Modelling and Verification of the Real-Time Publish and Subscribe Protocol using UPPAAL and Simulink/Stateflow (Appendix)

Qianqian Lin[1], Shuling Wang[1,*], Bohua Zhan[1,*], and Bin Gu[2,*]

[1] State Key Laboratory of Computer Science, Institute of Software,
Chinese Academy of Sciences, Beijing, China
[2] Beijing Institute of Control Engineering, Beijing, China

Appendix of JCST paper: Lin QQ, Wang SL, Zhan BH, Gu B. Modelling and verification of the Real-Time Publish and Subscribe Protocol using UPPAAL and Simulink/Stateflow. JOURNAL OF COMPUTER SCIENCE AND TECHNOLOGY 35(6), November 2020. DOI 10.1007/s11390-020-0537-8

In the appendix, we first present the more complete semantics of timed automata (TA), then the proofs of the soundness of the translation from Stateflow to TA and Proposition 1 respectively.

1 Semantics of TA

Formally, a timed automaton is a tuple (L, l_0, C, A, E, I), where L is a set of locations, $l_0 \in L$ the initial location, C the set of clocks, A the set of actions, co-actions and the internal τ-action, $E \subseteq L \times A \times B(C) \times 2^C \times L$ the set of edges between locations with an action, a guard and a set of clocks to be reset, and $I : L \to B(C)$ assigns invariants to locations. The semantics can be defined as a labelled transition system $\langle S, s_0, \to \rangle$, where $S \subseteq L \times \mathbb{R}^C$ is the set of states, $s_0 = (l_0, u_0)$ is the initial state, and $\to \subseteq S \times (\mathbb{R}_{\geq 0} \cup A) \times S$ is the transition relation such that

- $(l, u) \xrightarrow{d} (l, u + d)$ if $\forall d' : 0 \leq d' \leq d \Rightarrow u + d' \in I(l)$,
- $(l, u) \xrightarrow{a} (l', u')$ if there exists $e = (l, a, g, r, l') \in E \ s.t. \ u \in g, u' = [r \mapsto 0]u$ and $u' \in I(l')$,

where for $d \in \mathbb{R}_{\geq 0}$, $u + d$ maps each clock x in C to the value $u(x) + d$, and $[r \mapsto 0]u$ denotes the clock valuation which maps each clock in r to 0 and agrees with u over $C \setminus r$.

The TAs can be composed into a network of timed automata over a common set of clocks and actions, consisting of n TAs $A_i = (L_i, l_0^i, C, A, E_i, I_i), 1 \leq i \leq n$. Let $\bar{l}_0 = (l_1^0, \ldots, l_n^0)$ be the initial location vector. The semantics is defined as a transition system $\langle S', s_0, \to \rangle$, where $S' = (L_1 \times \cdots \times L_n) \times \mathbb{R}^C$ is the set of states, $s_0' = (\bar{l}_0, u_0)$ is the initial state, and $\to \subseteq S' \times S'$ is the transition relation defined by:

- $(\bar{l}, u) \xrightarrow{d} (\bar{l}, u + d)$, if $\forall d'. \, 0 \leq d' \leq \Rightarrow u + d' \in I(\bar{l})$,
- $(\bar{l}, u) \xrightarrow{a} (\bar{l}[l_i'/l_i], u')$, if there exists $l_i \xrightarrow{\tau g r} l_i' \ s.t. \ u \in g, u' = [r \mapsto 0]u$ and $u' \in I(\bar{l}[l_i'/l_i])$,

- $(\bar{l}, u) \xrightarrow{a} (\bar{l}[l'_i/l_i, l'_j/l_j], u')$ if there exist $l_i \xrightarrow{c?g_i r_i} l'_i$ and $l_i \xrightarrow{c!g_j r_j} l'_i$, s.t. $u \in (g_i \wedge g_j), u' = [r_i \cup r_j \mapsto 0]u$ and $u' \in I(\bar{l}[l'_i/l_i, l'_j/l_j])$.

2 Soundness proof

Theorem 1 (Soundness of translation) *Let $\mathcal{D}$ be a Stateflow chart and $\mathcal{U}$ be the TA obtained from $\mathcal{D}$ according to our translation. Assume $(Q_i, \rightarrow_i, Q_i^0, V_i)$ for $i = 1, 2$ are the transition systems of $\mathcal{D}$ and $\mathcal{U}$, with the same initial valuation for variables in $V_1 \cap V_2$. Then there exists a simulation relation $\mathcal{B} \subseteq Q_1 \times Q_2$ such that, for any $q_1^0 \in Q_1^0$, there exists $q_2^0 \in Q_2^0$ satisfying $(q_1^0, q_2^0) \in \mathcal{B}$.*

Proof According to the translation given in Section 5.1, for each state vector $(S_1, \cdots, S_n)$ in $\mathcal{D}$, there is a location vector $(U_1, \cdots, U_n)$ defined in $\mathcal{U}$ (each denoted by U_{S_i} in the following). We use q_i to represent state vectors in the following. We construct the simulation relation $\mathcal{B}$ between $\mathcal{D}$ and $\mathcal{U}$ as follows. First of all, add $\{q_1^0, U_{q_1^0}\}$ where $q_1^0 \in Q_1^0$, in fact $q_1^0 = (S_1, \cdots, S_2, v_0)$ for some initial v_0. Obviously, $U_{q_1^0} \in Q_2^0$.

Consider the initialization transition of $\mathcal{D}$, i.e. $([S_1, \cdots, S_n], v_0) \rightarrow ([S_1, \cdots, S_n], v'_0)$ where $v'_0 = \mathit{exec}(en_{S_{1de}}; \cdots; en_{S_{nde}}, v_0)$. For ease of presentation, below we will use en_{S_1} and ex_{S_1} to represent $en_{S_{1de}}$ and $ex_{S_{1de}}$ resp., i.e. the entry and exit actions are always referring to the ones of a state and its decomposition. According to our translation, in the translated TA $\mathcal{U}$, for each U_{S_i}, there is a transition

$$U_{S_i} \xrightarrow{\neg \mathit{entry}(S_i), en_{S_i}; \mathit{entry}(S_i)=1} U_{S_i}$$

enabled, as in the beginning, $\mathit{entry}(S_i)$ is 0. By taking the one-step transitions of all corresponding U_{S_i} for all i, we get the resulting location vector and the valuation. The resulting state and location vector are still the same as initial ones, and the valuations also satisfy the requirement, since parallel states do not write the same variables. Thus the resulting configurations are still in $\mathcal{B}$.

Now consider the more complex transition that corresponds to the execution of states within one sample time step. For some state S among the parallel states, if $(S, v) \rightarrow^* (p, v')$, there are four cases (the following assume a general setting that the Stateflow states have non-disjoint transitions and have decomposition inside. The simpler case can be proved easily).

- If p is a state by taking transition (3), there exists $h > 0$ such that $\mathit{trans}(S, h) = (e, c, ca, ta)$ with target p, $v \in c$, $\neg(v \in (\mathit{cond}(S, 1) \wedge \mathit{eventP}(S, 1)) \vee \cdots \vee (\mathit{cond}(S, h-1) \wedge \mathit{eventP}(S, h-1)))$ (the transitions with higher priority than h are disabled), gv_e is 1, and $v' = \mathit{exec}(ca; ex_A; \mathit{ACT}; ta; en_p, v)[\mathit{ACT} \mapsto \mathit{skip}, \mathit{clock}(A) \mapsto 0]$, where A is S if S is a state, otherwise $\mathit{sourceSta}(S)$ if it is a junction. In the translated TA, there is a corresponding transition

$$U_S \xrightarrow{c \wedge \mathit{entry}(S) \wedge gv_e \wedge \neg((\mathit{cond}(S,1) \wedge \mathit{eventP}(S,1)) \vee \cdots \vee \mathit{cond}(S,k-1) \wedge \mathit{eventP}(S,k-1)),}$$
$$\xrightarrow{(ca; ex_A; (ACT; ta); en_p; \mathit{entry}(p)=1; ACT=\mathit{skip})} U_p.$$

Obviously, the resulting valuations are equal over the common variables as they execute the same actions. Thus the targets are in $\mathcal{B}$.

- If p is a state without available outgoing transitions, then before reaching p, S tried all its outgoing transitions, execute the during transitions and its decomposition (if there is any). We prove that for each transition of S, there is a corresponding transition of U_S.
  If the transition (1) executes, there exists $h > 0$ such that $\mathit{trans}(S, h) = (e, c, ca, ta)$ with target t, $v \in c$, $\neg(v \in (\mathit{cond}(S, 1) \wedge \mathit{eventP}(S, 1)) \vee \cdots \vee (\mathit{cond}(S, h - 1) \wedge \mathit{eventP}(S, h - 1)))$ (the transitions with higher priority $< h$ are disabled), gv_e is 1, t is a junction, and $v' = \mathit{exec}(ca, v)[\mathit{ACT} \mapsto (v(\mathit{ACT}); tc), \mathit{source}(t) \mapsto S, \mathit{sourceSta}(t) \mapsto A]$, where A is S if S is a state, otherwise $\mathit{sourceSta}(S)$ if it is a junction. In the translated TA, there is a corresponding transition:
  $$U_S \xrightarrow{c \wedge \mathit{entry}(S) \wedge gv_e \wedge \neg((\mathit{cond}(S,1) \wedge \mathit{eventP}(S,1)) \vee \cdots \vee (\mathit{cond}(S,h-1) \wedge \mathit{eventP}(S,h-1))),}$$
  $$\xrightarrow{ca; ACT=(ACT; ta); \mathit{source}(U_t)=U_S; \mathit{sourceSta}(U_t)=\mathit{sourceSta}(U_S)} U_t.$$
  Obviously, the resulting valuations are equal over the common variables.
  If transition (2) executes, in the translated TA, there is a corresponding transition
  $$U_S \xrightarrow{gv_e, gv_e=0} U_S.$$
  The transition (3) will not happen for the case when the final state p is a state without available outgoing transitions.
  If transition (4) executes, in the translated TA, there is a corresponding transition:
  $$U_S \xrightarrow{\neg \mathit{during}(S) \wedge \neg((\mathit{cond}(S,1) \wedge \mathit{eventP}(S,1)) \vee \cdots \vee (\mathit{cond}(S,n) \wedge \mathit{eventP}(S,n))),}$$
  $$\xrightarrow{dur_S; \mathit{during}(S)=1; \mathit{clock}(S)=\mathit{clock}(S)+1} U_S.$$
  If transition (5) executes, i.e. $(S, v) \to (C, v[\mathit{during}(S) \mapsto 0])$, if $\neg(v \in (\mathit{cond}(S, 1) \wedge \mathit{eventP}(S, 1)) \vee \cdots \vee (\mathit{cond}(S, n) \wedge \mathit{eventP}(S, n)))$ and $v(\mathit{during}(S))$ is 1, and C is the outmost decomposition of S. In the translated TA, there is a corresponding transition:
  $$(U_S, O) \xrightarrow{\neg((\mathit{cond}(S,1) \wedge \mathit{eventP}(S,1)) \vee \cdots \vee (\mathit{cond}(S,n) \wedge \mathit{eventP}(S,n))) \wedge \mathit{during}(S), \mathit{during}(S)=0} (I, U_C)$$
  where
  $$U_S \xrightarrow{in!, \neg((\mathit{cond}(S,1) \wedge \mathit{eventP}(S,1)) \vee \cdots \vee (\mathit{cond}(S,n) \wedge \mathit{eventP}(S,n))) \wedge \mathit{during}(S)} I, O$$
  $$\xrightarrow{in?, \mathit{during}(S)=0} U_C$$
  hold. By hiding I and U, the transition is consistent with the Stateflow transition.
  If transition (6) executes, in the translated TA, there is a corresponding transition:
  $$U_S \xrightarrow{\neg isTer(J) \wedge \neg((\mathit{cond}(S,1) \wedge \mathit{eventP}(S,1)) \vee \cdots \vee (\mathit{cond}(S,n) \wedge \mathit{eventP}(S,n)))} \mathit{source}(U_S).$$

  If transition (7) executes, then first by induction, there are corresponding transitions $(U_{S_1}, w) \xrightarrow{gv_e} (U_{S'_1}, w_1)$ and $(U_{S_2}, w) \xrightarrow{gv_e=1} (U_{S'_2}, w_2) \xrightarrow{gv_e, gv_e=0} (U_{S'_2}, w_3)$. Then according to the semantics of TA, we have $(U_{S_1}, U_{S_2}, w) \rightarrow (U_{S'_1}, U_{S'_2}, w')$ such that $w' = w_1 \uplus w_3$. The disjoint case with event broadcasting can be proved easily.

  All the above cases produce equivalent valuations over common variables.

  For the case when p is a state without available outgoing transitions, the control will go back to the outmost source state as defined in the second transition rule for Stateflow chart. This is also achieved using the $out!$ and $out?$ between the source state and its decomposition inside in our translation.

- If p is a terminal junction, the execution process is part of the above case, and the proof is similar. If p is recursively a state vector, the case holds by structural induction.

From the above proof, starting from the states in the simulation set, for each one-step transition (including the transitions within one sample time step) of Stateflow, there is a corresponding transition of the translated TA such that the targets of the two transitions still satisfy the simulation relation. The proof of this theorem is completed.

3 Proof of Proposition

Proposition 1. *Let $\mathcal{D}$ be a Stateflow chart. Assume its transition system is $T = (Q, \rightarrow, Q^0, V)$, then if $\mathcal{D}$ contains no terminal junction, the transition relation $\rightarrow$ is total, i.e. for every configuration $C \in Q$, there is $C' \in Q$ such that $(C, C') \in \rightarrow$ is enabled.*

Proof Given a Stateflow chart $\mathcal{D}$ and its transition system $T = (Q, \rightarrow, Q^0, V)$, we need to prove that, for every configuration C of T, there exists a successor C' such that $C \rightarrow C'$.

First of all, the initial set Q^0 is not empty. Furthermore, for each configuration C, it is a tuple of the form $(S_1, S_2, \cdots, S_n, v)$, then we need to prove that for each S_i, there is a possible transition path starting from a corresponding state, to reach a next state. Notice that, according to the semantics presented in Section 5.2, the disjunction of the transitions for each state or junction in T constitutes a total set. We show the case for a state S. The rules (1)-(5) define the different cases for execution of S. If there is an enabled transition from S to another state or junction, then the transitions in (1) or (3) is taken. Otherwise, the condition of transition (4) or (5) must be true and taken, depending on whether the during action is executed or not. For the special case that the during action is not explicitly defined, the transition (4) performs a time progress on the activation clock of current state (this is not considered as deadlock, and this time progress is explicitly preserved in the corresponding TA model). The transition (2) is enabled after an event occurs. The case for junction is different as defined in transition (6). But the fact also holds for it under the condition that no terminal junction exists. From the fact that the disjunction of transitions constitutes a total set, we can conclude that the transition relation $\rightarrow$ is total directly.