IOP Conf. Series: Materials Science and Engineering **124** (2016) 012105 doi:10.1088/1757-899X/124/1/012105

Principles of E-network modelling of heterogeneous systems

D Tarakanov[1], I Tsapko[2], S Tsapko[2] and R Buldygin[2]
[1] Polytechnic Institute, Surgut State University, 1, Lenina ave., Surgut, 628400, Russia
[2] Tomsk Polytechnic University, 30, Lenina ave., Tomsk, 634050, Russia

E-mail: Tsiv@tpu.ru

Abstract. The present article is concerned with the analytical and simulation modelling of heterogeneous technical systems using E-network mathematical apparatus (the expansion of Petri nets). The distinguishing feature of the given system is the presence of the moduleб which identifies the parameters of the controlled object as well as the external environment.

1. Introduction
When modeling complex heterogeneous systems it is important to choose a scheme (basis) for analytical and simulation modeling [1]. The present paper examines the approach, which allows creating discrete and continuous models for a wide range of processes and systems using uniform methodological positions. It also enables modeling the reliability of particular elements and the whole system.

As the mathematical basis for constructing a computer model, we suggest using E-network apparatus, which expands Petri nets [1]. Due to their properties [2, 3] E-networks meet the initial requirements for being the mathematical basis for constructing complex technical systems, which have structural and parametric configuration changes, numerous elements functioning independently of each other, random processes, discrete and continuous components, etc.

2. Heterogeneous system model architecture
Figure 1 shows the architecture of the modeling system.

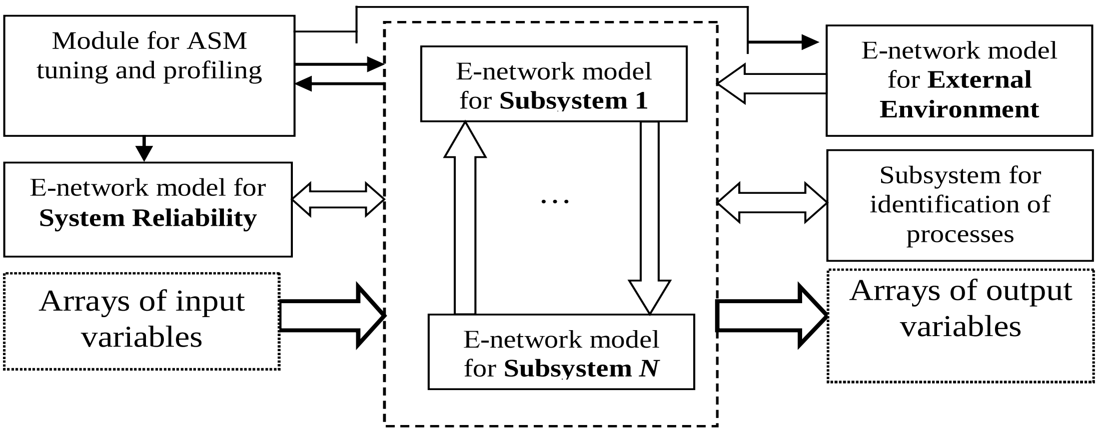

Figure 1. General structural scheme of heterogeneous processes modeling

The module for analytical and simulation model (ASM) tuning and profiling [2] allows configuring the structure and the parameters of the model depending on subsystem working modes, collecting data of modeling results, defining component working time in the model, etc. Every system to be examined is mathematically represented by E-network modeling scheme [1-3], which allows imitating logical and dynamic processes. In order to model continuous processes we propose to use E-network algorithmic modeling schemes [2, 3]. The model is adapted using the subsystem for identification of processes. The given subsystem realizes mathematical apparatus of artificial neural networks, which is widely used in order to identify processes. Besides, in the proposed system it is possible to use ASM for system reliability. The reliability parameters are assessed according to the results of process imitation in other subsystems.

3. Principles of constructing analytical and simulation model

The methodology of constructing analytical and simulation models for complex systems is based on the objective approach to the model organization and consists of the following basic positions:

- Any model is considered as a set of unidirectional objects (hereinafter referred to as components 'Comp') interacting with each other, which allows implementing a causal approach to the model construction;
- Model components are selected according to a minimum number of interconnections between model components and a minimum level of interaction (i.e., intensity of data exchange) between the components;
- Data is exchanged via communication channels;
- Model components are based on the hierarchical principle;
- Several independent processes can be performed in the model, i.e. components can function concurrently and independently of each other;
- Both concrete technical objects of the system and abstract mathematical objects, such as conversion operators, can be associated with model components.
- AIM can be adapted to the alteration in modelling aims, system parameters and structure and the environment.

The existing modelling methods including E-networks are quite effective in problem solving with the help of analytical and simulation modelling [4-6, 8]. Nevertheless, we can experience certain difficulties while creating a unified adaptive model of the system to be studied where all submodels are interrelated and cannot be viewed separately. It makes the interaction of parallel heterogeneous processes necessary. Figure 2 shows the general structure of the analytical and simulation model, which emulates parallel heterogeneous processes.

Figure 2 shows Ma*i* – analytical models of system elements or processes, Mi – analytical models obtained with the help of experimental data.

The given AIM architecture is adaptable according to the changing structure and parameters of the system to be studied as well as the environment. It can be implemented with the help of E-network logical and dynamic scheme of model control. The scheme solves the following tasks:

- Organizing model element interaction including the activation of required levels in the model hierarchy;
- Adapting the model, which includes changing model details of the system elements;
- Verifying the model;
- Teaching neural network elements to identify objects and elements of the system;
- System state backup;
- Calculating quality indicators of the system.

The procedure of determining the analytical dependence of the system elements can be conventionally divided into an analytical approach and an identification method, that is, constructing the model using experimental data. We propose to use both methods in AIM construction concept. After decomposing the system the model developer sets the system elements in one form or another.

One important task is to relate the elements of heterogeneous models to each other. For example, overheating one element of the system results in the change in system reliability and electrical properties. The further use of the object results in altering the quality level of the whole system.

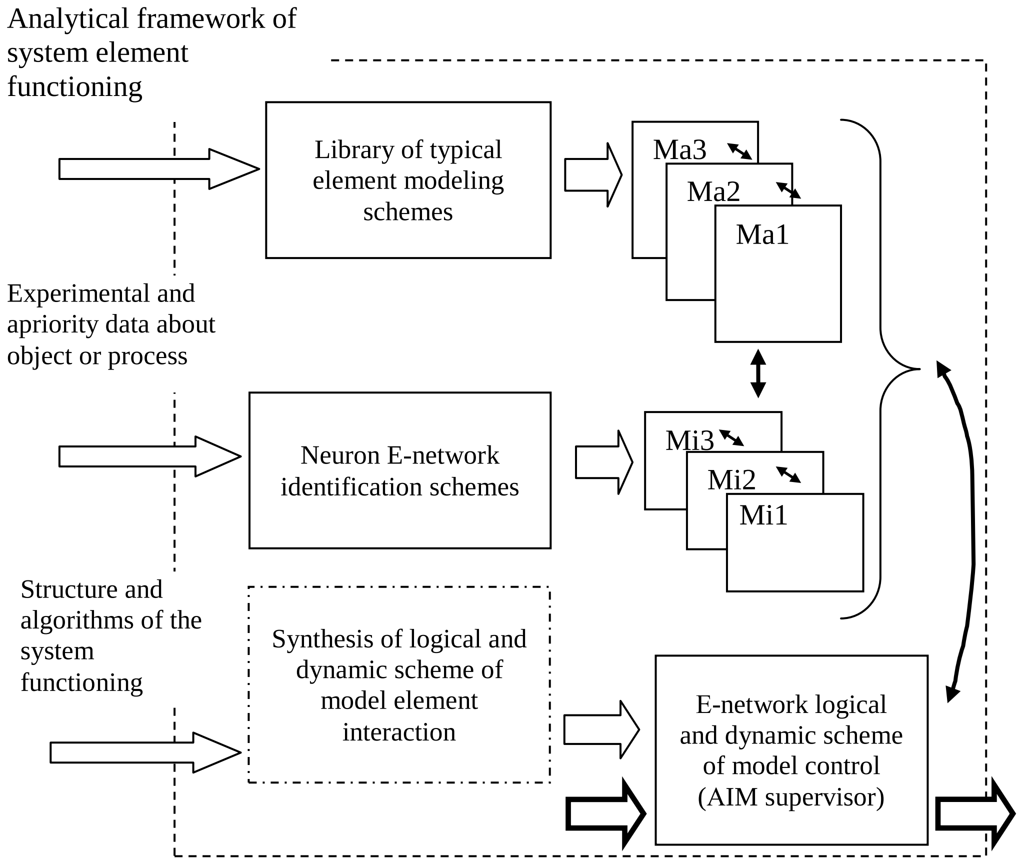

Figure 2. Structural scheme of E-network analytical and simulation model architecture

ASM of the given technical system Σ_0 is presented as the integration of components with certain interrelation. The number of model components is determined by the mode of operation of the multifunctional computer simulator (MCS) and the system complexity. Each component of the technical system model, implemented with E-network simulation apparatus, can be presented by a hierarchical 'input - state - output' structure. The functioning of model components is examined using E-network simulation apparatus at a countable set of times $T = \{t_0, t_1, \dots, t_v, \dots\}$.

The interaction of model components and the data processing flow from the environment model result in the set of events $E=E^{\prime}\cup E^{\prime\prime}$, where $E^{\prime}$ is a subset of external events $E' = \{E_1', E_2', \dots, E_\mu'\}$ characterized by input signals $\{X_\mu\}$ and E'' is a subset of internal events $E^{\prime\prime}=\{E^{\prime\prime}_1,\dots,E^{\prime\prime}_K\}$. The process of sending messages (tokens) to other components, the beginning and the end of the computational process can be considered as internal events. The model operation changes system state vector $\overline{Z(t_v)}$ z∈R. In general, the system state can have a hierarchical structure. The component state is characterized by phase variables $\overline{Z_l(t_v)}, \{z_l, l = \overline{1, L}\}$ at $t_v \in T$. State vector $\overline{Z_l(t_v)}$ is formed by processing the external (input) influences and the internal state of components at t_{v-1}. As stated above, data exchange between components is carried out via communication channels by moving tokens V_i in the network. The value is assigned to input signals $\overline{X(t_\mu)}, \{x_\mu, \mu = \overline{1, M}\}$ only 'outside', that is, while calculating token attributes. The value can be assigned to output signals $\overline{Y(t_\xi)}, \{y_\xi, \xi = \overline{1, \Xi}\}$ only 'inside' the model component unit. Input/output can be represented by control commands (including instruction commands), physical characteristics of components and environment data.

When constructing the analytical and simulation model we propose to define the model as a set of *static* $Stc_i, i = \{1, \dots, N\}$ and *dynamic* $Dc_j,\ j = \{1, \dots, M\}$ components $Stc \cup Dc \subset M_D$. The subset of static components $Stc = \{Stc_i\}$ of the technical system model includes physical elements, which are *permanently* presented in the technical system and do not depend on the phase of the current technological process. They perform a predetermined function and are in fact a constant resource in the system. Static components are characterized by certain structure *Struc*, functional algorithm *Alg*, conversion operator $H: X \xrightarrow{\Delta t} Y$ and time of the system reaction Δt. Thus static components of the model include the elements of the technological control system (controlling computer, execution mechanisms, sensors, etc.), as well as the elements of the controlled object. Dynamic components are variable resources in the technical system, which are characterized by making qualitative and quantitative changes in physical (information) parameters, depending on the phase of the process. Dynamic components in technical systems include material flows (fuel, electric current, etc.) and information flows with a given set of parameters referring to static components. In E-network apparatus the dynamic components are represented by token V_{DC} with a structured set of attributes: *X, Z, Y, Ind*, where $Ind \in N$ is the identification parameter of the token.

4. Organization of subsystem for identifying processes

In order to realize identification means we propose to use mathematical apparatus of neural networks in the given modeling system. Figure 3 shows the E-network graph, which imitates the work of an artificial neuron.

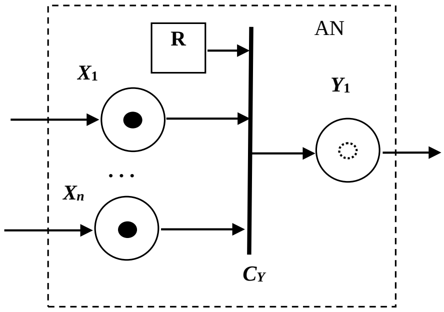

Figure 3. E-network graph of artificial neural network

Any transition of E-network can be described by three parameters [1]:

$$C_j = \left(S, t(C_j), \rho(C_j)\right),$$

where S is the type of transition S={T, X, Y, J, I, F}; $t(C_j)$ is the delay function, which determines the interval of the transition phase (i.e. the interval of the event simulated by this transition is calculated); $\rho(C_j)$ is the attribute transformation function.

Firing transition C_j results in:

- changing network layout: layout $M(b^i)$ turns into layout $M'(b^{i+1})$ according to the rule:

$$M'(b^{i+1}) = M(b^i) - I(C_j) + O(C_j);$$

where $I(C_j) = \{b \in B: P_{pre}(b,c) = 1\}$ is the array of input positions of C_j transition, $O(C_j) = \{b \in B_s \cup B_Q: P_{post}(b,c) = 1\}$ is the array of output positions of C_j transition;

- transformation of token attributes.

C_j transition takes the token from the input position and puts the label into the output position. The markup change is indicated as:

$M_0 \quad M', C_j$

where M_0 is the initial markup.

When modeling systems using E-network, apparatus transitions correspond to events and positions correspond to event conditions. In certain cases dummy activities with zero duration are introduced.

Each position b_i can have a label or can be null (position markup $M(b_i)$ is equal to '1' or '0' correspondingly). The label in the position indicates that the condition described by the given position is fulfilled. Positions with arrows entering the transition are called input positions. Positions with arrows going out of the transition are called output positions. If input positions have labels, the transition is called 'cocked'. If outcome positions are free, the transition is switched to active state after calculating the interval of active state *t*. After the delay the transition moves the label from the input position into the output position and converts attributes of the token.

Y type transitions have the position of condition *R*. The control procedure *S(R)* indicates the conditions which allow moving the token from position *X* to output position Y_i if position X_i is free at a given time, where *i* is the number of the outcome position which is determined by *S(R)*.

$$C_y = B(X_i) \wedge \overline{\boldsymbol{B(Y)}}, i = S(R), i \in \{1,2,\dots,n\}.$$

Let us consider artificial neural networks (ANN) based on E-network principles.

The functioning of artificial neurons can be divided into two stages (Fig.4): teaching and ANN functioning. At the first stage the library of teaching algorithms [4, 5] helps to set the structure, weight coefficients and ANN offsets so that the capacity $J = \sum(P_E - T_E)^2 \to min,$ where P_E is input experimental data array, T_E is output experimental data array. The given arrays are formed while conducting the experiment.

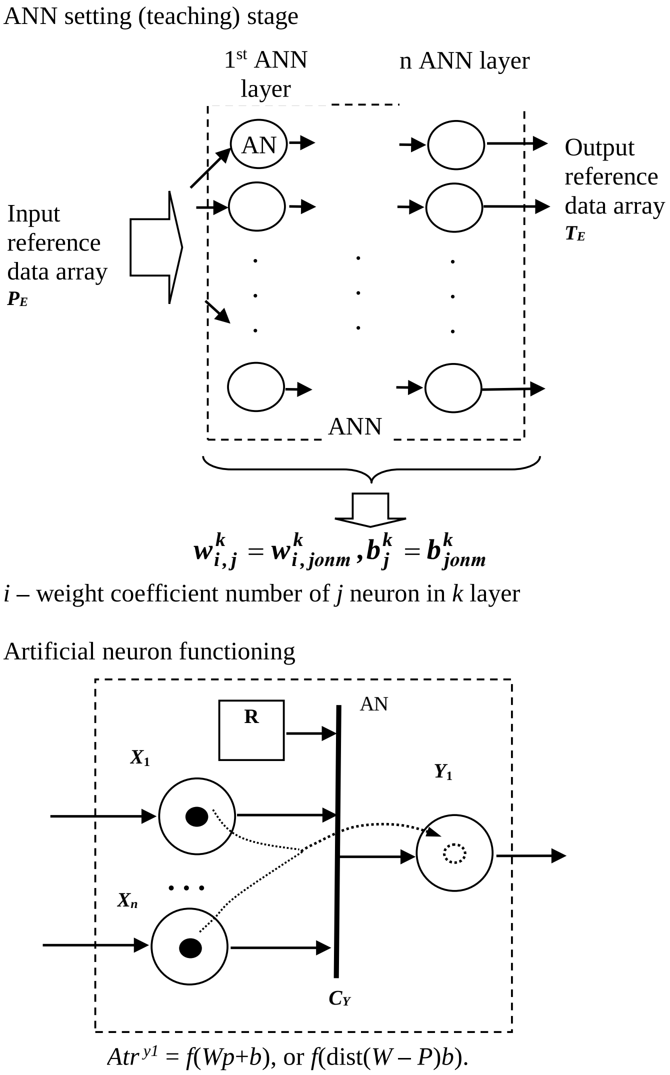

Figure 4. Stages of neural network functioning

The second stage means ANN functioning as the identification system.

When teaching neural networks we propose to use the following tuple of parameters as the attributes of tokens:

$$Atr_{ann} = \{A1, A2, A3, A4, A5\}.$$

The first attribute sets the coordinate of the artificial neuron in the network. The parameter A2 determines the type of activation function *f*, parameter A3 has weight coefficients of neuron $A3 = \{w_1, w_2, \dots, w_n\}$. The fourth attribute determines the value of offset b. Position A5 determines the length of the E-network transition delay. Each time the *Y* transition is fired at the stage of ANN teaching, the attributes are recorded in the memory buffer. Subsequently, when the neural network is taught the output activity calculation $a = f(Wp + b)$ will be based on the transition parameters recorded at the last clock period of ANN teaching. In order to teach the neural network the library of typical teaching algorithms is used, for example, the algorithms of reverse error development [2].

At the second stage the input data vector (experimental data or operator activity) is converted by each network neuron. At the stage of ANN functioning the input of the Y transition receives signals (tokens) which have at least one attribute $Atr_{ann} = \{A0\}$. The A0 network parameter has the signal value which goes to the network input. When the *Y* transition is fired the token is moved into the output position with the calculated value of the signal *a = f(Wp+b)* or *a = f(dist(W – P)b).*

5. Conclusion

E-network apparatus allows modeling heterogeneous interrelated processes and the functioning of artificial neural networks with different configuration, including dynamic neural networks. The given concept can be used for constructing computer simulators, designing and maintaining technical systems. In a multifunctional computer simulator the use of ANN components in order to identify controlled objects and processes reduces the ASM setting time. The distinguishing feature of the proposed method is that it links ASM and neural networks on the basis of expansion of Petri nets (E-network modeling) using uniform methodological positions.

References

[1] Tsapko G P, Tsapko S G and Tarakanov D V 2005 *Tomsk University Publishing* 228
[2] Best E and Fernandez C 1988 *EATCS Monographs on Theoretical Computer Science* **13**
[3] Pranyavichus G I and Shvanite D Yu 1980 *Mathematics and Mathematical Simulation* 68-72
[4] Sylla A N, Louvel M and Pacull F 2015 *PNSE'15 Petri Nets and Software Engineering* 325-326
[5] Reisig W 1985 *Petri Nets: An Introduction of Monographs in Theoretical Computer Science: An EATCS Series. Springer, Berlin* **4**
[6] Brogi A, Canciani A, Soldani J and Wang PengWei 2015 *PNSE'15 Petri Nets and Software Engineering* 191-205
[7] Jensen K 2013 *Springer Science & Business Media*
[8] Murata T 1989 *Petri nets: Properties, analysis and applications. Proceedings of the IEEE* **77(4)** 541–580