ISSN 1392 – 124X INFORMACINĖS TECHNOLOGIJOS IR VALDYMAS, 2004, Nr. 3(32)

THE USE OF FORMAL DESCRIPTION OF SYSTEMS' BEHAVIOR CREATING MARKOV MODELS

prof. habil. dr. Henrikas Pranevičius, Vytautas Germanavičius, Giedrius Tumelis

Kaunas University of Technology, Department of Business Informatics
Studentų 56, Kaunas, Lithuania

Abstract. The paper considers a method for construction of numerical models for systems described by Markov processes with discrete states and continuous time. The approach of the consequence embedding of Markov chains is used for computing stationary probabilities of Markov processes. The performance of the system described in the piece-linear aggregate approach is used for generating the system of Kolmogorov equations. An example of a numerical model for the data transmission tract with the adaptive commutation is presented.

Keywords: Markov models, stationary probabilities, embedded Markov chains, aggregate specification approach.

1. Indroduction

The Markov process is widely used for constructing functioning models of such systems as computer networks and telecommunication systems. It is known that the creation of analytical models requires large efforts. Using numerical methods permits to create a model for a wider class of systems. The process of creating numerical models for systems described by the Markov process consists of the following stages:

1. defining a state vector of a Markovian process;
2. computing a set of equations, describing the performance of the system under investigation, using a set of states and transition rates among them;
3. solving the obtained system of equations;
4. calculating probabilistic characteristics of the system.

It is necessary to emphasize that not all the stages of the automation are equally possible. The first stage is determined in a heuristic way. The description of the system functioning is the original information that determines the state vector. The principal requirement in that stage is to introduce a necessary number of coordinates that the process, describing the system behaviour, should be a Markovian.

The most difficult stage in the construction of a numerical model is to compose the system of equations describing the performance of the system. The number of equations is usually very large (counted in thousands). Thus it is very important to create an algorithm which would allow the automatic construction of the necessary equations.

In book (Pranevicius and Valakevicius 1996) the system that automatizes the construction of Markov models is presented. The embedded Markov chains are used for computing stationary probabilities of the Markov process. The performance of the system described in the event language is used for generating the system of Kolmogorov equations.

In this paper there is presented a system that automatizes the construction of Markov models when the system functioning is formally described by the aggregate method.

The name of aggregates comes from the piece-linear aggregate approach (PLA) proposed by Buslenko (Buslenko 1971) for the simulation of complex systems. Later Pranevicius (Pranevicius 1982; Pranevicius and Valakevicius 1990) developed a method for the formal description of PLA based on the notion of controlling sequences. In H. Pranevicius' doctoral thesis (Pranevicius 1983) it was shown how PLA with the semantics of controlling sequences can be used for the formal specification, validation and simulation of complex systems. The proposed mathematical model lies in the base of several implemented simulation system (Pranevicius et al. 1994) running on machines of various platforms. The model permits not only the simulation but also the correct analysis on the base of the same formal specification. The combination of these tasks is very important for designing distributed software systems.

The system proposed in this article differs from the widely known MACOM system (Scittinick and Muller-Clostermann 1990) which is a software tool for the model based performance evaluation of communication systems. The models are specified in MACOM by graphical interactive means, the model solution is performed by using numerical techniques for Markovian models. A MACOM model is a network of sources, sinks, resources and routing elements. In case a requested resource is currently not available, a load unit may utilize an alternative resource, may wait, or get lost.

In the Section 2 of the article there is presented an algorithm for calculating stationary probabilities of Markovian processes; this algorithm uses the method of inserted Markovian chains. In Section 3 an aggregate specification metamodel is presented. In Section 4 a formal specification of the data transfer tract functioning and the results of modeling are presented.

2. Computation of stationary probabilities usING sequential embedding of Markov chains

For the solution of the system of equations

$$\sum_{j=1}^{N} q_i \lambda_{ij} = \sum_{j=1}^{N} q_j \lambda_{ji}, \ i = \overline{1, N},$$

where $q_i = \lim_{t \to \infty} q_i(t), \ i = \overline{1, N}$ – steady state probabilities of Markov process. A Markov chain is embedded into Markov process. The steady state probabilities of

$$p_i = \sum_{j=1}^{N} p_i p_{ij}, \ i = \overline{1, N},$$

here λ_{ji} – the rate of transition and p_{ij} – the transition probability from state s_j to state s_i.

Computation of the stationary probabilities $p_i, \ i = \overline{1, N}$ of system states involves two stages: the stage of embedding the Markov chains and the one of computing the steady-stage probabilities (Pranevicius 1982).

The algorithm has the following form:

1. The stage of embedding the Markov chains:

$$p_{ij}^{(k)} = p_{ij}^{(k+1)} + p_{i,k+1}^{(k+1)} \cdot \frac{p_{k+1,j}^{(k+1)}}{1 - p_{k+1,k+1}^{(k+1)}},$$

2. The stage of computing the stationary probabilities:

$$p_i^{(1)} = 1;$$

$$p_i^{(k+1)} = \begin{cases} p_i^{(k)}, & i = \overline{1, k}, \\ \dfrac{\sum_{j=1}^{k} p_j^{(k)} p_{j,i}^{(k+1)}}{1 - p_{ii}^{(k+1)}}, & i = k+1; \end{cases} \quad k = \overline{1, (N-1)};$$

$$p_j = \frac{p_j^{(N)}}{\sum_{j=i}^{N} p_j^{(N)}}, \ j = \overline{1, N}.$$

The steady-state probabilities q_i of the Markov process are found from the formula:

$$q_i = \frac{p_i / \sum_{j=1}^{N} \lambda_{ij}}{\sum_{j=1}^{N} \left(p_i / \sum_{j=1}^{N} \lambda_{ij} \right)}, \ i = \overline{1, N}.$$

$$q_i = \frac{p_i / \sum_{j=1}^{N} \lambda_{ij}}{\sum_{j=1}^{N} \left(p_j / \sum_{r=1}^{N} \lambda_{jr} \right)}, \ i = \overline{1, N}$$

Usually in real systems the transitions on the given set of possible states are done by means of transition rates and not by transition probabilities. A slightly modified algorithm is proposed. Let $\lambda_N = \left(\lambda_{ij}^{(N)} \right)_{N \times N}$ be the matrix of transition rates for a Markov chain on the set of states $S = \{s_1, \ldots, s_N\}$. Then

$$\lambda_{ij}^{(k)} = \lambda_{ij}^{(k+1)} + \frac{\lambda_{i,k+1}^{(k+1)} \cdot \lambda_{k+1,i}^{(k+1)}}{\overline{S}_{k+1}^{(k+1)}},$$

$$\overline{S}_{k+1}^{(k+1)} = \sum_{\substack{j=1 \\ j \neq k+1}}^{N} \lambda_{k+1,j}^{(k+1)},$$

$$r_1^{(i)} = 1,$$

$$r_i^{(k+1)} = \begin{cases} r_i^{(k)}, & i = \overline{1, k}, \\ \dfrac{\sum_{j=1}^{N-1} r_j^{(k)} \lambda_{ji}^{(k+1)}}{\overline{S}_{k+1}^{(k+1)}}, & i = k+1; \end{cases} \quad k = \overline{1, (N-1)}$$

$$q_i = \frac{r_i^{(N)}}{\sum_{j=1}^{N} r_j^{(N)}}, \ i = \overline{1, N}.$$

3. The Metamodel of the aggregate specification approach

The aggregate specification method is presented in book (Pranevicius et al. 1994). Here is presented the metamodel of this specification method, written using the UML notation.

There is shown in Figure 1 the classes' diagram. It describes the structure of data of the linear aggregate, inputs, outputs, discrete and continuous variables of aggregate.

Continuous variables are used for describing the time moments of which the internal events occur after the operations terminate.

Signals coming through inputs of aggregate cause external events.

Each event of an aggregate (external and internal) is related to the transition operator *H* and the transition operator *G* (see Figure 2).

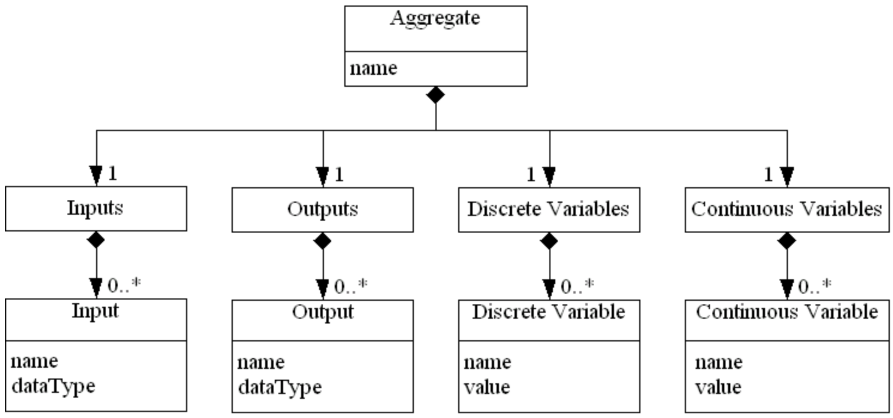

Figure 1. Data Structure of Aggregate

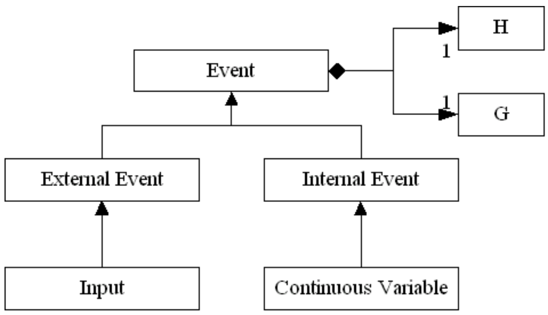

Figure 2. The events processing scheme

State vector of aggregate is constructed from discrete and continuous variables (see Figure 3).

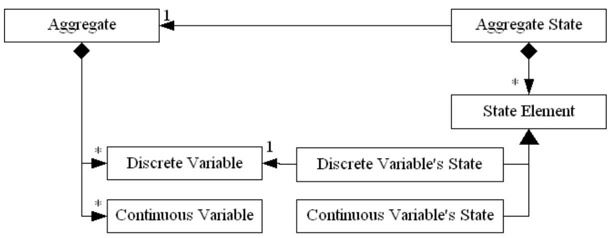

Figure 3. The structure of the aggregate states vector

The continuous variables of the aggregate in the state vector are represented as 0 or 1:

- 1 – means that the operation is active at the given moment,
- 0 – the operation is not active.

The modeling system consists from aggregates and connection between them (see Figure 4).

Aggregates interact in the system; in order to describe their interaction the connections are used. Signals can be transmitted by connections among the aggregates. The signals' data structure is shown in Figure 5.

The connections connect one output of one aggregate with one or several inputs of one or several aggregates of the same type.

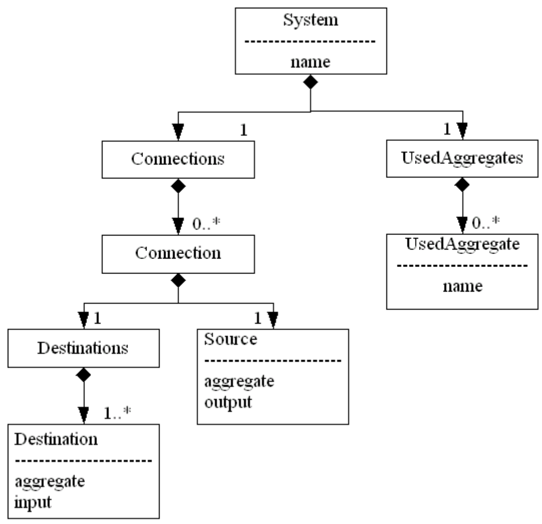

Figure 4. The structure of system of aggregates

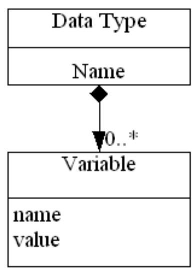

Figure 5. The Structure of inputs and outputs signals

3. The model of a data transmission track with an adaptive switching method

3.1. The conceptual model of the system

The behavior of such a system may be represented as the following queuing system (see Figure 6): the message flow arriving to the track is considered to be distributed by Poisson distribution and the service times are independent and exponentially distributed. Two request flows enter the system: the flow of file sessions with rate λ_1, and the packets flow with rate λ_2.

There are three groups of channels in the track: N_f channels are for the transmission of the file sessions, N_p channels for the transmission of the packets and N_{fp} channels for the transmission either the file sessions or the packets.

The file sessions are served if there are any free channels, otherwise they are rejected. If all the channels are occupied, the message packets wait in the queue Q, the length of which is limited to l.

In case all the channels N_f are occupied when the file session enters the system, it can occupy a free channel in the group N_{fp} only if the number of packets in the queue Q does not exceed l_1 ($0 < l_l < l$).

Packets are served by channels of the group N_p , by free channels of the group N_{fp} and during the intervals between the messages in a file sessions transmitted by the channels. A part of idle intervals in the file session is represented by ρ. If a packet finds more than l_2 ($l_1 < l_2$) requests in the queue, it occupies one channel occupied by file sessions in the group N_{fp}. A file session is served by a channel at the rate μ_1 and a packet is served by a channel at the rate μ_2.

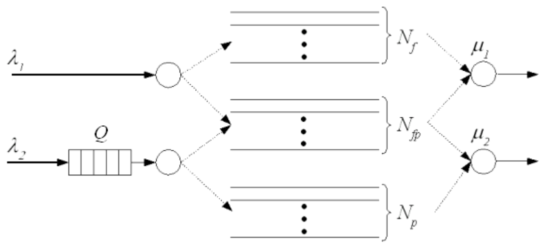

Figure 6. Presentation of data transmission track by queuing system

3.2. Aggregate specification of the system

Aggregate specification is presented by 9 items:

1. The set of input signals $X = \varnothing$
2. The set of output signals $Y = \varnothing$.
3. The set of external events $E' = \varnothing$.
4. The set of internal events

$$E'' = \{e_1'', e_2'', e_3'', e_4''\}$$

where e_1'' – the arrival of a file session;

e_2'' – the arrival of a packet;

e_3'' – the completion of serving the file session;

e_4'' – the completion of serving the packet.

5. The transition rates between the system states:

$$e_1'' \mapsto \lambda_1,\ e_2'' \mapsto \lambda_2,\ e_3'' \mapsto \mu_1,\ e_4'' \mapsto \mu_2 .$$

6. The discrete comment of state

$$v(t) = \{n_f(t), n_{fp}(t), n(t)\},$$

where $n_f(t)$; – the number of the occupied channels in the group N_f at the moment t;

n_{fp} – the number of the channels occupied by file sessions in the group N_{fp} at the moment t;

$n(t)$ – the length of the packet queue Q at the moment t.

7. The continuous component of state

$$z_v(t) = \{w(e_1'',t), w(e_2'',t), w(e_3'',t), w(e_4'',t)\}.$$

8. Initial state
9. Transition operators:

$H(e_1'')$: /The file session has arrived/

$$n_f(t+0) = \begin{cases} n_f(t)+1, \text{if } n_f(t) < n_f, \\ n_f(t), \text{otherwise}; \end{cases}$$

$$n_{fp}(t+0) = \begin{cases} n_{fp}(t)+1, \text{if } (n_f(t) = N_f) \wedge \\ \quad (n_{fp} < N_{fp}) \wedge (n < l_1), \\ n_{fp}, \text{otherwise}; \end{cases}$$

$$n(t+0) = n(t);$$

$$w(e_1'', t+0) = w(e_1'', t);$$

$$w(e_2'', t+0) = w(e_2'', t);$$

$$w(e_3'', t+0) = (n_f(t) + n_{fp}(t))^* \mu_1;$$

$$w(e_4'', t+0) = \begin{cases} n_p(t) + (N_{fp}(t) - n_{fp}(t)) + \\ \quad (n_{fp}(t) + n_f(t) \cdot \rho) \cdot \mu_2, \quad \text{if } n(t) > 0 \\ 0, \text{otherwise}; \end{cases}$$

$H(e_2'')$:

$$n_f(t+0) = n_f(t);$$

$$n_{fp}(t+0) = \begin{cases} n_{fp}(t) - 1, \text{if } (n_{fp} > 0) \wedge (n < l_2), \\ n_{fp}(t), \text{otherwise}; \end{cases}$$

$$n(t+0) = n(t) + 1;$$

$$w(e_1'', t+0) = w(e_1'', t);$$

$$w(e_2'', t+0) = w(e_2'', t);$$

$$w(e_3'', t+0) = w(e_3, t);$$

$$w(e_4'', t+0) = \begin{cases} n_p(t) + (N_{fp}(t) - n_{fp}(t)) + \\ \quad (n_{fp}(t) + n_f(t) \cdot \rho) \cdot \mu_2, \text{if } n(t) > 0 \\ 0, \text{otherwise}; \end{cases}$$

$H(e_3'')$:

$$n_{fp}(t+0)=\begin{cases} n_f(t)-1, \text{if } (n_f>0)\wedge n_{fp}(t)=0, \\ n_f(t), \text{otherwise}; \end{cases}$$

$$n_{fp}(t+0)=\begin{cases} n_{fp}(t)-1, \text{if } n_{fp}>0, \\ n_{fp}(t), \text{otherwise}; \end{cases}$$

$$n(t+0)=n(t);$$

$$w(e_1'',t+0) = w(e_1'',t);$$

$$w(e_2'',t+0) = w(e_2'',t);$$

$$w(e_3'',t+0) = (n_f(t)+n_{fp}(t))^* \mu_1;$$

$$w(e_4'',t+0)=\begin{cases} n_p(t)+(N_{fp}(t)-n_{fp}(t))+(n_{fp}(t)+n_f(t)\cdot\rho)\cdot\mu_2, \\ \qquad\qquad\qquad \text{if } n(t)>0 \\ 0, \text{otherwise}; \end{cases}$$

$H(e_4'')$:

$$n_f(t+0)=n_f(t);$$

$$n_{fp}(t+0)=n_{fp}(t);$$

$$n(t+0)=\begin{cases} n(t)-1, \text{if } n(t)>0, \\ 0, \text{otherwise}; \end{cases}$$

$$w(e_1'',t+0) = w(e_1'',t);$$

$$w(e_2'',t+0) = w(e_2'',t);$$

$$w(e_3'',t+0) = w(e_3,t);$$

$$w(e_4'',t+0)=\begin{cases} n_p(t)+(N_{fp}(t)-n_{fp}(t))+ \\ \quad (n_{fp}(t)+n_f(t)\cdot\rho)\cdot\mu_2, \quad \text{if } n(t)>0 \\ 0, \text{otherwise}; \end{cases}$$

4. Results of the modeling

The experiments were carried out with the following data, see Table 1:

Table 1. System's parameters

N_f	N_{fp}	N_p	l	l_1	l_2	μ_1	μ_2	ρ
10	9	1	10	3	8	0,0055	6	0,5

A fragment of the state graph of the data transmission track is presented in Figure 7, when *N_f=1, N_{fp}=2, l=5*.

Following formulas are used for calculation of the characteristics of the described system:

$$p = \sum_{n=0}^{l} p(n_f=N_f, n_{fp}=N_{fp}, n) + \sum_{n_{fp}<N_{fp}} \sum_{n=l_1+l_2}^{l} p(n_f=N_f, n_{nf}, n),$$

where p is the probability of rejection of a file session;

$$\overline{n_f} = \sum_{n_f=0}^{N_f} \sum_{n_{fp}=0}^{N_{fp}} \sum_{n=0}^{l} (n_f+n_{fp}) p(n_f, n_{fp}, n),$$

where $\overline{n_f}$ is the average number of the channels occupied by file sessions;

$$\overline{n} = \sum_{n_f=0}^{N_f} \sum_{n_{fp}=0}^{N_{fp}} \sum_{n=0}^{l} np(n_f, n_{fp}, n),$$

where $\overline{n}$ is the average length of a packet queue;

$$\overline{t} = \frac{\overline{n}+1}{(N_f+N_{fp}+N_p-\overline{n}+\overline{n}\cdot\rho)\cdot\mu_2},$$

where $\overline{t}$ is average time of the packet being in the system. The results of modeling are presented in Table 2.

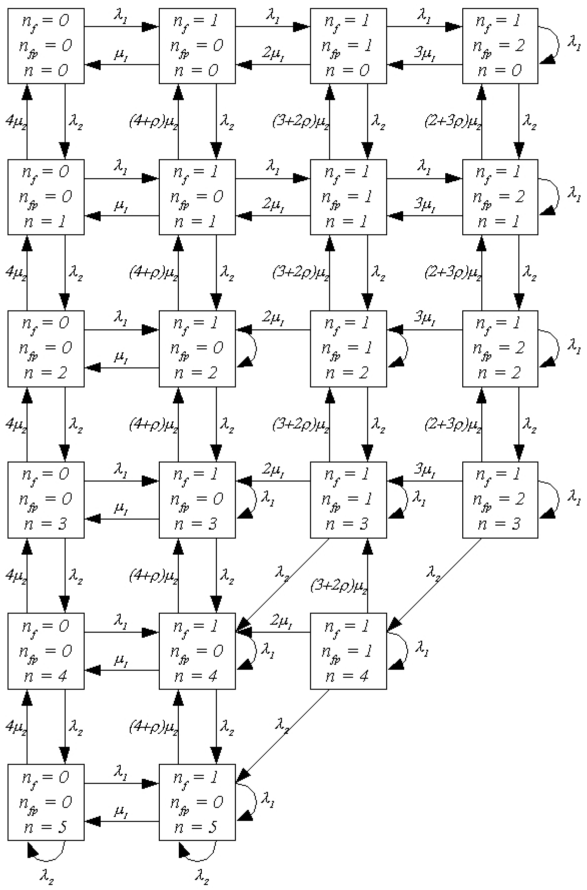

Figure 7. The state graph for the analyzed data transmission system

Table 2 Results of modeling

λ_1	λ_2	μ_1	μ_2	p	$\overline{n_f}$	$\overline{n}$	$\overline{t}$
0,0064	7,20	0,0055	6	7,49E^{-017}	1,2	0,02	0,05
0,0192	21,7	0,0055	6	1,49E^{-008}	3,5	0,22	0,07
0,0256	28,9	0,0055	6	1,19E^{-006}	4,6	0,43	0,08
0,0384	43,4	0,0055	6	2,46E^{-004}	6,5	1,25	0,13
0,048	50,6	0,0055	6	1,75E^{-003}	7,4	1,86	0,18

5. The modeling program MSM

The modeling is performed using the digital modeling program MSM (Markov Systems Modeling). This program has module for every step of numerical modeling(see Figure 8).

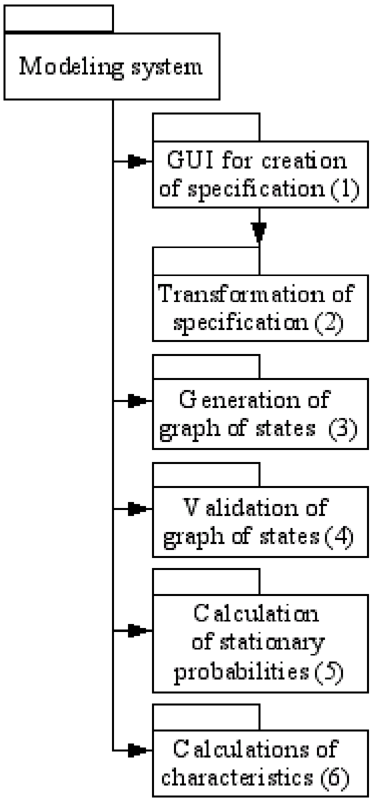

Figure 8. The structure of the MSM program

The graphical interface (1) is used to create the specification of the system using aggregate method. Specification is saved using the XML based language. XML and XLST technologies allows to make easier transformations.

The transformation module (2) transforms the aggregate specification described in the XML into a code for generating graph of states.

The graph generator module (3) compiles the graph generator code and generates graph of system states.

The graph validation module (4) validates graph of system states: checks the reachability graph nodes, detects dead-ends and cycles.

The module of stationary probabilities calculation (5) calculates the stationary probabilities of states of the system using the method of embedded Markov chains.

The module of characteristics calculation (6) calculates various characteristics of the system using the calculated stationary probabilities. This module calculates in the current version the average values of variables of the system.

6. Conclusions

The presented method allows to automatize creation of numerical Markov models: generating the set of states of the system and transition rates between states, calculating stationary probabilities and characteristics of the modeled system.

References

[1] **N. Buslenko.** On a class of complex systems. *Problems of applied mathematics and mechanics, Nauka, Moscow (In Russian)*, 1971, 56-68.

[2] **H. Pranevicius.** Models and methods for computer system investigation. *Mokslas, Vilnius, (In Russian),* 1982, 1-228.

[3] **H. Pranevicius.** Automatization of developing numerical and simulation models for computer systems. *PhD dissertation. (In Russian),* 1983.

[4] **H. Pranevičius, V. Pilkauskas, A. Chmieliauskas.** Aggregate approach for specification and analysis of computer network protocols. *Kaunas University of Technology, Technologija,* 1994, 1-152.

[5] **H. Pranevicius, E. Valakevicius.** Numerical models of systems specified by Markovian proceses. *Kaunas University of Technology, Kaunas, Technologija*, 1996, 1-101.

[6] **M. Scittinick, B. Muller-Clostermann.** MACOM – A tool for the Markovian Analysis of Communication System. *Proc. Of the 4th Int. Conf. on Data Communications Systems and their Performance, Barcelona, Spain*, 1990, 456-470.

DOI: 10.5755/j01.itc.32.3.11861