UDC 519.179.2

PARALLELISM OF FUNCTIONING OF LOGICALLY DIVIDED SUBSYSTEMS IN A COMPLEX SYSTEM AT E-NETWORK SIMULATION

S.G. Tsapko, I.V. Tsapko

Tomsk Polytechnic University
E-mail: serg@aics.ru

Principles of object-oriented simulation have been considered. Applicability of mathematical simulation device for describing interaction logic of complex system inner components was shown. Comparative analysis of description methods of parametric components in complex system was carried out. The main points of E-network simulation method of interactive processes were given. The example of forming conversion circuit of token attributes on the basis of model E-network graph and functions describing parametric information conversions was shown.

According to the main principles of object-oriented simulation stated in [1, 2] any complex system consists of final set of logically completed modules called subsystems of the given complex system and having intermodule logical and parametric communications. Each module in its turn may be also presented in the form of a set of subsystems on a lower abstraction layer. Therefore, any complex system may be imaged in the form of collection of small autonomous subsystems having external and internal constraints, its inner structure, its inner state, sets of input actions and output parameters. Each subsystem may be presented as an object described in terms of E-network. E-network model of selected complex system may be obtained on the basis of the method of obtained object arrangement.

With the help of mathematical simulation device the model describing adequately inner processes of any system may be obtained. If a user interacts with a complex system through interface implemented by PC it is possible to develop software simulator of dynamics of the given system functioning. Examining physical object structure from position of four-dimensional coordinate system where one of coordinates is a time coordinate, simultaneous parallel functioning of simple subsystems being a part of the complex system may be supposed. It is obvious that any simple subsystem exists in each time moment and has its inner state changing depending on time coordinate.

Uniprocessor computing devices can not simultaneously perform several processes simulating simple subsystem functioning in complex system as each process implies tracing conditions of changing simulated object state as well as change proper of their state. There are many software products allowing performing several processes pseudo-parallel. Each process occupies processor for a certain set time and then it finishes with current state saving and put into queue of processes waiting for their performance. Thus, each process has an access to processor computing facilities inversely to a number of processes performed simultaneously in PC.

To implement parametric communication of two parallel processes DSM are applied as a rule. This mechanism implementation requires deep knowledge of operational system functioning as well as using additional software tools for organizing process parametric communication. The device of E-network simulation is intended for simulating processes occurring parallel in time and allows developing a library of simple subsystems in terms of E-network. Using main approaches stated in methodology of E-network information logical design [5] it is possible to obtain E-network model of the complex system from the collection of models connected in parallel-serial bonds.

The implementation problem is construction of a logic chain of attributive information transfer in the process of cyclic functioning of E-network model. Dynamic functioning of simulation models of single subsystems does not allow forming complete parametric model of their interaction. There are many approaches to this problem solution [1]. The authors of the article suggested adding parametric feedback into E-network graph and introducing notion of instantaneous state vector in complex technical system. Parametric data array the elements of which characterize adequately the state of complex engineering system is implied by the state vector of complex engineering system. In this case array fields are grouped into a subset each of which characterizes separately the state of one of subsystems.

The main advantage of the device of E-network simulation allowing organizing the simulation process of parallel functioning subsystems is clocking and division into cycles of E-network model functioning in dynamics [4, 5]. Time required for response of each allowed transition activated by token single appearance in output positions of transition-generators is taken as a cycle. Therefore, final designation of E-network graph is achieved by a certain amount of model cycles as a result of a serial change of E-network position designation. Token motion path depends on logic of allowed transition response according to the given mathematical expressions. The process of designation serial change defines the cycle of E-network model functioning.

To simulate interacting subsystems functioning in parallel the method of E-network simulating of interacting processes was suggested. It consists in:

1. At the first stage any complex system is divided into a number of simple subsystems. Logic of their functioning can be described by E-network graph. For each subsystem an inner structure is determined. It is represented by E-network graph and software model characterizing the nature of transformation at change of E-network graph designation. Besides, the original state of designation characterizing current state of si-

mulated subsystem is selected. Vector of possible input actions and vector of output parameters are also determined for each subsystem. Basic library of E-network models for simulation of selected complex system is formed from the obtained E-network models.

2. At the second stage vector of input actions and output parameters of E-network models from basic set are analyzed. On the basis of this analysis the integrated vectors of input actions and output parameters are formed. Then the integrated vectors are analyzed for detecting identical parameters characterizing interaction of parallel functioning subsystems. Simulation of liquid motion in a tube owing to blade operation driven by electrical motor may be given as an example illustrating this point. Here rate depending on rotation frequency of motor impeller is simultaneously the output parameter for the model of propulsion device and input one for the model of a tube with water. Therefore, this parameter characterizes interaction of two parallel functioning subsystems.
3. The third stage is characterized by combination of E-network models from the library obtained according to the point 1. Input of each E-network model from the library combines with transition multiplying token and each output – with transition combining input tokens. Multiplying and combining transitions are connected by feedback, Fig. 1.

In the given case the multiplying transition is transition T_2, and the combining one is transition T_4. Output tokens of all parallel functioning systems enter the input of transition T_4. Into T_4 transition the function is added. It forms token in position S_{n+1} attributes of which are the attributes of tokens entered the input of transition T_4. This token characterizes the complex system state and its attributes characterize parameters of input action vectors of parallel functioning subsystem. Token formed at output of transition T_4, enters through the transition T_1 to the input of multiplying transition T_2. In transition T_2 function of input action vector formation is introduced for each parallel functioning subsystem. Parameters of input action vectors are transferred into parallel functioning E-network models with input token attributes (S_5, S_6, ..., S_k). Transition T_1 which allows introducing external actions into functioning model as well as influencing E-network model operation by control signals is introduced In Fig. 1.

Each token at E-network model dynamic functioning possesses parametric attributes characterizing each subsystem state. According to conventional denotation reference to token attribute consists of two parts: indication of position in network graph and number of used attribute. For example, N attribute of K token is presented by notation of the type s[K].atr[N]. In generalized view information transfer in model experiment may be presented in the form of diagram of attribute transformation, Fig. 2.

In Fig. 2 vectors of output parameters of each parallel functioning subsystem enter into group of vectors $\mathbf{A}_i$. Then parameters of vectors of group $\mathbf{A}_i$ are combined into a single vector characterizing the complex system state at the current simulation cycle. The obtained vector refers to the group of vectors $\mathbf{B}_i$. Group of vectors $\mathbf{B}_i$ characterizes summation action to the complex system. It combines internal action reflecting interactions of in-

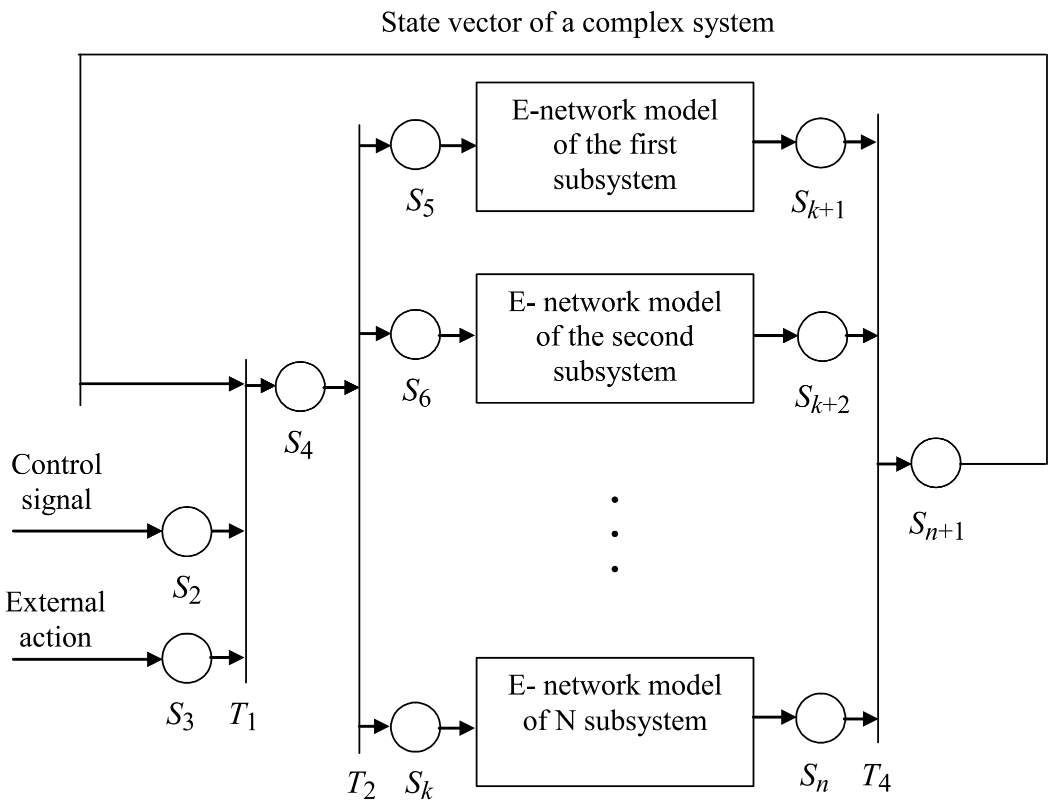

Fig. 1. Connection of E-network models from the basic library

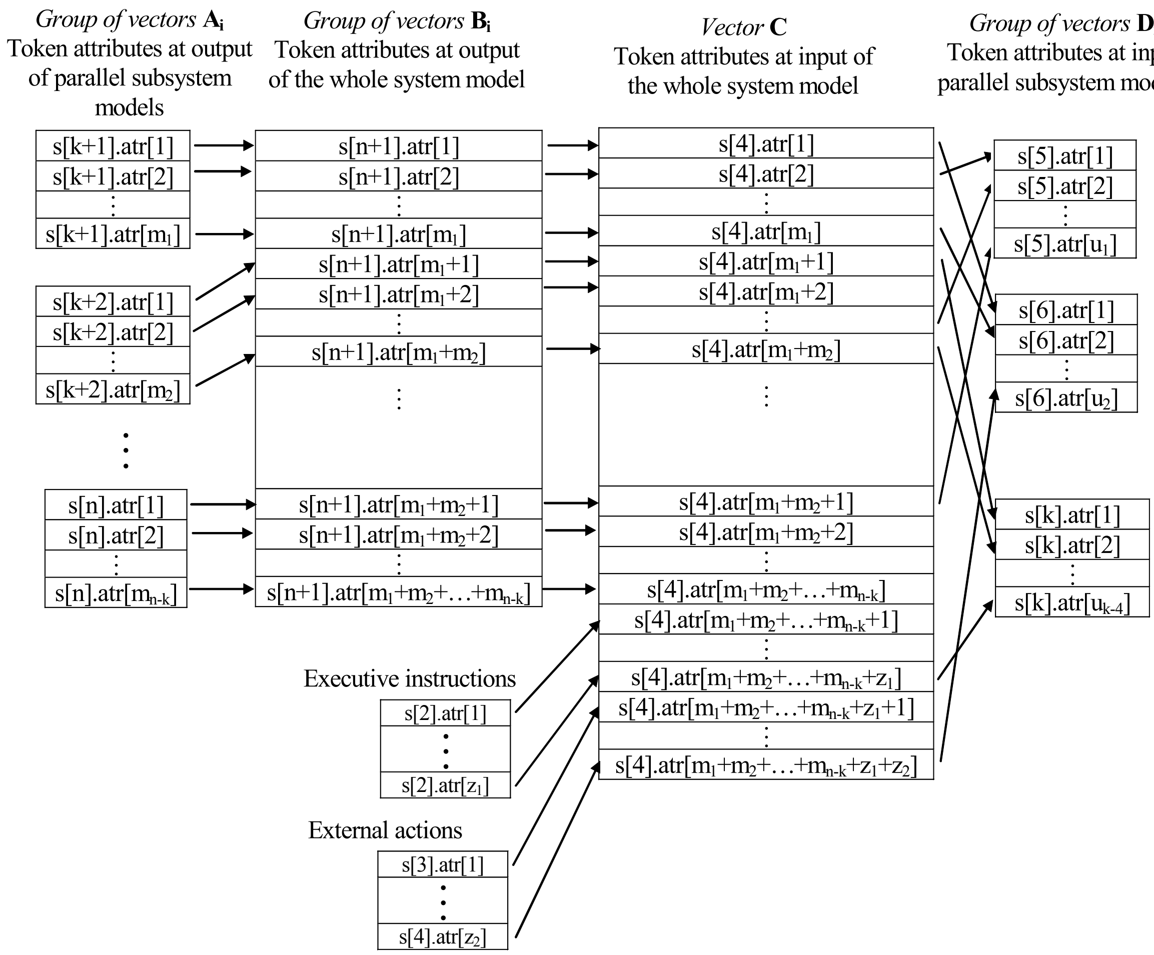

Fig. 2. *Information transfer at simulation*

ner subsystems as well as external actions and executive instructions. Vector **C** which is a combined vector of external actions vectors of parallel functioning subsystems is formed from group of vectors $\mathbf{B}_i$**.** In Fig. 1 attributes of token in position S_4 contain values of vector **C** parameters. In Fig. 2 the example of forming vectors of group $\mathbf{D}_i$ from vector **C** is shown schematically. This function of forming input action vector of parallel functioning subsystems should be described in transition T_2 (Fig. 1) as a transfer of token attribute values from input position to token attributes of output positions.

Thus, the obtained E-network model of the complex system consists of parallel functioning models of simple subsystems. Parallelism of functioning E-network models is stipulated by clocking mechanism and interaction – by ability of parameter transfer by attributes of moving token. It is seen from Fig.1 that a token enters the input of any E-network model from parallel situated at the moment of transition T_2 response. This fact may be considered as a fact of object start for performance. Input token contains a set of parameters specified in the form of attributes which can be considered as a vector of input actions. Each parallel model is in one of its defined states which is determined in its turn by the values of marked position attribute. The token activating the process of response of the object presented by E-network model generates appearance of the token at the output of E-network model in time equal to a cycle of this E-network model operation.

Transitions in E-network model response serially but model cycle termination is characterized by absence of allowed transitions. Therefore, all that performed in time interval from the beginning of the cycle to its end is considered to be performed parallel for user. In this case transition T_4 (Fig. 1) is started over only when all its input positions are marked that is contain a token. This situation is possible only at token appearance at output of all E-network models simulating operation of simple subsystems. So transition T_4 is started over in time interval determined as the greatest operation cycle among the presented parallel situated E-network models. Transition T_4 in this case is a clocking and determining the parallelism of functioning E-network models of simple subsystems being a part of complex system.

The conclusion about the ability of using the device of E-network modeling for development and maintenance of simulation models functioning parallel and having parametric interaction follows from the considered above. The suggested method and algorithm of its implementation allow developing arbitrarily complex simulation model described in terms of the device E-network modeling. This method was tested at development of software simulator of functioning and dynamic interaction logic of spacecraft subsystems.

REFERENCES

1. Buch, Grady. Object-oriented analysis and design with application examples on C+: Translated from English / G. Buch. – 2[d] issue – Moscow: Binom, 2000. – 560 p.: ill.
2. Kazimir V.V., Demshevskaya N.V. Formal object-oriented approach to complex system simulation // In collected articles: «Перша Міжнародна науково-практична конференція з програмування УкрПрог'98». – Kiev: Cybernetic centre of Ukraine NAS, 1998. – P. 593–598.
3. Tsapko S.G. E-network facilities of modeling and simulation of processes of functioning complex dynamic configurable systems in tasks of construction of computer simulators: Thesis of ... candidate of tech. science. – Tomsk, 2003. – 192 p.
4. Tsapko G.P., Tsapko S.G. Tarakanov D.V. Basic principles of E-network model construction of complex engineering system // Bulletin of the Tomsk Polytechnic University. – 2006. – V. 309. – № 4. – P. 152–157.
5. Tsapko S.G., Tsapko I.V. Object presentation of simulation models of complex engineering system subsystems in terms of E-network // Bulletin of the Tomsk Polytechnic University. – 2006. – V. 309. – № 5. – P. 167–170.

Received on 25.06.2007

UDC 519.876.2(004.4'22)

TECHNIQUE AUTOMATED OF DIAGRAM CONSTRUCTION IN BUSINESS PROCESS MANAGEMENT SYSTEMS

I.G. Ozerova, E.A. Dmitrieva, G.P. Tsapko, V.N. Vichugov

Tomsk Polytechnic University
E-mail: ira@aics.ru

Technique allowing reducing stages of analysis and design of application while implementing Business Process Management System (BPMS) has been suggested. It was possible due to elimination of enterprise activity examination stage and formation of business process models on the basis of structural functional models obtained as a result of reengineering project or developing quality management system. The required steps of model construction in BPMS were revealed. Appropriateness of business process modeling with the help of traditional meansand further use of models for transfer into BPMS by conversion was validated. Algorithm of automated transformation on the basis of processing XML-files of models was suggested.

In the radical reengineering becomes unacceptable method for improving business processes as its labor content does not allow reacting rapidly to changeable market demands. Corporate information systems of type *ERP* (*Enterprise Resource Planning*) introduced by reengineering results allow adapting to any enterprise structure but frequently long duration of reconfigurations disables enterprise to manuge their business process managing in real time.

Investigations of enterprise management principles from the position of process approach are widely given in scientific literature by both foreign and domestic authors [1–5]. At the beginning of the current century the process approach has got supporting in the form of software tools *BPMS* and now is called *BPM – Business Process Management* [6–9]. *Unify NXJ* (Unify), *Oracle BPEL Process Manager* (Oracle), *ActiveBPEL* (Active Endpoints) may be given as the examples of *BPM*-systems (and their developers). *BPMS* architecture including graphics editor, engine, monitoring module allows updating existing processes in the required rate.

A code in a special programming language, for example *BPEL* (*Business Process Execution Language*) corresponds to the diagram of the process in *BPMS*. The process in *BPEL* language itself does not fulfill any functions and intended exclusively for coordination (or *orchestration*) of web services. *BPEL* specification is approved as standard of OASIS (Organization for the Advancement of Structured Information Standards) [10]. Some part of developers uses this standard and others use their own nonstandardized languages of process description.

At present applications in *BPMS* are developed on the basis of user manuals which contain description of interfaces and installation steps. New concept requires novelty in its usage as well.

The suggested technique of process diagram construction in *BPMS* includes the following main stages:

1. Simulation of business processes using facilities of *CASE* (*Computer-Aided Software/System Engineering*).
2. Conversion of *CASE*-model into *BPM*-model.

Here and further *CASE*-facilities are implied as their subset which is intended for simulating business processes. For example, *AllFusion Process Modeler* (earlier *BPwin*), *ARIS Toolset. BPwin* supports the following methodologies: *IDEF*0 (*Integrated Computer Aided Manufacturing* (*ICAM*) *DEFinition language* 0), *DFD* (*Data Flow Diagram*), *IDEF*3; *ARIS Toolset – VACD* (*Value-added chain diagram*), *eEPC* (*extended Event-driven Process Chain*), *FAD* (*Function allocation diagram*), *IFD* (*Information flow diagram*) etc. Models implemented