Journal of Southeast University (English Edition) Vol. 21, No. 3, pp. 304 – 309 Sept. 2005 ISSN 1003—7985

Modeling method of hybrid systems using extended Petri nets

Zhou Binghai[1] Wang Shijin[1] Tao Lihua[2]

([1] School of Mechanical Engineering, Shanghai Jiaotong University, Shanghai 200030, China)

([2] National Key Laboratory of Flexible Manufacturing System Technology, Changchun 130012, China)

Abstract: In order to model effectively hybrid systems, a new modeling method of extended Petri nets, which is called extended object-orient hybrid Petri net (EOHPN), is proposed. To deal with the complexity of hybrid systems, object-oriented abstraction mechanisms such as encapsulation and classifications are merged into EOHPN models. To combine the continuous part and discrete part of hybrid systems and to reduce the complexity of hybrid systems, a hybrid Petri net is introduced and extended with object-oriented modeling technology. Development of object models is suggested on the basis of the defined EOHPN. Finally, an application-oriented case is presented to illustrate that how the proposed EOHPN is used to model hybrid systems. The resulting model validates that the EOHPNs can deal with the modeling complexity of hybrid systems.

Key words: Petri nets; object-oriented model; hybrid system; modeling

Most practical systems are complex and hybrid in nature. These systems demonstrate the behavior of a continuous dynamic system until they encounter certain abrupt structural or operating condition changes. It is a challenge to model the complexity of a hybrid system, but modeling is the foundation for efficient production planning, control, scheduling, etc. The modeling study of such hybrid dynamic systems has attracted increasing attention in recent years. Several mathematical formal expressions have been proposed in order to represent the model of a complex hybrid system. We can cite hybrid automata, hybrid Petri nets, the Branicky model, Bond-Graphs with commutation, etc[1]. For example, Kontini et al. presented a modeling method of linear hybrid automata in order to verify the performance specification of a hybrid system[2]. In order to model both the continuous/discrete dynamics and the switching between different operating conditions, Giancarlo et al. proposed a framework of mixed logic dynamic systems[3]. In order to model integration time-driven mechanical features of hybrid systems, Thoms et al. presented an approach toward comprehensive discrete-continuous modeling and accurate dynamical simulation of simple manipulation[4]. These approaches are interesting for describing complex hybrid dynamics, i. e., the interactions between the continuous part and the discrete part. However, they do not provide the needed modularity required to represent complex systems made up of many simple interacting hybrid systems. Moreover, these tools have been primarily developed for theoretical purposes and do not allow concise representation of the real physical systems[5].

In order to overcome the weaknesses mentioned above, an extended object-oriented hybrid Petri net (EOHPN) is presented to model a complex hybrid system provided with modularity and practicability. To deal with the complexity, the EOHPN models are constructed in object-oriented abstraction mechanisms such as encapsulation, classification and inheritance. These abstraction mechanisms make the resulting models more compact, less complex, and consequently more manageable. To combine a discrete part and a continuous part of hybrid systems, hybrid Petri nets are merged into the EOHPN models.

1 Definition of EOHPNs

Based on Petri nets, object-oriented Petri nets are introduced in Ref. [6] for modeling complex systems; hybrid Petri nets are introduced in Ref. [7] for modeling systems that combine a discrete part and a continuous part. EOHPNs are developed to manage the complexity involved in the modeling of hybrid systems. From the view of object-orientation, a hybrid system is composed of objects and their interconnection relations. An EOHPN model is developed according to this conception, i. e. an EOHPN model consists of three parts: P_O, R, M_0. Where $P_O = \{P_{O1}, P_{O2}, \ldots, P_{Ok}\}$ is the

Received 2005-02-25.

Foundation items: The National Key Laboratory Program (No. 51458060104JW0316), the National High Technology Research and Development Program of China (863 Program) (No. 2003AA414120).

Biography: Zhou Binghai (1965—), male, doctor, associate professor, bhzhou@sjtu.edu.cn.

set of finite object places, in which P_{Oi}, $i = 1, 2, \ldots, K$ is an object place in an EOHPN model corresponding to a physical object that is described by an encapsulated hybrid Petri net; R is the set of communicating relations between objects that are described by message places between object subnets; M_0 is the initial marking of the EOHPN model.

1.1 Object subnet

An object subnet is defined as a 6-tuple:

$$P_{Oi} = (P_i, T_i, h_i, \tau_i, F_i, M_{0i})$$

where $P_i = P_{Di} \cup P_{Ci} \cup P_{Ei}$ is the set of finite places, P_{Di} is a set of finite discrete places, P_{Ci} is a set of finite continuous places, a continuous place is represented in a drawing as two concentric circles, P_{Ei} are the sets of finite extended places in object place P_{Oi}, P_{Ei} is the union of P_{IMi} and P_{OMi} that are the sets of finite input message places and output message places, respectively. $T_i = T_{Di} \cup T_{Ci}$ is the non empty set of finite transitions, T_{Di} is a set of finite discrete transitions, T_{Ci} is a set of finite continuous transitions, a continuous transition is represented in a drawing as a hollow bar. Where $P_i \cap T_i = \varnothing, P_i \cup T_i \neq \varnothing$. $h_i: P_i \cup T_i \rightarrow \{C, D\}$ indicates for each node if it is a discrete (D) or a continuous (C) node. $\tau_i: T \rightarrow \Gamma^+$ associates with each transition a positive real number d_j, where the time delay associated with a D-transition t_j is d_j. The maximal firing speed associated with a C-transition t_j is $v_j = 1/d_j$. $F_i: P_i \times T_i \cup T_i \times P_i \rightarrow (0, 1)$ is the set of flow relations in the object. $F_i = F_{Ii} \cup F_{Oi}$, where F_{Ii} and F_{Oi} are input incidence mapping and output incidence mapping, respectively. F_{Ii}: $P_i \times T_i \rightarrow \Gamma$ if $h(p_k) = D$; $P_i \times T_i \rightarrow \Gamma^+$ if $h(p_k) = C$. F_{Oi}: $P_i \times T_i \rightarrow \Gamma$ if $h(p_k) = D$; $P_i \times T_i \rightarrow \Gamma^+$ if $h(p_k) = C$. F_{Ii} and F_{Oi} must meet the following condition: If p_k and t_j are such that $h(p_k) = D$ and $h(t_j) = C$, then $F_{Ii}(p_k, t_j) = F_{Oi}(p_k, t_j)$ must be verified, where $F_{Ii}(p_k, t_j)$ and $F_{Oi}(p_k, t_j)$ are input and output flow from p_k to t_j. This ensures marking of D-places to be an integer whatever evaluation occurs. M_{0i} is the set of initial markings of all state and message places inside P_{Oi}.

The marking $M(t)_i$ of object place P_{Oi} at time t can be deduced from marking M_{0i} at time 0 using the following fundamental equation:

$$M(t)_i = M_{0i} + (F_O(p, t) - F_I(p, t)) \cdot \left(\sigma(t) + \int_0^t v(u)\,\mathrm{d}u\right)$$

where $\sigma(t)$ represents the number of times that each D-transition has been fired (discrete interpretation) between initial time and time t. The components associated with C-transitions are equal to zero. The components of $\boldsymbol{v}(u)$ vector represent instantaneous firing speeds associated with C-transitions. D-transitions are equal to zero. This equation separates the discrete evolution from the continuous one. It represents a trajectory in the marking space. In line with the approach considered for computing the instantaneous firing speed vector, many models have been proposed[8]. In this paper, only the constant speed continuous Petri net is considered[9].

1.2 Interconnection relations of objects

In an EOHPN, the interconnection between the objects depends on their extended places of objects (input message places and output message places), i. e., if $P_{Ei} \cap P_{Ej} \neq \varnothing$, then the two object places P_{Oi} and P_{Oj} have communication. The interconnection relations of objects at the system level may be realized by a set of gates. Mathematically, the interconnection of objects is defined as follows:

$$R = \{R_{ij}, i, j = 1, 2, \ldots, I, i \neq j\}$$

where R_{ij} specifies the message passing relations between the sending object place P_{Oi} and the receiving object place P_{Oj}.

$$R_{ij} = \{A_{Oij}, G_{ij}, A_{Iij}, E_{ij}\}$$

where G_{ij} is the set of finite special transitions called gates that are located between output message places P_{OMi} of P_{Ei} and input message places P_{IMj} of P_{Ej}. The union of all the gates associated with R is the set of gates in the system. A_{Oij} is the set of finite output connection arcs from P_{OMi} of P_{Ei} to G_{ij}; A_{Iij} is the set of finite input connection arcs from P_{IMj} of P_{Ej}; E_{ij} is the expression function of connection arcs between P_{OMi} and P_{IMj}.

$$E_{ij} = [\mathrm{IAF}(A_{Iij}, G_{ij}), \mathrm{OAF}(A_{Oij}, G_{ij})]$$

where $\mathrm{IAF}(A_{Iij}, G_{ij})$ is the input expression function for an arc which connects gate G_{ij} to the input message place A_{Iij} in P_{Oj}, and $\mathrm{OAF}(A_{Oij}, G_{ij})$ is the output expression function for an arc which connects the output message place A_{Oij} in P_{Oi} to gate G_{ij}.

2 Development of Object Subnet Models

2.1 Description of physical objects

The semiconductor wafer fabrication system (SWFS) is one of the most typical complex hybrid systems. In order to illustrate how EOHPN models are developed for a hybrid system, a typical SWFS is considered in this paper. In general, the typical SWFS is composed of processing machines (e. g., cutting machines, dressing machines, etc.), varnishing machines, heat treatment facilities, material handling systems (MHSs), WIP buffers, other auxiliary facilities, etc. To establish object subnet models of physical objects in the typical SWFS, all physical objects are mainly classified as five main objects:

① Processing machine object (PMO) represents a general processing machine (PM), which has an input stocker and an output stocker. A PM draws wafers for processing from the input stocker. Before the PM starts processing, a minimal level of B wafers (a lot) in the input stocker is required.

② Varnishing machine object (VMO) represents a general varnishing machine (VM), which has an input stocker and an output stocker. A VM starts processing only when all lot (batch) wafers have reached the input stocker. The VM once processes a lot regardless of the type of components.

③ Heat treatment facility object (HTFO) represents a heat treatment facility (HTF), which has an input stocker and an output stocker. When wafers arrive in the input stocker, they are grouped in boxes, which can contain at most Q wafers. In general, when the box is full, the heat treatment starts. Treatment duration (time) is the same regardless of the box state. Different lots cannot be contained in the same box. The last box is not necessarily full[7].

④ Transporter object (TO) represents MHSs equipment (e. g., AGVs, industrial robots) and operators, which transport wafers among WIP buffers and machines/facilities.

⑤ WIP buffer object (WBO) represents a general WIP buffer, which accommodates WIP wafers waiting for processing in the current system.

2.2 Development of object models

A PMO model abstracts the behavior of the majority of PMs. The PMO subnet is shown in Fig. 1. Each place and transition has the meaning given in Tab. 1 and Tab. 2. Input message place p_{im11} receives the message, which requests a lot of wafers to be loaded into an input stocker for processing. The input stocker (place p_{12}) starts to be continuously fed by launching a lot which contains B wafers and a lot is launched every d time units. The token in place p_{11} is reserved for firing transition t_{11}. After d time units, transition t_{11} is fired and the PM starts processing. The number of wafers in the input stocker decreases by the PM rate which equals V. When the marking of p_{13} (the output stocker) reaches B, all the lot wafers will have been processed. The transition t_{13} is instantaneously fired. The token is removed from p_{14} and put into p_{15}, the PM returns back to the idle state and the output message place p_{om11} sends the message, which requests a lot of wafers to be unloaded from the output stocker to the WIP buffer.

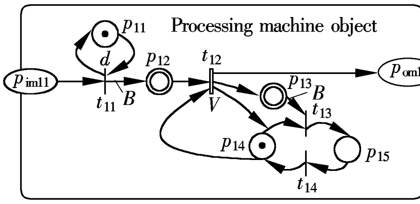

Fig. 1 Object subnet model of the PMO

Tab. 1 Definition meanings of places in PMO subnet

Place	Meanings
p_{11}	Input stocker is available
p_{12}	A lot waits to be loaded into the input stocker
p_{13}	A lot waits to be loaded into the output stocker
p_{14}	When a token is here, the PM is processing
p_{15}	When a token is here, the PM is idle and available
p_{im11}	A lot requests to be loaded into the input stocker
p_{om11}	A lot requests to be unloaded to the WIP buffer

Tab. 2 Definition meanings of transitions in PMO subnet

Transition	Meanings
t_{11}	A lot is entering the input stocker
t_{12}	A lot is being processed at speed V
t_{13}	A lot has been processed, the PM is idle
t_{14}	A lot is ready to be processed

A VMO model abstracts the behavior of general vanishing machines. The VMO subnet is shown in Fig. 2. Each place and transition has the meaning given in Tab. 3 and Tab. 4.

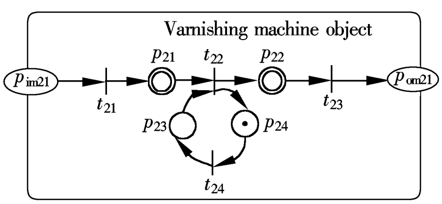

Fig. 2 Object subnet model of the VMO

Tab. 3 Definition meanings of places in VMO subnet

Place	Meanings
p_{21}	A lot waits to be loaded into the input stocker
p_{22}	A lot waits to be loaded into the output stocker
p_{23}	When a token is here, the VM is being used
p_{24}	When a token is here, the VM is idle and available
p_{im21}	A lot requests to be loaded into the input stocker
p_{om21}	A lot requests to be unloaded to the WIP buffer

Tab. 4 Definition meanings of transitions in VMO subnet

Transition	Meanings
t_{21}	A lot is entering the input stocker
t_{22}	A lot is being varnished
t_{23}	A lot has been varnished, the VM is idle
t_{24}	A lot is ready to be varnished

An HTFO model abstracts the behavior of heat treatment facilities. The HTFO subnet is shown in Fig. 3. Each place and transition has the meaning given in Tab. 5 and Tab. 6.

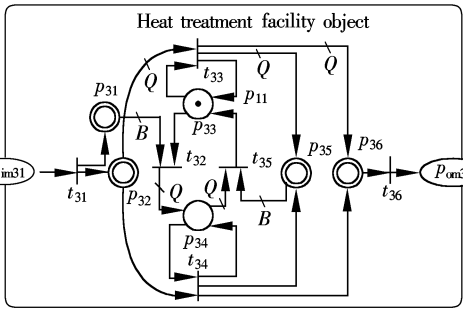

Fig. 3 Object subnet model of the HTFO

Tab. 5 Definition meanings of places in HTFO subnet

Place	Meanings
p_{31}	Count the number of entering wafers
p_{32}	Lots wait to be loaded into the input stocker
p_{33}	When Q tokens are here, the HTF is treating
p_{34}	When the last tokens are here, the HTF is treating
p_{35}	Count the number of treated wafers
p_{36}	Lots wait to be loaded into the output stocker
p_{im31}	Lots request to be loaded into the input stocker
p_{om31}	Lots request to be unloaded to the WIP buffer

Tab. 6 Definition meanings of transitions in HTFO subnet

Transition	Meanings
t_{31}	Lots are entering the input stocker
t_{32}	The last lot tokens enter, p_{34} is marked
t_{33}	Boxes are full, lots are being treated
t_{34}	The last lot box is being treated
t_{35}	The Q tokens enter, p_{33} is marked
t_{36}	The lots have been treated, the HTF is idle

A TO model abstracts the behavior of majority transportation equipment. The TO object subnet is shown in Fig. 4. Each place and transition has the meaning given in Tab. 7 and Tab. 8.

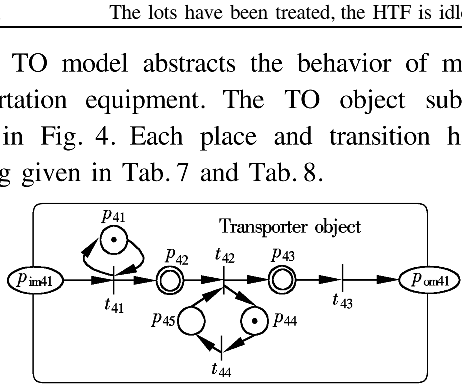

Fig. 4 Object subnet model of the TO

Tab. 7 Definition meanings of places in TO subnet

Place	Meanings
p_{41}	Transporter' carrier is available
p_{42}	Lots wait to be loaded into the carrier
p_{43}	Lots wait to be unloaded from the carrier
p_{44}	When a token is here, the transporter is idle
p_{45}	When a token is here, the transporter is being used
p_{im41}	Lots request to be loaded into the carrier
p_{om41}	Lots request to be unloaded from the carrier

Tab. 8 Definition meanings of transitions in TO subnet

Transition	Meanings
t_{41}	Lots are being loaded into the transporter carrier
t_{42}	Lots are being transported
t_{43}	Lots wait to be unloaded
t_{44}	Lots are ready to be transported

A WBO model abstracts the behavior of general WIP buffers in the system. The WBO object subnet is shown in Fig. 5. Each place and transition has the meaning given in Tab. 9 and Tab. 10.

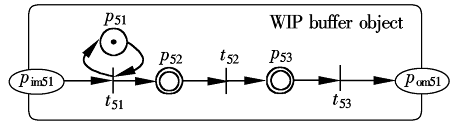

Fig. 5 Object subnet model of the WBO

Tab. 9 Definition meanings of places in WBO subnet

Place	Meanings
p_{51}	Count the lot numbers in WIP buffer
p_{52}	Lots wait to be loaded/unloaded
p_{53}	Lots are loaded/unloaded operation
p_{im51}	Lots request to be loaded/unloaded
p_{om51}	Send the message to finish loading/unloading

Tab. 10 Definition meanings of transitions in WBO subnet

Place	Meanings
t_{51}	Lots are waiting to be loaded/unloaded
t_{52}	Lots are being loaded/unloaded
t_{53}	Lots are loaded/unloaded to/from the WIP buffer

3 Application-Oriented Development of EO-HPN Models

Construction of an EOHPN model is derived from the object subnet models. First each encapsulated object subnet is represented as an object place and the message passing gate as an activity transition, then the internal behavior (i. e., states of the object subnet) of each physical object can be represented by the associated tokens. The procedure of constructing an EOHPN model is summarized as follows:

Step 1 Represent each encapsulated object subnet as an object place $P_{Oi}(i=1,2,\ldots,m)$.

Step 2 Represent each gate involved in an EO-HPN model as an activity transition $G_j(j=1,2,\ldots,n)$. A gate is connected with its input message places with an and (or) relation if it can be fired when all (some/any of) the connected message places have the required specific tokens. A gate is connected with its output message places with an and (or) relation if the tokens will be moved to all (some of) the connected message places.

Step 3 Represent the [P_{Oi}—gate—P_{Oj}] message passing relations.

Step 4 Check whether the EOHPN reflects the hybrid system operation concisely and if not, modify the EOHPN until it models the system.

To illustrate the procedure of constructing an EOHPN model, we consider the following example: The workshop consists of a cutting machine, a dressing machine, a varnishing machine, a molding machine, a heat treating facility, a testing machine, three WIP buffers and two transporters. Wafers travel the workshop, which are produced onto components such as transistors or diodes. The hybrid system is characterized by a unique routing of different components: cutting→dressing→varnishing→molding→heat treating →testing. The WIP buffer 1 is used to keep wafers for the cutting machine and dressing machine. The WIP buffer 2 is used to keep wafers for the varnishing machine and molding machine. The WIP buffer 3 is used to keep wafers for the heat treating facility and testing machine. The cutting machine, dressing machine, molding machine and testing machine are abstracted to processing machine objects. Each encapsulated object subnet model and gate in the OHPN model shown in Fig. 6 corresponding to object places (P_{O1}—cutting machine, P_{O2}—dressing machine, P_{O3}—varnishing machine, P_{O4}—molding machine, P_{O5}—heat treating facility, P_{O6}—testing machine, P_{O7}—transporter 1, P_{O8}—transporter 2, P_{O9}—WIP buffer 1, P_{O10}—WIP buffer 2, P_{O11}—WIP buffer 3), and activity transition $G_j (j = 1, 2, \ldots, 5)$. The "OR" relation exists among the inputs to gates G_2, G_3, G_4, G_5 and among the gates G_{11}, G_{12}, G_{13}, G_2, G_3, G_4, G_5 to outputs. Here, due to limited space, the detailed procedure of constructing steps is omitted.

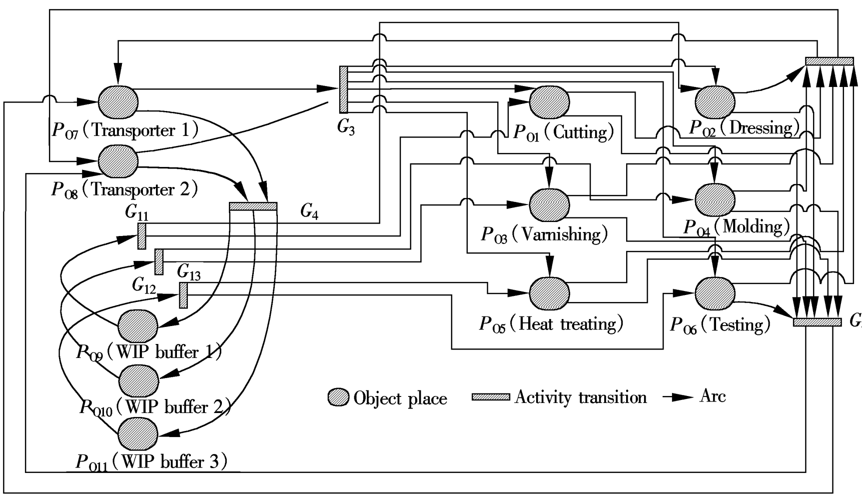

Fig. 6 EOHPN model for a workshop of typical SWFSs

4 Conclusion

Extended Petri nets have recently emerged as a powerful tool for modeling of hybrid systems. To systematically construct a model of the hybrid system, a general system modeling method has been proposed on the basis of the extended object-oriented Petri net. The proposed model method has been illustrated with an application-oriented case.

Since deadlock-free and conflict resolving strategies are very important for the hybrid systems, the ongoing works include the methods for detecting deadlocks and conflicts in an EOHPN model. A multiple-objective scheduling algorithm and real-time dispatching policies will also be integrated into an EOHPN model. Based on this work, the results obtained through EOHPN modeling can be used in system performance analysis, scheduling, control and simulation.

References

[1] Flaus J M. Modeling and analysis of hybrid dynamical systems: a beginner's guide [J]. *Japan Electronics Show Association*, 1998, **32**(7, 8): 797 – 830.

[2] Kotini I, Hassapis G, Mavridis L. Verification of hybrid system specifications using linear hybrid automata [J]. *Computational Methods in Circuits and Systems Applications*, 2003. 171 – 176.

[3] Giancarlo F T, Eduardo G, Matteo S, et al. Modeling and control of co-generation power plants: a hybrid system approach [J]. *IEEE Transactions on Control Systems Technology*, 2004, **12**(5): 694 – 705.

[4] Thoms S, Martin B, Gunther S. A hybrid system approach toward modeling and dynamical simulation of dextrous manipulation [J]. *IEEE/ASME Transactions on Mechatronics*, 2003, **8**(3): 352 – 361.

[5] Thevenon L, Flaus J M. Modular representation of com-

plex hybrid systems: application to the simulation of batch processes [J]. *Simulation Practice and Theory*, 2000, **8** (3): 283 - 306.

[6] Lee Y K, Park S J. Opnets: an object-oriented high-level Petri net model for real-time system modeling [J]. *Journal of System Software*, 1993, **20**(1): 69 - 86.

[7] Allam M, Alla H. Modeling and simulation of an electronic component manufacturing system using hybrid Petri nets [J]. *IEEE Transactions on Semiconductor Manufacturing*, 1998, **11**(3): 374 - 383.

[8] Allam M. *Petri nets and grafcet* [M]. London: Prentice-Hall, 1992.

[9] Zerhouni N, Alla H. Dynamic analysis of manufacturing systems using continuous Petri nets [A]. In: *Proc* 1990 *IEEE International Conference on Robot Automation* [C]. Cincinnati, 1990. 1070 - 1075.

基于扩展 Petri 网的混合系统建模方法

周炳海[1]　王世进[1]　陶丽华[2]

([1] 上海交通大学机械与动力工程学院,上海 200030)
([2] 柔性制造系统技术国家重点实验室,长春 130012)

摘要:为了对混合系统进行有效地建模,提出了一种扩展的面向对象混合 Petri 网(EOHPN)建模方法. 针对混合系统的复杂性,在 EOHPN 模型中融入了面向对象的抽象机制,例如封装和类定义. 为了结合混合系统的连续部分和离散部分,减少混合系统的复杂性,引入混合 Petri 网并用面向对象建模技术作了扩展. 在定义 EOHPN 模型的基础上,拓展了对象模型. 最后,用实例描述了基于 EOHPN 的混合系统建模过程,同时验证了 EOHPN 模型在处理复杂混合系统建模时是有效的.

关键词: Petri 网;面向对象模型;混合系统;建模
中图分类号:TP393