UDC 004.94

M.I. Kochergin

Interpretation of the statechart diagram into a multilevel simulation language

The paper deals with the matters of simulating physics-and-technics hybrid systems in the multilevel modeling environment MARS. This study gives a comparison of a straight algorithmic approach suggested by the language of modeling algorithmic constructions (MAC-language) of MARS and using statecharts for modeling hybrid systems. This paper gives the rationale for interpretation of the statechart diagram into MAC-language of MARS and describes the component «State» developed for modeling discrete dynamic behavior of a system in MARS.

Keywords: discrete-continuous dynamic system, hybrid system, computer modeling, simulation, statechart, finite-state machine.
doi: 10.21293/1818-0442-2017-20-4-122-125

To qualitatively describe a large class of practical problems, it is necessary to take into account not only continuous but also discrete behavior of systems. Systems that have both continuous behavior (for example, physical and chemical behavior) and discrete (logical) behavior that determines the algorithm of the system's activity are called *discrete event dynamic systems* (DEDS) or *hybrid systems* (HS). Such systems are also called *discrete continuous systems*, *variable structure systems*, *event-driven systems,* etc. A system may have the behavior of both types simultaneously because of the following main factors: 1) co-operation of continuous and discrete objects, 2) instant qualitative changes in the continuous object, 3) changing the composition of the system [1]. Various approaches can be used to describe a hybrid system: by finite-state machine, Petri net, state diagram (UML), statecharts of D. Harel.

To model hybrid systems, various special instrumental tools can be used: the simulation environments (AnyLogic, Arena) or the modeling environments (Simulink, Rand Model Designer). Software of both types is not suitable for modeling physics-and-technics systems, since software of the first type do not allow creating models of continuous behavior of the required level of abstraction, and software of the second type allow integrating into the state diagrams only continuous models written in explicit form (in the form of a program code). The modeling environment MARS [2] is based on the method of component circuits [3] and uses multilevel approach [4] for modeling systems of different nature. MARS combines simulation and analytical modeling methods. Algebraic and differential equations describing a continuous model in MARS can be written in an implicit form. At the moment MARS uses straight algorithmic approach to represent discrete behavior of systems which was suggested in the language of modeling algorithmic constructions (MAC-language) [5]. In turn, this leads to increasing the complexity of the model and decreasing its readability. In this regard, an essential task is integrating new tools into the MAC-language for effective and easy constructing models of DEDS in MARS.

Multilevel approach to modeling in MARS

MARS uses the technology of visual modeling which allows the user to compile a model from the existing typical components of the component model library (CML) and to develop the missing ones using component model generator (CMG) [6]. The cumulative equations system of the constructed multicomponent system is automatically formed and transmitted to the «solver» for the numerical solution. The results of the numerical solution are visualized to the user. MARS uses different visual languages to model discrete, continuous and visual behavior of systems and objects. In the multi-level representation of the model each language corresponds to its level which has its own layer (separate window) in the MARS model editor (Fig. 1). This approach assumes decomposition of an object's computer model into 3 levels: the object scheme level (corresponding to physical and chemical representation of an object's continuous behavior, which is described by means of mathematical modeling), algorithmic level (corresponding to the discrete-event representation of an object's behavior, which is described by the means of simulation) and visual level (corresponding to the external representation of an object's activity and to the user interface). All three levels are displayed in the respective layers of the MARS model editor [8] (Fig. 1).

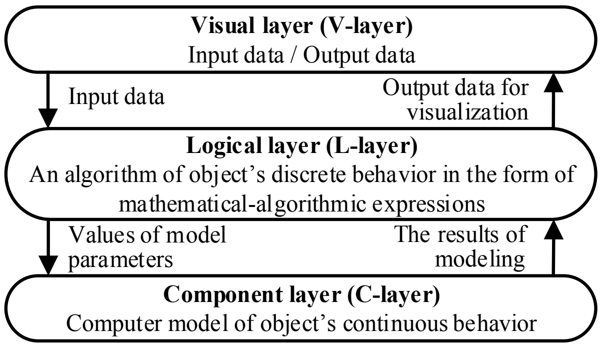
Visual layer (V-layer)
Input data / Output data
Input data
Output data for visualization
Logical layer (L-layer)
An algorithm of object's discrete behavior in the form of mathematical-algorithmic expressions
Values of model parameters
The results of modeling
Component layer (C-layer)
Computer model of object's continuous behavior

Fig. 1. Structure of computer models multilevel representation

Visual layer (the V-layer) is a user interface designed for data input and output in the form of graphs, diagrams or graphic animation. The V-layer is directly intended for controlling the working process of a computer model.

Component layer (the C-layer) contains a mathematical model of a system or an object in the form of component circuits (CC) describing its continuous (physical-chemical) behavior in the form of algebraic-differential equations.

Logical layer (the L-layer) carries out data transfer between the C-layer and the V-layer. Also the L-layer contains an algorithmic CC written in the MAC-language and which defines the algorithm for the discrete (logical) behavior of an object or for an experiment scenario. Thus, an object's continuous behavior described on the C-layer can be supplemented by the discrete-event behavior described on the L-layer.

Language of modeling algorithmic constructions (MAC-language)

The MAC-language is a language used to build computer models on the L-layer. It applies a straight algorithmic approach to construct models describing discrete behavior of objects. This means that the MAC-language has the sets of basic components corresponding to arithmetic operations, cycles, conditional statements and other mathematical-algorithmic constructions. Therefore, modeling the complex behavior of an object will require the compilation of a large number of components and will cause the unreadability of CC that may be not acceptable in many cases. Due to this, integrating more complex algorithmic constructions to describe discrete (logical) behavior of objects of different nature into MAC-language is the essential task. The combination of using such constructions on the L-layer (to describe discrete behavior of objects) and special components on the C-layer (to describe continuous behavior of objects) will enable more effective and easy modeling of the hybrid systems and objects.

Example of a hybrid system model compiled in MARS by means of MAC-language

It is proposed to consider modeling the problem of the car motion along three sections of the road with different acceleration. This problem can be described by the following formulation: «The car moved with the uniform acceleration along three sections of its path. The car passed s_1 km with an acceleration of a_1 m/s^2 then s_2 km with an acceleration of a_2 m/s^2 and s_3 km with an acceleration of a_3 m/s^2. What was the final velocity of the car?».

The continuous behavior of the car in this problem is described by the equation $s(t) = v_0t+at^2/2$ (also it can be described by a second-order linear differential equation or two first-order differential equations: $dx/dt = v$, $dv/dt = a$). The discrete behavior of the car in this problem is characterized by changing numerical value of the parameter «current acceleration» due to the occurrence of a certain event (the car reaches the end of the current section). So the discrete behavior of the car can be described by the following expression (1)

$$a=\begin{cases} a_1, & \text{if} \quad s(t)<s_1, \\ a_2, & \text{if} \quad s_1 \le s(t) \le s_2, \\ a_3, & \text{if} \quad s(t)>s_1+s_2. \end{cases} \tag{1}$$

Thus, the discrete behavior of the car in the problem can be divided into three elementary discrete time intervals in each of which the car has a state represented by a tuple of real numbers named state variables. The algorithmic CC describing the discrete behavior of the car according to expression (1) is presented in Fig 2. The first expression ($s(t)<s_1$) corresponds to the component «1 $s(t)<s_1$», the second one ($s_1 \le s(t) \le s_2$) corresponds to the group of components «2.1 $s(t) \ge s_1$», «2.2 $s(t)<s_2$», «$s_1 \le s(t)<s_2$» and the last one ($s(t)>s_1+s_2$) corresponds to another group of components including «3.1 $s(t) \ge s_2$», «3.2 $s(t)<s_3$», «3.3 $s_2 \le s(t)<s_3$».

The scheme shown in Fig. 2 uses components named *data holders*: «send a1», «send a_2», «send a_3», which transmit the value from the input (the left node) to the output (the right node) at every iteration only if the control response «true» is received (by the above node). If the condition $s(t)<s_1$ becomes true, the component «1 $s(t)<s_1$» sends value «true» at every iteration to the data holder «send a1» which then starts transmitting value a_1 to the CC on the C-layer, which describes continuous behavior of the car. If both conditions 2.1 «$s(t) \ge s_1$» and 2.2 «$s(t)<s_2$» are true, the data holder «send a_2» transmits value a_2 to the C-layer until other both conditions 3.1 «$s(t) \ge s_2$» and 3.2 «$s(t)<s_3$» become true. The model will be stopped when condition 4 «$s_3 \le s(t)$» comes true.

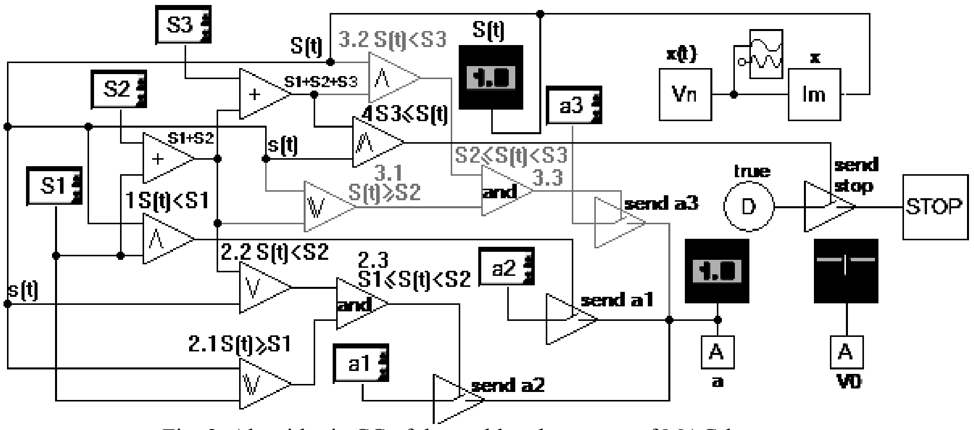

Fig. 2. Algorithmic CC of the problem by means of MAC-language

The given example of the algorithmic CC compiled by means of the MAC-language allows us to infer the necessity of developing and integrating new components for modeling discrete behavior of hybrid systems into the modeling environment MARS.

Statechart diagram

The statechart diagram (or the state diagram) in the Unified Modeling Language (UML) or state machine diagrams (in UML 2.0), which are based on the statecharts of D. Harel [7], represents a conceptual model of an object in the form of a finite-state machine characterizing the behavior of an object as a sequence of its states replacing one another under certain conditions. In the UML the *state* of an object is understood as an element of its behavior during which a certain condition takes place. The *state* can be specified through a set of its properties (relations and attributes) and their current values. There are the following types of actions available for an object in a state: *entry action*, *exit action*, *do activity* and *deferrable events* [8]. There are also special kinds of the state (*pseudostates*): *initial state* and *final state*, intended to indicate the beginning and the end of the procedure for changing states. The *event* is understood in UML as the occurrence of a fact initiating the *transition* which is a procedure of changing the state of the object (changing the values of its attributes). *Transitions* can be *trigger* (conditional) and *non-trigger* (unconditional).

The discrete behavior of the car in the considered physics may be represented by the statechart shown in Fig. 3.

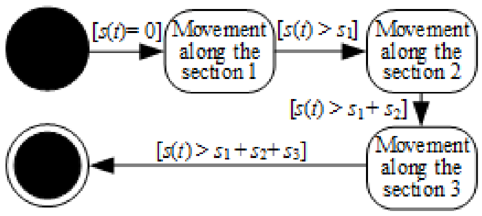
[s(t)= 0]
Movement along the section 1
[s(t) > s1]
Movement along the section 2
[s(t) > s1+s2]
Movement along the section 3
[s(t) > s1+s2+s3]

Fig. 3. Statechart diagram of car behavior

Interpretation of the statechart diagram into MAC-language

In the context of the MAC-language we divide the state into two complementary classes: the *discrete state* and the *continuous state*. We understand the *discrete state* as a set of the object parameters and its mathematical model of continuous behavior that takes place on a discrete time interval during which these parameters are assumed to be unchanged and the certain condition is satisfied. We understand the *continuous state* as set of all object attributes (including variables and parameters) at the current time and its mathematical model defining the law of changing these attributes. On the behavioral scheme (on the L-layer) the *state* through its structural relationships determines the description of the continuous behavior of the object by a mathematical model of the C-layer with given parameters and in the current mode. So the *discrete state* determines not only the quantitative (value of parameters) aspect in the behavior of the object but also the qualitative one (the view of the mathematical model). The so-called *initial* and *final states* consider as a signal to the start and the stop of the computer model execution and refer them to the category of events. By the *event* we mean the change in values of the internal or external model attributes which initiates the *transition* from one discrete state to another one. We consider only trigger transitions and assume that all transitions are conditional.

As a practical result of interpretation the statechart diagram into the MAC-language, we developed the L-layer component named «State» with variable number of nodes to represent the object discrete behavior in the form of a finite-state machine. The graphical view of the component «State» for MARS is shown in Fig. 4.

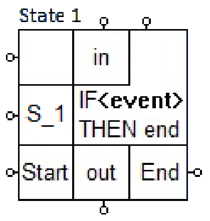

Fig. 4. L-layer component «State»

After receiving value «true» into the node «Start», the component «State» starts transmitting values from node «In» (on the upper side) to the corresponding nodes «Out» (in the bottom side), which then are transmitted at every iteration into C-layer mathematical model through special links marked by the letter «A» on the scheme (Fig. 5).

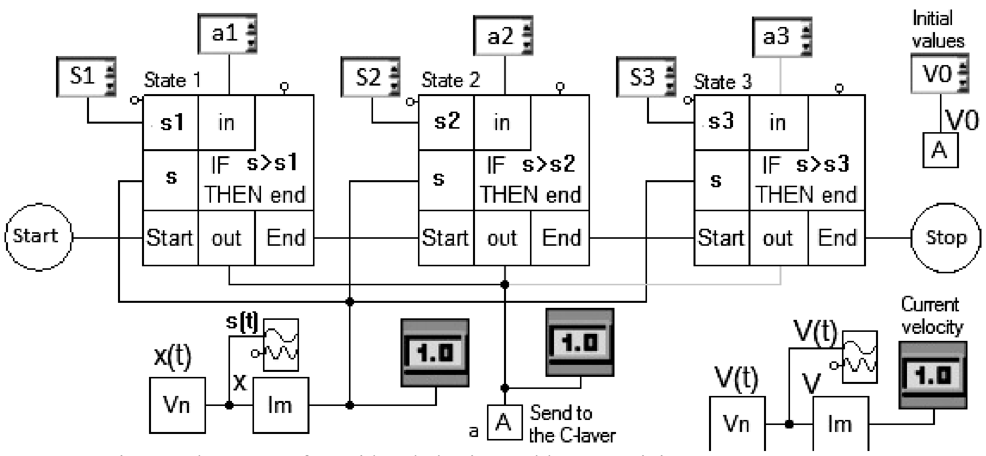

Fig. 5. L-layer CC of considered physics problem containing component «State»

At each iteration the MARS simulation kernel checks the condition written by the user in a symbolic form in the field<event>. If the condition returns «true», the current component stops and transmits the value of «true» to the output node «End», where it is received by the input node «Start» of the next component «State» or by the input node of the component «STOP» which stops the entire model. Thus, only one component «State» on the L-layer and one CC on the C-layer are active at a moment. The result of modeling discrete behavior of the car in the considered physics problem with using new developed component «State» is shown in Fig. 5.

In contrast to the straight algorithmic approach used in the MAC-language (Fig. 2), which is more complex and less understandable, the presented CC explicitly depicts the number of discrete states of the considered object and the conditions of the transition between them.

Conclusion

Interpretation of the statechart diagrams into the MAC-language of the modeling environment MARS will allow modeling a wider range of practical problems as well as will provide an opportunity to build models of discrete-event systems in MARS more easily and efficiently. The suggested approach based on using the statechart diagram simplifies the visual language of the modeling environment MARS and complements the multilevel approach for representing computer models of systems that supposes a decomposition of the system's behavior into discrete, continuous and visual constituents.

References

1. Kolesov Yu.B., Senichenkov Yu.B. Modelirovaniye sistem. Dinamicheskiye i gibridnyye sistemy [Modeling of systems. Dynamic and hybrid systems]. – Saint Petersburg: BKHV-Peterburg Publ., 2012. – 224 p. (In Russian).

2. Dmitriyev V.M., Shutenkov A.V., Zaychenko T.N., Gandzha T.V. MARS – sreda modelirovaniya tekhnicheskikh ustroystv i sistem [MARS – environment for modeling technical devices and systems]. – Tomsk, V-Spektr Publ., 2011. – 277 p. (In Russian).

3. Dmitriyev V.M., Gandzha T.V., Shutenkov A.V. Postroyeniye komp'yuternykh modeley mnogofraktsionnykh fiziko-khimicheskikh sistem gazopromyslovykh ob»yektov v formate metoda komponentnykh tsepey [Computer modeling multifractional physical-chemical systems of gas-field objects in the format of the method of component circuits]. Proceedings of Tomsk State University of Control Systems and Radioelectronics. – 2012. – Vol. 2, No. 1. – P. 145–150 (In Russian).

4. Dmitriyev V.M., Gandzha T.V. Korotina T.Yu. Sistema vizualizatsii i upravleniya vychislitel'nym eksperimentom v srede mnogourovnevogo modelirovaniya MARS [System for visualization and control of computational experiment in the environment of multilevel modeling]. Proceedings of Tomsk State University of Control Systems and Radioelectronics. – 2010. – Vol. 1, No. 2. – P. 149–155 (In Russian).

5. Dmitriyev V.M., Gandzha T.V. Metod i yazyk modelirovaniya intellektual'nykh sistem upravleniya slozhnymi tekhnologicheskimi ob»yektami [The method and the language for modeling intelligent control systems for complex technological objects] // Object systems. – 2015. – No. 10. – P. 44–50 (In Russian).

6. Dmitriyev V.M., Gandzha T.V., Korotina T.Yu. Generator modeley komponentov fizicheski neodnorodnykh tsepey na baze interaktivnoy matematicheskoy paneli [Generator of component models of physically inhomogeneous circuits on the basis of an interactive mathematical panel]. Proceedings of Tomsk State University of Control Systems and Radioelectronics. – 2009. – Vol. 2. – P. 94–99 (In Russian).

7. Harel D. Statecharts: A visual formalism for complex systems // Science of Computer Programming. – 1987. – Vol. 8, No. 3. – P. 231–274.

8. Chunikhin O.Yu. Formal'naya model' mashin sostoyaniy UML [Formal model of UML state machines]. Journal of Applied and Industrial Mathematics. – 2004. – Vol. 7, No. 1. – P. 151–165 (In Russian).

Kochergin M.I.
PhD student, assistant, Department of Computer Control and Design Systems, Tomsk State University of Control Systems and Radioelectronics (TUSUR)
Ph.: +7 (382-2) 41-39-15
E-mail: max24kochergin@gmail.com

М.И. Кочергин

Интерпретация диаграмм состояний в язык многоуровневого моделирования

Рассматривается моделирование гибридных физико-технических систем в среде многоуровневого моделирования МАРС (СМ МАРС). Проводится сравнение общеалгоритмического подхода языка моделирования алгоритмических конструкций (МАК) СМ МАРС к моделированию гибридных систем и подхода с использованием диаграмм состояний. Обосновывается необходимость интерпретации диаграмм состояний в язык МАК СМ МАРС. Описывается разработанный компонент «Состояние» для описания дискретного поведения объектов на логическом слое СМ МАРС.

Ключевые слова: дискретно-непрерывные системы, гибридные системы, компьютерное моделирование, диаграммы состояний, конечный автомат.