operating costs, reduce development time and disaster recovery of MLS.

The study aims to develop a model and a method of synthesis of tests for the diagnosis of MLS components with less build-time tests and their length due to the development of test, identification of extensions of Petri nets.

1. Formal model and method. To achieve this goal it is necessary to solve the problem of testing model components MLS and their compositions, based on the identification, feasibility and broadcast behavior of Petri nets [1] in their compositions, as well as development-processing method for constructing tests for autonomous components MLS and their compositions in the form of tests implemented and broadcast in the compositions of Petri nets.

Problem Statement: Given a composition of extended Petri nets representing the behavior of components of a MLS. Necessary to define the model of the testing process and the method of synthesis of tests for any Petri net in terms of identified [2], implemented and broadcast in the composition of fragments of behavior. Based on the model and the method you want to set the appropriate reference and verifiable Petri nets for any components of the formulation.

Based on the model S(f) is defined SN-composition of $\forall S(f)_h \in S(f)$^, as the object of analysis for the test of interacting components MLS form:

$$SN = (X, Y, S(f)\wedge, a\wedge) \qquad (1)$$

with input and output alphabets of the entire network, multiple component of Petri nets, the set of alphabetic corresponding component of Petri nets defined in step merger of their position in the composition.

Used in the analysis of SN operations network functional composition of Petri nets, performed at the confluence of the positions that involve parallel, based on the function marking M(f), the work component of Petri nets. In their structure – step a) the sequence of the compound $(S(f)_k{}^{@}S(f)_m)$, where the output position of $S(f)_k$ are input positions for $S(f)_m$; b) parallel connection of $S(f)_k{}^{\times}S(f)_m$, when $S(f)_k$ and $S(f)_m$ is the general position of the input; c) compounds with feedback $(S(f)_k{}^{@'}S(f)_m)$, where the output position of $S(f)_k$ are input positions for $S(f)_m$, while some output places $S(f)_m$ is the input positions for $S(f)_k$.

SN defines the conditions implemented and translated into behavior SN arbitrary component $S(f)_h$ of the set S(f)^. Evaluation and execution of these conditions require you to perform the reverse (from inputs $S(f)_h$ to the input SN) and direct (from the outputs $S(f)_h$ to the output SN) model the behavior of S(f)^.

Operation $(S(f)_k{}^{@}S(f)_m)$, $S(f)k^{\times}S(f)_m$, $(S(f)_k{}^{@'}S(f)_m)$ and Structure of $\alpha^{-1}\wedge = \cup_{h\in H}\, \alpha_h{}^{-1}$ и $\alpha\wedge = \cup_{h\in H}\, \alpha_h$ forms: first, the subnet $T^{-1}(S(f)_h)$ of the input $S(f)_h$ to the input SN in reverse progress with matching sets own input words in the realized input $S(f)_h$ number of its own output words, secondly, direct subnet $T(S(f)_h)$ of the output $S(f)_h$ to the output SN in direct promotion of compliance with its own set of input words implemented in the output SN number of its own output words.

For $S(f)_h$ without external inputs for the implementation of its behavior is determined by the input test input set of words R_h in her alphabet $X_h = \alpha_h(X \times Y_{h'})$, realized by $T^{-1}(S(f)_h)$. Similarly, for $S(f)_h$ without external output to achieve its output test behavior is determined by the output set of words R_h' in the alphabet SN $Y_{h'} = \alpha_h(X \times Y_h)$, realized by $T(S(f)_h)$. The basis of the definition of sets R_h and R_h '– operations $(S(f)_k{}^{@}S(f)_m)$, $S(f)_k{}^{\times}S(f)_m$, $(S(f)_k{}^{@'}S(f)_m)$ and Structure $\alpha^{-1}\wedge = \cup_{h\in H}\, \alpha_h{}^{-1}$ and $\alpha\wedge = \cup_{h\in H}\, \alpha_h$ components subnets $T^{-1}(S(f)_h)$ and $T(S(f)_h)$ are defined (minimizing the possible) compliance of the set $X_{T\text{-}1(S(f)h)}*$ input words $T^{-1}(S(f)_h)$ in a variety of output words R_h input $S(f)_h$, and the conformity of the many R_h' output words at the output $S(f)_h$ in a variety of output words subnet $T(S(f)_h))$. The result is minimized automata F_h, F_h' events in the output alphabet backward and forward subnets representing R_h and R_h'.

To determine the loss of output information $S(f)_h$ in $T((S(f)_h)$ to construct a set of words Tr_h in the alphabet Y_h, broadcast subnetwork $T((S(f)_h)$ to the outputs of SN. Underlying the definition of words Tr_h – generalization of $G(AS(f)_h)$ testing graph, known for machines without loss of information, the subgraph $G' \subseteq G(A_{S(f)h})$ which is the basic mechanism for analyzing the detected behavior:

$$G(A_{S(f)h}) = (B(P_h \grave{E} T_h), Y_h\,\acute{}\,X_h\,\acute{}\,(P_h \grave{E} T_h)^2, D_h, (P_h \grave{E} T_h)) \qquad (2)$$

The use of $G(A_{S(f)h})$ and operations $(S(f)_k{}^{@}S(f)_m)$, $(S(f)_k{}^{\times}S(f)_m)$, $(S(f)_k{}^{@'}S(f)_m)$ and Structure of $\alpha\wedge = \cup_{h\in H}\, \alpha_h$ components subnet $T(S(f)_h)$ to determine compliance with the known distribution of the set of input words Tr_h into many output words $Y_{T(S(f)h)}*$ with possible to minimize it. Obviously $Tr_h \subseteq R_h'$.

Testing model for SN – a model TS = (R ^,Tr ^,TS ^) component: a) input sets of words R_h verifiable $S(f)_h \in S(f)$^ of SN, sold $T^{-1}(S(f)_h)$ of the input SN; b) the output of sets of words Tr_h, verifiable $S(f)_h \in S(f)$^ of SN, broadcast $T(S(f)_h)$ to the outputs of SN; c) the number of individual testing models

$TS\^{} = \cup_{h\in H} TS_h$ [3] for verifiable $S(f)_h \in S(f)\^{}$ of SN, selected Relations SN, implemented and broadcast on the border of SN.

To test the model TS SN behavior is defined as a set of systems $\{W_h'\}$ shared, synchronized by common bonds, conditions and events of behavior for $S(f)_h \in S(f)\^{}$, each system consists of a set of words $W_h' = \{w_1, w_2, \dots w_k\}$ in the full alphabet $U_h = (N'\times X_h)^n \times Y_h \times (N'\times X_h)^n$ – (vector sets preconditions)×events× (vector sets postconditions). Pacing, conditions and events suggests the feasibility and broadcast. For a set of words W_h 'front feasibility defined as $W_h'' = \{w'' \in W_h'' | w'' \in W_h \ \&\ pr_1(w'') \cap R_T^{-1}{}_{(S(f)h)} \neq \varnothing\}$, output marketability – as $W_h''' = \{w''' \in W_h''' | w''' \in W_h \ \&\ pr_2(w''') \cap X_{T(S(f)h)*} \neq \varnothing\}$, collapsability – as $W_h'''' = =\{w'''' \in W_h'''' | w'''' \in W_h \ \&\ pr_2(w'''') \cap Tr_{T(S(f)h)} \neq \varnothing\}$. Then the condition of simultaneous feasibility and broadcast is defined as $W_h' = W_h \cap W_h'' \cap W_h'''' \neq \varnothing$.

Collection of sets selected, implemented and broadcast, verifiable properties $\{Pr_h'\}$ – is the set of matches for each $S(f)_h \in S(f)\^{}$ form

$$Pr_h'\,\text{Í}\,F_h:((P_h\,\text{´}\,T_h)\,\text{È}\,(T_h\,\text{´}\,P_h))\,\text{®}\,N\text{¢}, \tag{3}$$

those that are implicitly included as arcs incidence of places and transitions provided conduct W_h'.

In the composition of the original automata $A_{S(f)}\^{} = \cup_{h\in I} A_{S(f)h}$, of the search for identity in the SN and test primitive is the collection of sets respectively identifiers $\{Id_h\}$ and test primitives $\{Ex_h\}$ for $S(f)_h \in S\ (f)\^{}$, considered autonomous.

A set of synchronized common bonds, conditions and events identifiers $\{Id_h'\}$ [2] is defined as the set of fragments of the behavior of each $S(f)_h \in S\ (f)\^{}$ form $Id_h' = \{id_1, id_2, \dots, id_k'\}$ in external alphabet $U_h'' = (N'\times X_h'')^n \times Y \times (N'\times X_h'')^n$ for vectors P_h^n positions P_h standard $S(f)_h$, the input and output with respect to transitions, and a system ratio-making $\{\sigma_h, \eta_h, \tau_h\}$ on the set Id_h ', where σ_h, η_h, τ_h – respectively for compatibility and incompatibility of uncertainty. The condition of the feasibility and broadcast to $\{Id_h'\}$ is defined as

$$"Id_h'\,\text{Î}\,\{Id_h'\}(\$w'\,\text{Î}\,W_h'(\$w_1,w_2\,\text{Î}\,W_h(w'=w_1\#Id_h'\#w_2))). \tag{4}$$

The set of test primitives $\{Ex_h$ '}, synchronized generic references, conditions and events, defined as a set of fragments of behavior for each $S(f)_h \in S(f)\^{}$

$$Ex_h'\,\text{Ì}\,((P_h\,\text{´}\,N\text{¢}\,\text{´}\,X_h)^n\,\text{´}\,(T_h\,\text{´}\,Y_h)\,\text{´}\,(P_h\,\text{´}\,N\text{¢}\,\text{´}\,X_h)^n)\#Id_h', \tag{5}$$

Here, as before, «#» - Hitch operation sets the conditions for the vectors of output places $(P\times N'\times X)^n$ and identifiers of Id_h' with the incidence relations. Conditions occur-duro and broadcast to $\{Ex_h'\}$ is defined

$$"Ex_h'\,\text{Î}\,\{Ex_h'\}(\$w'\,\text{Î}\,W_h'(\$w_1,w_2\,\text{Î}\,W_h(w'=w_1\#Ex_h'\#w_2))). \tag{6}$$

SN testing model also forms the algebraic system

$$ATSN=((R\^{}, Tr\^{}, TS\^{}), \{^{@}, \,^{\text{´}}, ^{@'}, \#\}, \{s, h, t, n\}) \tag{7}$$

in the first, as the support of the input word sets $R\^{}$ sold for verifiable $S(f)_h \in S(f)\^{}$, the output sets of words $Tr\^{}$ translated for verifiable $S(f)_h \in S(f)\^{}$, many individual testing models $TS\^{}$ for verifiable $S(f)_h \in S(f)\^{}$; secondly, of the composition – @ serial, parallel ×, feedback @', hitch#, and thirdly, the relationship compatibility σ, incompatibility η, uncertainty τ and quasi ν.

Network synthesis method tests for SN includes, as part of the base, the local method of constructing tests [3] for the component $S(f)_h \in S(f)\^{}$. The initial test for the network primitives that are possible in SN, is a collection of sets of test primitives $\{Ex_h\}$ for all $S(f)_h \in S(f)\^{}$. In the network method: a) using the method of synthesis of tests autonomous Petri nets $S(f)\^{}$ defined set of identifiers $Id\^{}$ and test primitives $Ex\^{}$; b) on the basis of forward and reverse pass on the structure of the network model SN ties to all of its components (I/O $S(f)_h$) define sets of inverse realizing $T^{-1}(S(f)_h)$ and direct broadcast T $((S(f)_h)$ trees and the sets sold R_h and broadcast Tr_h words, c) based on sets R_h implemented and broadcast Tr_h words and sets of identifiers $Id\^{}$ and test primitives $Ex\^{}$ determined feasible and broadcast identifiers $Id\^{}'$ and test primitives $Ex\^{}'$; g) of the sets sold and broadcast identifier $Id\^{}'$ and test primitives $Ex\^{}'$ for all $S(f)\^{}$ pass through the structure of the network model is formed and implemented a lot of broadcast test $TEx\^{}'$, as psevdoeylers rounds each component Petri $S(f)_h$.

Pacing, conditions and events of test cases primitives $Ex\^{}' = \cup_{h\in H} Ex_h'$ was founded valid on SN verifiable properties $Pr\^{}' = \cup_{h\in H} Pr_h'$, implesized $R\^{}' = \cup_{h\in H} R_h$ 'and broadcast $Tr\^{}' = \cup_{h\in H} Tr_h'$ sets, identifiers, vector sets of positions Id^'= $\cup_{h\in H}$ Id$_h$' in the test subnet SN^'= $\cup_{h\in H}$ SN$_h$'. These combine to form a six network models

$$NPr'=(X, Y, Pr^{\wedge}{}',a^{\wedge}),\ NR^{X}{}' =(X, Y, R^{\wedge}{}', a^{\wedge}),$$
$$NTr^{Y}{}'=(X, Y, Tr^{\wedge}{}',a^{\wedge}),\ NId' =(X, Y, Id^{\wedge}{}', a^{\wedge}),$$
$$NEx'=(X, Y, Ex^{\wedge}{}',\grave{E}_{h\,\hat{I}H}\ a_h),\ NS' =(X, Y, SN^{\wedge}{}', a^{\wedge}) \tag{8}$$

Arbitrary automaton A$_{S(f)h}$ in nonredundant SN for each state k$_h\in$K, represented set position type $((p_1,i_1),(p_2,i_2),...,(p_j,i_j),...,(p_n',i_n'))\in P_h$, has at least one implemented and shown in MTP the ID Id$_{kh}$. Obvious statement.

Statement 1. A necessary and sufficient condition for the construction of a test for A$_{S(f)h}$ in the SN is the existence of at least one implemented and broadcast network identification torus Id$_{ph}$ for each state k$_h$.

NPr', NR', NTr', NId', NEx', NS' are based on extensions of the known conditions for the implementation of automatic inputs X network to the inputs X$_h$ and outputs of the broadcast Y$_h$ to the network output Y. Terms of the sale of fixed networks - NEx' and NS' are represented as follows:

$$''Ex_h'\,\hat{I}\,Ex^{\wedge}{}'((\text{Æ}^{1}Ex_h'\,\acute{I}\,Ex_h)\&(np_1Ex_h'\,\acute{I}$$
$$\acute{I}\,np_1Ex_h\text{Ç}R_h)\&(np_2Ex_h'\,\acute{I}\,np_2Ex_h\text{Ç}Tr_h))$$
$$''S(f)_h'\,\hat{I}\,S(f)^{\wedge}{}'((\text{Æ}^{1}S(f)_h'\,\acute{I}\,S(f)_h)\&(np_1S(f)_h'\,\acute{I}$$
$$\acute{I}\,np_1S(f)_h\text{Ç}R_h)\&(np_2S(f)_h'\,\acute{I}\,np_2{}^{\tilde{}}S(f)_h\ \text{Ç}Tr_h)), \tag{9}$$

where pr1 and pr2 – the first and second components of the projection vectors. Subnets of S(f)^', including-regulated in SN, it retains some of its audited Pr^', realizing R^', translating Tr^', identifying Id ^' properties. Displaying automata statement for generalized Petri nets.

Statement 2. Inclusion of NP'$\subseteq$NP, NR'$\subseteq$NR, NTr'$\subseteq$NTr, NId'$\subseteq$NId, NS'$\subseteq$NS generate inclusion NEx'$\subseteq$NEx.

Performing this analysis determines the development of information technology of synthesis of tests for SN as a network model of a set of six networks – networks of NP', NR', NTr', NId', NS' and test primitives NEx' for them.

The network uses a method of synthesis of tests a) structural analysis of graphs, b) analysis of network matches α^ for component S(f)h, determined by the alphabetical synchronization γ_{na}; c) analysis of the positions of the merger component S(f)$_h$.

2. Implementation of the model and method. In the information technology of synthesis tests are five stages (see Fig. 1), the first stage is defined by constructing a test battery S (f) h, the second to the fifth stage – network method of synthesis of tests for SN.

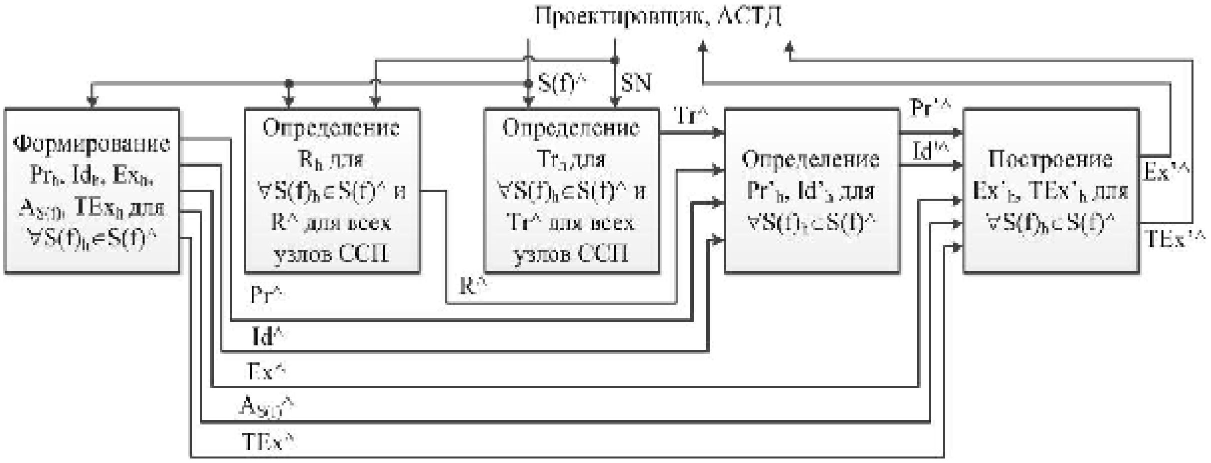
Проектировщик, АСТД
S(f)^
SN
Формирование Pr$_h$, Id$_h$, Ex$_h$, A$_{S(f)}$, TEx$_h$ для ∀S(f)$_h$∈S(f)^
Определение R$_h$ для ∀S(f)$_h$∈S(f)^ и R^ для всех узлов ССП
Определение Tr$_h$ для ∀S(f)$_h$∈S(f)^ и Tr^ для всех узлов ССП
Tr^
Определение Pr'$_h$, Id'$_h$ для ∀S(f)$_h$⊂S(f)^
Pr'^
Id'^
Построение Ex'$_h$, TEx'$_h$ для ∀S(f)$_h$⊂S(f)^
Ex'^
TEx'^
R^
Pr^
Id^
Ex^
A$_{S(f)}$^
TEx^

Figure. 1. Information technology synthesis of tests

The first basic step that contains three steps, defines a set of test primitives Ex^ for arbitrary S(f)$_h$ of SN. In the second phase, which contains three steps, we define sets of input words R^', which implements the SN on the inputs of any S(f)$_h\in$ S(f)^. In the third phase, which contains three steps, we define sets output words Tr^', translated from the output of any $S(f)_h \in S(f)$^ to the outputs Y SN. In the fourth phase, which contains four steps that define sets the property to check Pr^' and identifiers positions Id^' for all $S(f)_h \in S(f)$^. In the fifth stage, which contains three steps, based on the identity Id^' and checked the properties of Pr^' defined network NEx' as many possible in SN test primitives Ex^' and formed many TEx^' network tests.

Software implementation of a package of synthesis of tests carried out in the form of interacting units. Methodological basis for the implementation of the package is an object-component models and programming techniques and fusion tests. The package includes a self-contained units (SP-test synthesis) and network (SSP-test synthesis) models and methods, power network synthesis, interface unit and the unit database.

Conclusion. Testing distributed network systems, exploring the network communication protocol implementations and functionality requires the use and development of effective formal models. Pending test models based on extended Petri nets provide an opportunity to reduce the combinatorial complexity and improve the accuracy and flexibility of analysis.

The proposed model and test method of synthesis of tests suggest an implicit reference implementations, and the model of the single explicit reference model standard (standard) protocol mechanisms for when the restrictions on the power of alphabets implementations. This feature eliminates the need to transfer all implementations, that is, is too much deviation from the standard options.

We study the identification of the structural component of Petri nets (incident conductivity), assuming the continuation of separate analysis functions Petri nets and the entire network of Petri overall.

References

1. Peterson J., Petri net theory and the modeling of system. – M. : "Mir", 1984. – 264 p.
2. Grunsky I. S., Kozlovsky V.A. Synthesis and identification of the automata. – K. : "Naukova dumka", 2004. – 246 p.
3. Al Sharif Waseem Test Models for WiFi mechanisms. Science-Technical Journal "Electro technical and computer systems. Information systems and technology", № 08 (84), 2012, Odessa. – P. 105-109.

Список литературы

1. Питерсон Дж., Теория сетей Петри и моделирование систем. М. : «Мир», 1984. – 264 с.
2. Грунский И. С., Козловский В. А. Синтез и идентификация автоматов. – К. : Издательство «Наукова думка», 2004. – 246 с.
3. Аль Шариф Васим Модели тестирования механизмов Wi-Fi, Научно-технический журнал «Электротехнические и компьютерные системы. Информационные системы и технологии», № 08 (84), 2012, Одесса. – С. 105–109.

Стаття надійшла до редакції 11.04.2013.

Відомості про авторів:

Аль Шариф Васим, Одесский национальный политехнический университет.

Мартынюк А. Н., кандидат технических наук, доцент, Одесский национальный политехнический университет.

Нестеренко С.А., доктор технических наук, профессор, Одесский национальный политехнический университет.