Equivalent Semantic Translation from Parallel DEVS Models to Time Automata

Shoupeng Han and Kedi Huang

College of Mechaeronics Engineering and Automation,
National University of Defense Technology, 410073, Changsha Hunan, China
hspself@163.com

Abstract. Dynamic reconfigurable simulation based on Discrete Event System Specification (DEVS) requires efficient verification of simulation models. Traditional verification method of DEVS model is based on I/O test in which a DEVS model is regarded as a black box or a grey box. This method is low efficient and insufficient because input samples are often limited. This paper proposes a formal method which can translate Parallel DEVS model into a restrict kind of Timed Automata (TA) with equivalent behaviors. By this translation, a formal verification problem of Parallel DEVS model can be changed into the formal verification of according timed automata.

Keywords: Discrete Event System Specification (DEVS), Timed Transition System (TTS), Semantic Equivalence, Timed Automata (TA).

1 Introduction

One of the intended consequences of utilizing simulations in dynamic, data-driven application system (DDDAS) [1] is that the simulations will adjust to new data as it arrives. These adjustments will be difficult because of the unpredictable nature of the world and because simulations are so carefully tuned to model specific operating conditions. Accommodating new data may require adapting or replacing numerical methods, simulation parameters, or the analytical scientific models from which the simulation is derived. These all require simulation modules in DDDAS to be dynamically reconfigurable. Furthermore, the substitutions must be verified before they are injected into the running simulation so that they can behave required properties. So, how to verify simulation system and simulation component more efficiently is a critical problem for dynamic reconfigurable simulation. Formal verification is a kind of method which can search the whole state space in acceptable time. In contrast to traditional verification method based on I/O test, it has many advantages, such as the verification process can be automated and the resources are more limited etc. This paper will propose a formal verification method for models formalized with discrete event system specification (DEVS) [2] [3] which is a representative formalism in modeling time varying reactive discrete systems. This method is based on trace equivalence which can change the verification problem of DEVS model to the verification of TA [4] with equivalent behaviors.

Y. Shi et al. (Eds.): ICCS 2007, Part I, LNCS 4487, pp. 1246 – 1253, 2007.

2 Formal Semantic for Parallel DEVS Model

Though Parallel DEVS has a well-defined syntax, its behavior is given as a kind of abstract simulator informally. In order to verify Parallel DEVS models formally, a formal behavior semantic of it must be defined at first [5] [6]. In this paper, Timed Transition System (TTS) is selected as the semantic base of DEVS models' behaviors.

2.1 Parallel DEVS [2]

Parallel DEVS formalism consists of two parts, atomic and coupled models. An atomic model is a structure:

$$M = < X,\ Y,\ S,\ s_0,\ \delta_{\text{int}},\ \delta_{ext},\ \delta_{con},\ \lambda,\ ta > \tag{1}$$

With

$X = \{(p,v) \mid p \in IPorts,\ v \in X_p\}$ is the set of input events, *IPorts* is the set of input ports, and X_p is the range of values of port p;

$Y = \{(p,v) \mid p \in OPorts,\ v \in Y_p\}$ is the set of output events, *OPorts* is the set of output ports, and Y_p is the range of values of port p;

S is the set of states, s_0 is the initial state;

$\delta_{\text{int}} : S \to S$ is the internal transition function;

$\delta_{ext} : Q \times X^b \to S$ is the extern transition function, X^b is a set of bags over elements in X, $Q = \{(s,e) \mid s \in S, 0 < e < ta(s)\}$, e is the elapsed time since last state transition;

$\delta_{con} : S \times X^b \to S$ is the confluent transition function, subject to $\delta_{con}(s,\phi) = \delta_{int}(s)$;

$ta : S \to R^+_{0,\infty}$ is the time advance function, $R^+_{0,\infty}$ refers to nonnegative real value;

Atomic models may be coupled in the DEVS formalism to form a coupled model. A coupled model tells how to connect several component models together to form a new model. A coupled model is defined as follows:

$$N =< X, Y, D, \{M_i \mid i \in D\}, \{I_i\}, \{Z_{i,j}\} > \tag{2}$$

where,

X is a set of input events;

Y is a set of output events;

D is a set of components names;

For each i in D, M_i is a component model; I_i is the set of influencees for i;

For each j in I_i, $Z_{i,j}$ is the *i*-to-*j* output translation function. It includes three cases: (1) $Z_{i,j} : X \to X_j$, *if* $i = N$; (2) $Z_{i,j} : Y_j \to Y$, *if* $j = N$; (3) $Z_{i,j} : Y_i \to X_j$, *if* $i \neq N$ *and* $j \neq N$. Above three functions are also called *EIC*, *EOC* and *IC*。

2.2 Timed Transition System (TTS)

A Timed Transition system T_t is represented as a 5-tuple:

$$T_t =< S, init, \Sigma, D, T > \tag{3}$$

where

S is a possibly infinite set of states;

init is an initial state;

$\Sigma = Act \bigcup R_0^+$ is the alphabet, where Act is a set of discrete actions;

D is a set of discrete transitions, noted $s \xrightarrow{x} s'$, where $x \in \Sigma$ and $s, s' \in S$, asserting that "from state s the system can instantaneously move to state s' via the occurrence of the event x";

T is a set of time-passage transitions, noted $s \xrightarrow{d} s'$, where $s, s' \in S$, asserting that "from state s the system can move to state s' during a positive amount of time d in which no discrete events occur". Time-passage transition is assumed to satisfy two axioms:

Axiom 1: If $s \xrightarrow{d} s'$ and $s' \xrightarrow{d'} s''$, then $s \xrightarrow{d+d'} s''$;

Axiom 2: Each time-passage step $s \xrightarrow{d} s'$ has a trajectory.

The trajectory used usually is I -trajectory. A I -trajectory is defined as a function $\omega : I \to S$, where I is a closed interval of real value beginning with 0. ω also satisfy following property: for $\forall d, d' \in I$, if $d < d'$, then $\omega(d) \xrightarrow{d'-d} \omega(d')$. We often use $\omega.ltime$ to represent the supremum of I, $\omega.fstate$ to denote the first state $\omega(0)$ in I, and $\omega.lstate$ to denote the last state $\omega(\omega.ltime)$ in I. So, there is a $[0, d]$-trajectory with transition $s \xrightarrow{d} s'$, where $\omega.fstate = s$ and $\omega.lstate = s'$.

Timed execution fragment: Given a TTS, its timed execution fragment is a finite altering sequence $\gamma = \omega_0 a_1 \omega_1 a_2 \omega_2 \cdots a_n \omega_n$, where ω_i is a I -trajectory, a_i is a discrete event and $\omega_i.lstate \xrightarrow{a_{i+1}} \omega_{i+1}.fstate$. The length and initial state of γ is noted as $\gamma.ltime$ and $\gamma.ltime$, where $\gamma.ltime = \Sigma_i \omega_i.ltime$ and $\gamma.fstate = \omega_0.fstate$. All possible timed execution fragments of T_t is described as a set $execs(T_t)$.

Timed trace: Every timed execution fragment $\gamma = \omega_0 a_1 \omega_1 a_2 \omega_2 \cdots a_n \omega_n$ has an according timed trace noted $trace(\gamma)$, which is defined as an altering sequence consists of pairs noted $(a_i, \omega_i.ltime)$, where the orders of a_i in $trace(\gamma)$ is just the same as they occur in γ. All possible timed traces of T_t is described as $traces(T_t)$.

2.3 Semantic of DEVS Model Based on TTS

Behavior semantic of atomic Parallel DEVS model can be described by timed execution fragment and time trace. For a Parallel DEVS model defined in equation (1), its execution fragment is a finite altering sequence $\gamma = \omega_0 a_1 \omega_1 a_2 \omega_2 \cdots a_n \omega_n$ which includes two kinds of transitions:

Time-passage transition: each ω_i in γ is a map from interval $I_i = [0, t_i]$ to global state space of D. For $\forall j, j' \in I_i \mid j < j'$, if $\omega_i(j) = (s, e)$, then $\omega_i(j') = (s, e + j' - j)$.

Discrete event transition: discrete events in parallel can be divided into two categories: input and output events. According to the execution of DEVS, when they occur concurrently, output events must be dealt first. As shown in abstract simulator of DEVS, when an atomic model is going to send output, it will receive a event "$*$" first, "$*$"event and output event can been seen as a pair because they must be concatenated simultaneity; if there is no input event when output event occurs, the component will receive an empty event ϕ. If $(s, e) = \omega_{i-1}(\sup(I_{i-1}))$ and $(s', e') = \omega_i(\inf(I_i))$, one of following conditions should be satisfied:

$$a_i = *, e = ta(s),\ (s', e') = (s, e) \text{ and } \lambda(s) \in Y \tag{3.1}$$

$$a_i \in X, \delta_{ext}(s, e, a_i) = s',\ e' = 0 \text{ and } e < ta(s) \tag{3.2}$$

$$a_i \in X, \delta_{con}(s, a_i) = s',\ e' = 0 \text{ and } e = ta(s) \tag{3.3}$$

$$a_i = \phi, \delta_{int}(s) = s',\ e' = 0 \text{ and } e = ta(s) \tag{3.4}$$

If two timed systems have the same timed trace set, they have the same timed behavior set obviously. The behavior semantic of Parallel DEVS model can be described by a TTS which has the same timed trace set of the DEVS model. For a DEVS model $M = < X, Y, S, s_0, \delta_{\text{int}}, \delta_{ext}, \delta_{con}, \lambda, ta >$, there is an translation noted T_{ts} upon it, which can translate M into a semantic equivalent TTS:

$$T_{ts}(M) =< S_M, init_M, \Sigma_M, D_M, T_M > \tag{4}$$

with:

$S_M = Q_M = \{(s, e) \mid s \in S, 0 \le e \le ta(s)\}$ is the state set, it is equal to the global state space of M;

$init_M = (s_0, 0)$ is the initial state;

$\Sigma_M = (X \bigcup \{\phi, *\}) \bigcup Y \bigcup R_0^+$ is the alphabet;

D_M is the discrete event transition, for $\forall x \in ((X \bigcup \{\phi\}) \bigcup Y)$, it is defined as:

$D_M = \{(s, e) \xrightarrow{x} (s', 0) \mid (s, e)$ and s' satisfy the condition described in equation(3)$\}$

T_M is the time-passage transition, for $\forall d \in R_0^+$, it is defined as:

$$T_M = \{(s, e) \xrightarrow{d} (s, e') \mid (s, e) \in Q_M, e' = e + d, 0 \le e' \le ta(s)\}$$

Obviously, $T_{ts}(M)$ has the equivalent semantic with M.

3 Semantic of Coupled Parallel DEVS Model Based on TA

3.1 Timed Automata

A timed automata consists of a finite automata augmented with a finite set of clock variables, and transitions with clock constraints, additionally the locations can contain local invariant conditions.

For a set of clocks noted X, clock constraint over it is defined by: $\varphi := x \sim c \mid x - y \sim c \mid \neg\varphi \mid \varphi_1 \wedge \varphi_2$, where x and y are clocks in X, c is a nonnegative real value integer constant, φ_1 and φ_2 are clock constraints. The set of all constraints over X is noted $\Phi(X)$.

A clock assignment over a set X of clocks is a function $v: X \to R^+_{0,\infty}$, it assign each clock a real value. If a clock constraint φ is true for all clocks under a clock assignment v, we call v satisfy φ, noted $v \in \varphi$. For $Y \subseteq X$, $[Y \mapsto t]v$ means each clock $x \in Y$ is assigned a value t, while each clock in $X - Y$ satisfy v.

A timed automata is defined as a 6-tuple: $A = < L, l_0, \Sigma, C, I, E >$, where L is a finite set of location; l_0 is an initial location; Σ is a finite alphabet; C is a finite set of clock; $I: L \to \Phi(X)$ is a map, it assign each $l \in L$ a clock constraint in $\Phi(X)$; $E \subseteq L \times \Sigma \times \Phi(C) \times 2^C \times L$ is a set of transition, $< l, a, \varphi, \lambda, l' >$ means a transition from l to l' when an action a occurs and φ is satisfied by all clocks, $\lambda \subseteq C$ is the set of clocks needed to be reset when the transition occurs. $< l, a, \varphi, \lambda, l' >$ is also noted $l \xrightarrow{a, \varphi, \lambda} l'$.

For a timed automata $A = < L, l_0, \Sigma, C, I, E >$, its semantic is defined as a TTS:

$$T_t = < S_A, init_A, \Sigma_A, D_A, T_A >$$

where

S_A is a set of state which consists of a pair (l, v), where $l \in L$ and $v \in I(l)$;

$init_A = (l_0, v_0)$, $v_0 \in I(l_0)$ is a clock assignment, and for each $x \in C$, $v_0(x) = 0$;

$\Sigma_A = \Sigma \bigcup R^+_{0,\infty}$ is the alphabet;

T_A is a set of time-passage transition, for a state (l, v) and a nonnegative real value $d \geq 0$, if each $0 \leq d' \leq d$ satisfies $(v + d') \in I(l)$, then $(l, v) \xrightarrow{d} (l, v + d)$;

D_A is a set of discrete event transition, for a state (l, v) and a transition $< l, a, \varphi, \lambda, l' >$, if $v \in \varphi$, then $(l, v) \xrightarrow{a} (l', [\lambda \mapsto 0]v)$ and $[\lambda \mapsto 0]v \in I(l')$.

3.2 From Atomic Finite DEVS to TA

TA has more expressiveness than DEVS model to model timed systems. In this section, we will seek to find a translation function T_{aut} which can translate a Parallel DEVS model M to a trace equivalent TA noted $T_{aut}(M)$. As shown in Fig.1, TTS is selected as the common semantic base for these two models. Because infinite DEVS model cannot be formal verified for its infinite global state space, we only consider finite Parallel DEVS here, the finiteness of DEVS model include two aspects: first, the number of states is finite; second, the number of external transitions should be also finite, i.e. each external transition function should be a finite piecewise function.

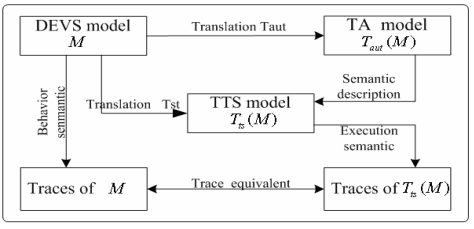

Fig. 1. Illustration of semantic equivalence between DEVS model M and its according TA model $T_{aut}(M)$. TTS model is the middle base for this equivalence.

For an atomic DEVS model $M = < X, Y, S, s_0, \delta_{\text{int}}, \delta_{ext}, \delta_{con}, \lambda, ta >$, there is an equivalent semantic translation noted T_{aut} upon it, which can translate a DEVS model into a semantic equivalent TA (shown in Fig.2):

$$T_{aut}(M) =< L_M, l_0(M), \Sigma_M, C_M, I_M, E_M > \tag{5}$$

with

$L_M = S$ is a set of location;

$l_0(M) = s_0$ is an initial location;

$\Sigma_M = \{X \bigcup \{\phi, *\}\} \bigcup Y$ is an alphabet;

C_M is a clock set, because there is a time variable e in atomic DEVS model, so there is also a clock in C_M, noted $C_M = \{e\}$;

$I_M : L_M \to \Phi(C_M)$ is a map, for $\forall l \in L_M$, $I_M(l) : 0 \le e < ta(l)$;

$E_M \subseteq L_M \times \Sigma_M \times \Phi(C_M) \times 2^{C_M} \times L_M$ is the set of transition, for $(l \xrightarrow{a,\varphi,\kappa} l') \in E_M$, it includes three cases:

(1) If $a = *$, then $(\varphi : e = ta(l)) \wedge (\kappa = \{\}) \wedge (l' = l))$; (6.1)

(2) If $a \in X$, then $(\varphi : 0 < e < ta(l)) \wedge (\kappa = \{e\}) \wedge (l' = \delta_{ext}((l,e),a))$ or $(\varphi : e = ta(l)) \wedge (\kappa = \{e\}) \wedge (l' = \delta_{con}(l,a))$; (6.2)

(3) If $a = \phi$, then $(\varphi := e = ta(l)) \wedge (\kappa = \{e\}) \wedge (l' = \delta_{int}(l))$; (6.3)

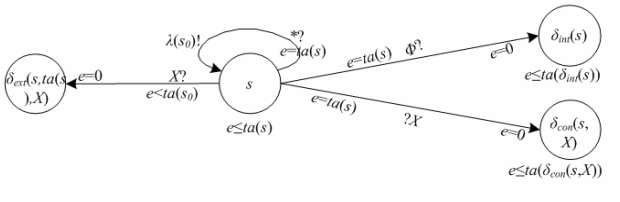

Fig. 2. Illustration of the translation from DEVS model M to its according TA model $T_{aut}(M)$. Where "!" refers to output event, "?" refers to received event, "*" refers to an internal event sent by the coordinator of M, which will be discussed in section3.3.

It is easy to find that the semantic of $T_{aut}(M)$ is just also the TTS $T_{ts}(M)$ defined in equation (4). Thereby, the semantic of $T_{aut}(M)$ is equal to the semantic of M.

3.3 From Coupled DEVS Model to Composition of TAs

Given two TAs $A_1 =< L_1, l_1^0, \Sigma_1, C_1, I_1, E_1 >$, $A_2 =< L_2, l_2^0, \Sigma_2, C_2, I_2, E_2 >$ and two sets of clock C_1 and C_2 don't intersect, the parallel composition of them is a TA noted $A_1 \| A_2 =< L_1 \times L_2, l_1^0 \times l_2^0, \Sigma_1 \bigcup \Sigma_2, C_1 \bigcup C_2, I, E >$, where $I(l_1, l_2) = I(l_1) \wedge I(l_2)$ and transition E is defined as follows:

(1) For $\forall a \in \Sigma_1 \bigcap \Sigma_2$, if transition $< l_1, a, \varphi_1, \lambda_1, l_1^{'} >$ and $< l_2, a, \varphi_2, \lambda_2, l_2^{'} >$ exist in E_1, E_2 respectively, then $< (l_1, l_2), a, \varphi_1 \wedge \varphi_2, \lambda_1 \bigcup \lambda_2, (l_1^{'}, l_2^{'}) >$ is included in E;

(2) For $\forall a \in \Sigma_1 \setminus \Sigma_2$, transition $< l_1, a, \varphi_1, \lambda_1, l_1^{'} >$ in E_1 and $l_2 \in L_2$, transition $< (l_1, l_2), a, \varphi, \lambda_1, (l_1^{'}, l_2) >$ is included in E;

(3) For $\forall a \in \Sigma_2 \setminus \Sigma_1$, transition $< l_2, a, \varphi_2, \lambda_2, l_2^{'} >$ in E_2 and $l_1 \in L_1$, transition $< (l_1, l_2), a, \varphi_2, \lambda_1, (l_1, l_2^{'}) >$ is included in E;

The composition of DEVS model is completed by a coordinator, which has the same interface as an atomic DEVS model and it can also be included in another coordinator. The communication among DEVS models is a weak synchronization relationship which is controlled by the coordinator: when time elapsed equal to the time advancement, state transition of child model must occurs in parallel after all of them have sent the output. This weak synchronization can be depicted by channels in TA, where a channel is signed by "?" and "!" event with the same event name. In this way, the coupled model of a coordinator and its child models have just the same behavior as parallel composition of TAs (shown in Fig.3.) according with them.

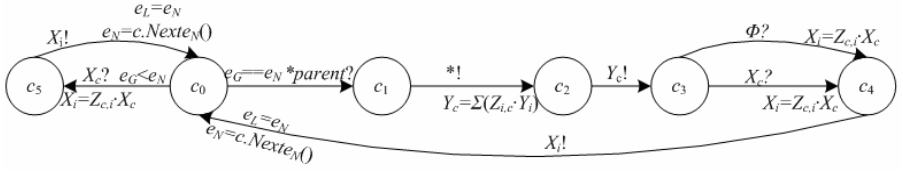

Fig. 3. Illustration of the translation from coordinator of coupled DEVS model to its according TA model. Where "*c*" is the coordinator discussed here; "!" refers to output event, "?" refers to received event, "**parent*" refers to an internal event for driven output sent by the parent of the coordinator; "*" is the internal event for driving output the coordinator send to its child models; "X_c", "Y_c" refer to input and output event the coordinator interact with outside environment; "X_i" , "Y_i" refers to the input and output events the coordinator send to and receive from its child models, "X_i"; "e_L" is the time of last event; "e_N" is the time of next output event scheduled, function $Next_N()$ refers the function to update the value of e_N when transition has just occurred. In many kinds of TAs (such as TA defined in UPPAAL[7]), global variables are allowed in TA, so X_i, Y_i, X_c, and Y_c can all be defined as integer value or integer vectors, which will improve the efficiency. "e_G" is the global virtual time controlled by the root coordinator.

3.4 Semantic of the Whole DEVS Model

A virtual simulation system is a closed system; the execution of the whole simulation model is controlled by a time management unit which is due to schedule the time advancement of the whole system. In DEVS based simulation, the time advancement is managed by a special coordinator named root coordinator, which has no interaction with outside environment and only be responsible for advancing global virtual time e_G and translating the output of one of its child model to the input of another child model. It is a reduced coordinator and its according TA is showed in Fig.4.

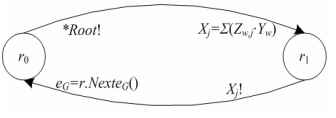

Fig. 4. Illustration of the translation from root coordinator of the whole DEVS simulation model to its according TA model. Where "*r*" refers to the root coordinator discussed here; "**Root*" refers to an internal event for driven output of its child models; "e_G" is the global virtual time; function $Nexte_G()$ is the function to update the value of e_G when a simulation step has just passed. Internal event between it and its child models noted "X_j" and "Y_w" are also be defined as integer value or integer vectors like Fig.3.

4 Conclusion

Trace equivalence is laid as the basis of the translation from Parallel model to Timed Automata model. This is an observable equivalence because trace can be observed from outside. This kind of equivalence can be used in DEVS component based simulation where the things we concerned most are not the internal detail of the components but the behaviors of them. Based on this equivalent translation, formal DEVS model verification can be realized easily using existing model checking methods and tools upon timed automata.

References

1. Darema, F.: Dynamic Data Driven Application systems: A New Paradigm for Application Simulations and Measurements. Lecture Notes in Computer Science, Vol. 3038. Springer-Verlag, Heidelberg (2004) 662–669
2. Zeigler, B.P., Praehofer, H., Kim, T.G.: Theory of Modeling and Simulation, 2nd edn. Academic Press, New York (2000)
3. Hu, X., Zeigler, B.P., Mittal, S.: Variable Structure in DEVS Component-Based Modeling and Simulation. SIMULATION: Transaction of the Society for Modeling and Simulation International, Vol.81. (2005)91–102
4. Alur, R., Dill, D.L.: A theory of timed automata. Theoretical Computer Science, Vol. 126. (1994) 183–235
5. Labiche, D.: Towards the Verification and Validation of DEVS Models. In: Proceedings of the Open International Conference on Modeling & Simulation. (2005)295–305
6. Dacharry, H.P., Giambiasi, N.: Formal Verification with timed automata and DEVS models. In: Proceedings of sixth Argentine Symposium on Software Engineering. (2005)251–265
7. Larsen, K.G., Pettersson, P.: Uppaal in a Nutshell. Int. Journal on Software Tools for Technology Transfer, Vol. 1. (1997)134–152