APPENDIX A
FORMAL DESCRIPTION ON ATOMIC ELABORATION OF HYBRID AUTOMATON

Our formal narrations assume the following conventions.

1. Given $\vec{x}^a = (x_1^a, x_2^a, \ldots, x_n^a) \in \mathbb{R}^n$ and $\vec{x}^b = (x_1^b, x_2^b, \ldots, x_m^b) \in \mathbb{R}^m$, $(\vec{x}^a, \vec{x}^b) \stackrel{\text{def}}{=} (x_1^a, x_2^a, \ldots, x_n^a, x_1^b, x_2^b, \ldots, x_m^b) \in \mathbb{R}^{n+m}$.
2. Given an n-element tuple $X = (x_1, x_2, \ldots, x_n)$, we use $X|_i$ ($i \in \{1, 2, \ldots, n\}$) to denote X's ith element: x_i. We use $\mathsf{elements}(X) \stackrel{\text{def}}{=} \{x_1, x_2, \ldots, x_n\}$ to denote the set of all elements of X.

The formal description on how to carry out atomic elaboration of hybrid automaton is as follows.

Given a hybrid automaton $A = (\vec{x}(t), V, \mathsf{inv}, F, E, g, R, L, \mathsf{syn}, \Phi_0)$, location $v \in V$, and a simple hybrid automaton $A' = (\vec{x}'(t), V', \mathsf{inv}', F', E', g', R', L', \mathsf{syn}', \Phi_0')$ such that A and A' are independent, then we can create the "atomic elaboration of A at v with A'", i.e. a hybrid automaton $A'' = (\vec{x}''(t), V'', \mathsf{inv}'', F'', E'', g'', R'', L'', \mathsf{syn}'', \Phi_0'')$, according to the following steps.

1. $\vec{x}''(t) \stackrel{\text{def}}{=} (\vec{x}(t), \vec{x}'(t))$; denote A and A''s dimensions as respectively n and n', then $\vec{x}''(t) \in \mathbb{R}^{n+n'}$.
2. $V'' \stackrel{\text{def}}{=} (V \cup V')\backslash\{v\}$.
3. $\forall u \in V\backslash\{v\}$, $\mathsf{inv}''(u) \stackrel{\text{def}}{=} \mathsf{inv}(u) \times \mathsf{inv}'(v')$, where "$\times$" means Cartesian product, v' is an arbitary location in V'; $\forall u \in V'$, $\mathsf{inv}''(u) \stackrel{\text{def}}{=} \mathsf{inv}(v) \times \mathsf{inv}''(u)$.
4. $F'' \stackrel{\text{def}}{=} \{f_u''|f_u'' : \mathbb{R}^{n+n'} \mapsto \mathbb{R}^{n+n'}, \forall u \in V''\}$ such that at location $u \in V''$, we have
   4.1. if $u \in V\backslash\{v\}$, then
   $$f_u''\big((x_1, x_2, \ldots, x_n, x_1', x_2', \ldots, x_{n'}')\big)|_i \stackrel{\text{def}}{=} \begin{cases} f_u\big((x_1, x_2, \ldots, x_n)\big)|_i & \\ \qquad\qquad \text{(when } 1 \le i \le n\text{)}, & \\ 0 \qquad\qquad \text{(otherwise)}; & \end{cases}$$
   4.2. if $u \in V'$, then
   $$f_u''\big((x_1, x_2, \ldots, x_n, x_1', x_2', \ldots, x_{n'}')\big)|_i \stackrel{\text{def}}{=} \begin{cases} f_v\big((x_1, x_2, \ldots, x_n)\big)|_i & \\ \qquad\qquad \text{(when } 1 \le i \le n\text{)}, & \\ f_u'\big((x_1', x_2', \ldots, x_{n'}')\big)|_{i-n} \quad \text{(otherwise)}. & \end{cases}$$
5. $L'' \stackrel{\text{def}}{=} L \cup L'$.
6. E'', g'', R'', syn'' are created according to the following process.
   6.1. Initially $E'' = \varnothing$.
   6.2. For each $e = (v_1, v_2) \in E$, where $v_1, v_2 \neq v$ (hence $v_1, v_2 \in V''$), we add edge $e'' = (v_1, v_2)$ into E''. Furthermore, we define $g''(e'') \stackrel{\text{def}}{=} g(e) \times \mathbb{R}^{n'}$;
   $$r_{e''}''\big((s_1, s_2, \ldots, s_n, s_1', s_2', \ldots, s_{n'}')\big) \stackrel{\text{def}}{=} r_e\big((s_1, s_2, \ldots, s_n)\big) \times \{(s_1', s_2', \ldots, s_{n'}')\}, \quad \forall (s_1, s_2, \ldots, s_n, s_1', s_2', \ldots, s_{n'}') \in \mathsf{inv}''(v_1);$$
   and $\mathsf{syn}''(e'') \stackrel{\text{def}}{=} \mathsf{syn}(e)$.
   6.3. For each $e = (v_1, v) \in E$, where $v_1 \neq v$ (hence $v_1 \in V''$), we add for each $v' \in \Phi_0'|_{V'}$ (i.e. Φ_0''s projection on V') an edge $e'' = (v_1, v')$ into E''. Furthermore, we define $g''(e'') \stackrel{\text{def}}{=} g(e) \times \mathbb{R}^{n'}$;
   $$r_{e''}''\big((s_1, s_2, \ldots, s_n, s_1', s_2', \ldots, s_{n'}')\big) \stackrel{\text{def}}{=} r_e\big((s_1, s_2, \ldots, s_n)\big) \times \{(s_1', s_2', \ldots, s_{n'}')\}, \quad \forall (s_1, s_2, \ldots, s_n, s_1', s_2', \ldots, s_{n'}') \in \mathsf{inv}''(v_1);$$
   and $\mathsf{syn}''(e'') \stackrel{\text{def}}{=} \mathsf{syn}(e)$.
   6.4. For each $e = (v, v_2) \in E$, where $v_2 \neq v$ (hence $v_2 \in V''$), we add for each $v' \in V'$ an edge $e'' = (v', v_2)$ into E''. Furthermore, we define $g''(e'') \stackrel{\text{def}}{=} g(e) \times \mathbb{R}^{n'}$;
   $$r_{e''}''\big((s_1, s_2, \ldots, s_n, s_1', s_2', \ldots, s_{n'}')\big) \stackrel{\text{def}}{=} r_e\big((s_1, s_2, \ldots, s_n)\big) \times \{(s_1', s_2', \ldots, s_{n'}')\}, \quad \forall (s_1, s_2, \ldots, s_n, s_1', s_2', \ldots, s_{n'}') \in \mathsf{inv}''(v');$$
   and $\mathsf{syn}''(e'') \stackrel{\text{def}}{=} \mathsf{syn}(e)$.
   6.5. For each $e = (v, v) \in E$, we add for each $v' \in V'$ an edge $e'' = (v', v')$ into E''. Furthermore, we define $g''(e'') \stackrel{\text{def}}{=} g(e) \times \mathbb{R}^{n'}$;
   $$r_{e''}''\big((s_1, s_2, \ldots, s_n, s_1', s_2', \ldots, s_{n'}')\big) \stackrel{\text{def}}{=} r_e\big((s_1, s_2, \ldots, s_n)\big) \times \{(s_1', s_2', \ldots, s_{n'}')\}, \quad \forall (s_1, s_2, \ldots, s_n, s_1', s_2', \ldots, s_{n'}') \in \mathsf{inv}''(v');$$
   and $\mathsf{syn}''(e'') \stackrel{\text{def}}{=} \mathsf{syn}(e)$.
   6.6. For each $e' = (v_1, v_2) \in E'$ (hence $v_1, v_2 \in V''$), we add an edge $e'' = (v_1, v_2)$ into E''. Furthermore, we define $g''(e'') \stackrel{\text{def}}{=} \mathbb{R}^n \times g'(e')$;
   $$r_{e''}''\big((s_1, s_2, \ldots, s_n, s_1', s_2', \ldots, s_{n'}')\big) \stackrel{\text{def}}{=} \{(s_1, s_2, \ldots, s_n)\} \times r_{e'}'\big((s_1', s_2', \ldots, s_{n'}')\big); \quad \forall (s_1, s_2, \ldots, s_n, s_1', s_2', \ldots, s_{n'}') \in \mathsf{inv}''(v_1);$$
   and $\mathsf{syn}''(e'') \stackrel{\text{def}}{=} \mathsf{syn}'(e)$.
7. Φ_0'' is created according to the following process.
   7.1. Initially $\Phi_0'' = \varnothing$.
   7.2. For each $(v_1, \vec{s}) \in \Phi_0$, where $v_1 \neq v$, we add $(v_1, (\vec{s}, \underbrace{0, 0, \ldots, 0}_{n' \text{ zeros}}))$ into Φ_0''.
   7.3. For each $(v, \vec{s}) \in \Phi_0$, we add for each $(v', \vec{s}') \in \Phi_0'$ a state value $(v', (\vec{s}, \vec{s}'))$ into Φ_0''.
8. For PTE CPS, there is the issue of partitioning V'' into safe-locations V''^{safe} and risky-locations V''^{risky}. In case $v \in V^{\text{safe}}$, $V''^{\text{safe}} \stackrel{\text{def}}{=} V' \cup (V^{\text{safe}}\backslash\{v\})$ and $V''^{\text{risky}} \stackrel{\text{def}}{=} V''\backslash V''^{\text{safe}}$; otherwise, $V''^{\text{risky}} \stackrel{\text{def}}{=} V' \cup (V^{\text{risky}}\backslash\{v\})$ and $V''^{\text{safe}} \stackrel{\text{def}}{=} V''\backslash V''^{\text{risky}}$.

We denote A'', the elaboration of A at v with A', as

$$A'' = \mathcal{E}(A, v, A').$$

APPENDIX B
DETAILED DIAGRAMS FOR DESIGN PATTERN HYBRID AUTOMATA

Please see Fig. 8, 9, and 10 for the detailed diagram of A_{supvsr}, A_{initzr}, and $A_{\text{ptcpnt},i}$, the hybrid automaton for Supervisor, Initializer, and the ith Participant respectively.

APPENDIX C
DETAILED DESIGNS FOR LASER TRACHEOTOMY

The detailed designs for the wireless laser tracheotomy entities are shown in Fig. 11, 12, and 13 respectively. Some data state variable names and/or locations names in the corresponding design patterns are modified to better reflect their meanings in laser tracheotomy.

APPENDIX D
PROOF OF THEOREM 1

First, we can prove the following lemma.

Lemma 1 (Guaranteed Resetting): Suppose $evt\xi_0 To\xi_1 LeaseReq$ happens at time t_0, then we have

Claim 1.1: $\forall i \in \{0, 1, \ldots, N\} \cdot \ell_i(t_0^-) \equiv$ "Fall-Back".

Claim 1.2: Within $T^{\max}_{\text{wait}} + T^{\max}_{\text{LS1}}$, ℓ_i ($i = 0 \sim N$) will return to its "Fall-Back".

Proof of Claim 1.1: First, because $evt\xi_0 To\xi_1 LeaseReq$ happens at t_0 and ξ_0's location "L0" allows 0 dwelling time, we have $\ell_0(t_0^-) =$ "Fall-Back". ξ_0's location "L0" allows 0 dwelling time also implies $evt\xi_N To\xi_0 Req$ happens at t_0, so $\ell_N(t_0^-) =$ "Fall-Back" (see Fig. 9).

Now all we need to prove is

$$\ell_i(t_0^-) = \text{ "Fall-Back" } (i = 1 \sim N-1). \tag{2}$$

We can prove this inductively. As all hybrid automata of $\mathcal{H}$ start from respective "Fall-Back" locations, if t_0 is the first time $evt\xi_0 To\xi_1 LeaseReq$ happens, Claim (2) sustains.

Suppose Claim (2) sustains for t_0, a time instance that $evt\xi_0 To\xi_1 LeaseReq$ happens. Suppose $t_1 > t_0$ is the next time instance that $evt\xi_0 To\xi_1 LeaseReq$ happens. Then there can be two cases for the interval of $[t_0, t_1)$.

Case 1: During $[t_0, t_1)$, $evt\xi_1 LeaseExpire$ never happens.

Suppose during $[t_0, t_1)$, ℓ_0 ever reaches location "Lease ξ_j" but not location "Lease ξ_{j+1}" ($j \in \{1, 2, \ldots, N-1\}$).

Then throughout $[t_0, t_1)$, ξ_{j+1}, ξ_{j+2}, ..., ξ_{N-1} never get a chance to leave their respective "Fall-Back" locations.

By checking all possible exit paths from "Lease ξ_j" in Fig. 8, and knowing that $evt\xi_1 LeaseExpire$ never happens, we find for each entity ξ_k (where $k = j, j-1, \ldots, 1$), $\exists t \in [t_0, t_1) \cdot$ either $evt\xi_k To\xi_0 LeaseDeny$ or $evt\xi_k To\xi_0 Exit$ happens at t. For otherwise, ℓ_0 cannot be at "Fall-Back" at t_1^-, which has already been proven at the beginning of this proof. Meanwhile, according to Fig. 10, as soon as $evt\xi_k To\xi_0 LeaseDeny$ or $evt\xi_k To\xi_0 Exit$ happens, ℓ_k enters "Fall-Back", and stays there till t_1^- because there is no more $evt\xi_0 To\xi_1 LeaseReq$ (hence $evt\xi_0 To\xi_k LeaseReq$) during $[t, t_1)$.

The same analysis applies to the case when ℓ_0 ever reaches location "Lease ξ_N" during $[t_0, t_1)$.

Therefore, Claim (2) sustains for t_1 in Case 1.

Case 2: $\exists t \in [t_0, t_1)$, $evt\xi_1 LeaseExpire$ happens. Then according to Fig. 8, 9, and 10, $t_0 + T^{\max}_{\text{LS1}} \leq t$. As defined, $t < t_1$, so $t_0 + T^{\max}_{\text{LS1}} < t_1$.

On the other hand, the latest time during $[t_0, t_1)$ for ξ_i ($i \in \{1, 2, \ldots, N-1\}$) to leave "Fall-Back" (if it ever leaves) is at $t_0 + (i-1)T^{\max}_{\text{wait}}$ (see Fig. 8, 10). After that, ξ_i's maximal stay outside of "Fall-Back" is $T^{\max}_{\text{enter},i} + T^{\max}_{\text{run},i} + T_{\text{exit},i}$ (see Fig. 10). Because of Condition c4, this means by $t_0 + T^{\max}_{\text{LS1}} < t_1$, ξ_i should have returned "Fall-Back". Due to the same reason as in Case 1, after the return, ξ_i should have stayed in "Fall-Back" till t_1^-.

Therefore, Claim (2) sustains for t_1 in Case 2.

Due to Case 1 and 2, Claim (2) sustains for t_1. Induction sustains.

Note no matter there are infinite, or finite occurrences of $evt\xi_0 To\xi_1 LeaseReq$, the above induction based proof sustains. ■

Proof of Claim 1.2: Because of Condition c2, A_{supvsr} is time-block-free. By checking every possible path of A_{supvsr}, after t_0, within $T^{\max}_{\text{wait}} + T^{\max}_{\text{LS1}}$, ℓ_0 will return to "Fall-Back". Suppose it is at $t_1 \in [t_0, t_0 + T^{\max}_{\text{wait}} + T^{\max}_{\text{LS1}}]$ that ξ_0 returns to "Fall-Back" for the first time. We also have two cases.

Case 1: During $[t_0, t_1]$, $evt\xi_1 LeaseExpire$ never happens.

Suppose during $[t_0, t_1]$, ℓ_0 ever reaches location "Lease ξ_j" but not location "Lease ξ_{j+1}" ($j \in \{1, 2, \ldots, N-1\}$).

Then throughout $[t_0, t_1]$, ξ_{j+1}, ξ_{j+2}, ..., ξ_N never get a chance to leave their respective "Fall-Back" locations.

By checking all possible exit paths from "Lease ξ_j" in Fig. 8, and knowing that $evt\xi_1 LeaseExpire$ never happens, we find for each entity ξ_k (where $k = j, j-1, \ldots, 1$), $\exists t \in [t_0, t_1] \cdot$ either $evt\xi_k To\xi_0 LeaseDeny$ or $evt\xi_k To\xi_0 Exit$ happens at t. For otherwise, ℓ_0 cannot be at "Fall-Back" at t_1. Meanwhile, according to Fig. 10, as soon as $evt\xi_k To\xi_0 LeaseDeny$ or $evt\xi_k To\xi_0 Exit$ happens, ℓ_k enters "Fall-Back", and stays there till t_1 because there is no more $evt\xi_0 To\xi_1 LeaseReq$ (hence $evt\xi_0 To\xi_k LeaseReq$) during $[t, t_1]$.

The same analysis applies to the case when ℓ_0 ever reaches location "Lease ξ_N" during $[t_0, t_1]$.

Case 2: $\exists t \in [t_0, t_1]$, $evt\xi_1 LeaseExpire$ happens. Then according to Fig. 8, 9, and 10, $t_0 + T^{\max}_{\text{LS1}} \leq t$. As defined, $t \leq t_1$, so $t_0 + T^{\max}_{\text{LS1}} \leq t_1$.

On the other hand, the latest time during $[t_0, t_1]$ for ξ_i ($i \in \{1, 2, \ldots, N\}$) to leave "Fall-Back" (if it ever leaves) is at

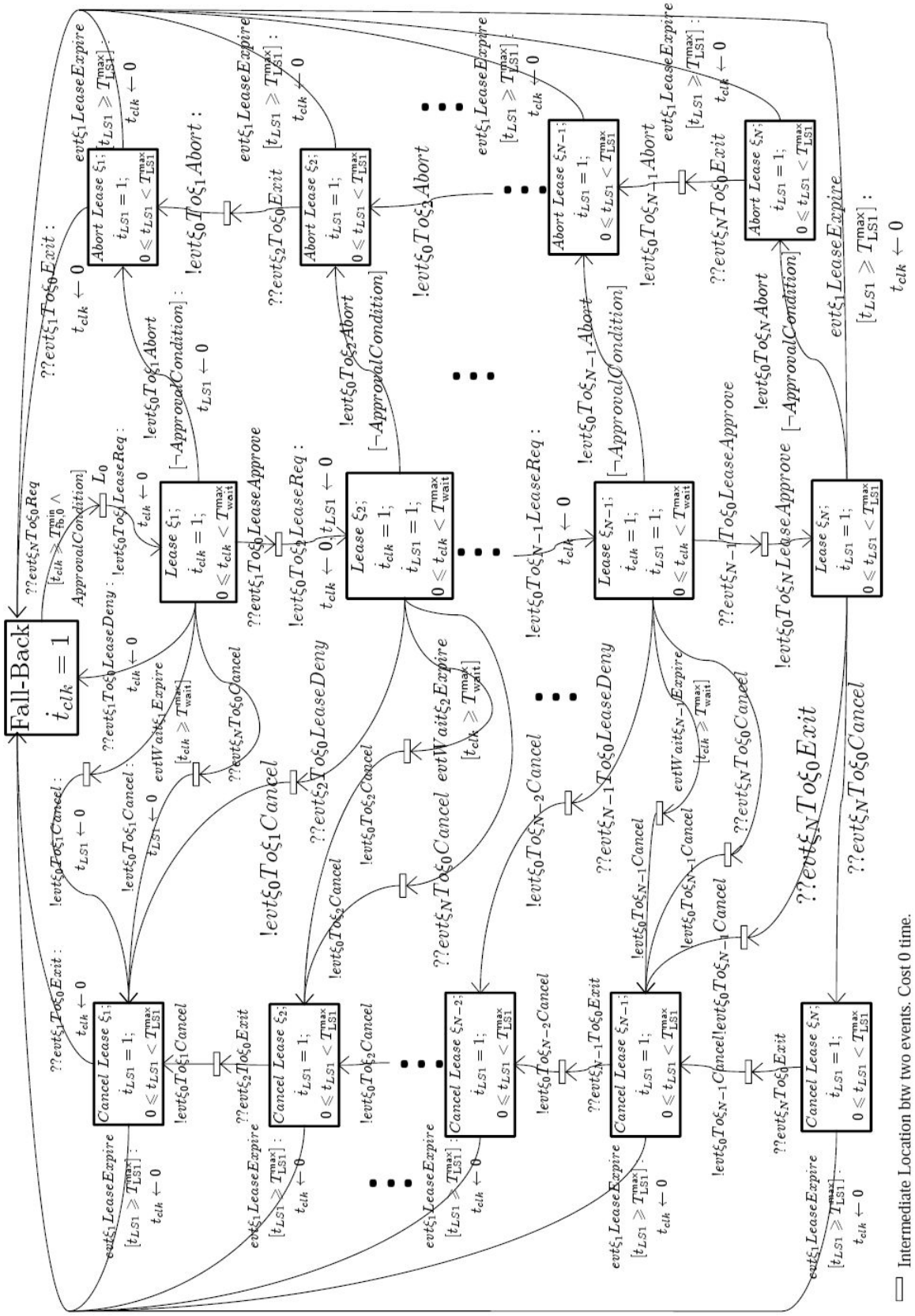

Fig. 8. Diagram of Hybrid Automaton A_{supvsr}, the Design Pattern for Supervisor. Each rectangle box indicates a location. Inside the box, the first line is the location's name (note state variable names and location names are local to their respective hybrid automata; hence two distinct locations of two distinct hybrid automata may have the same name). Annotations to each edge (aka transition) comply with the following conventions. Before the ':' are the synchronization label and the guard formula (quoted by brackets "[]") for the edge. After the ':' are the data state value resets ("$x \leftarrow a$" means "assigning x of value a"). The above notational conventions also apply to Fig. 9 and 10.

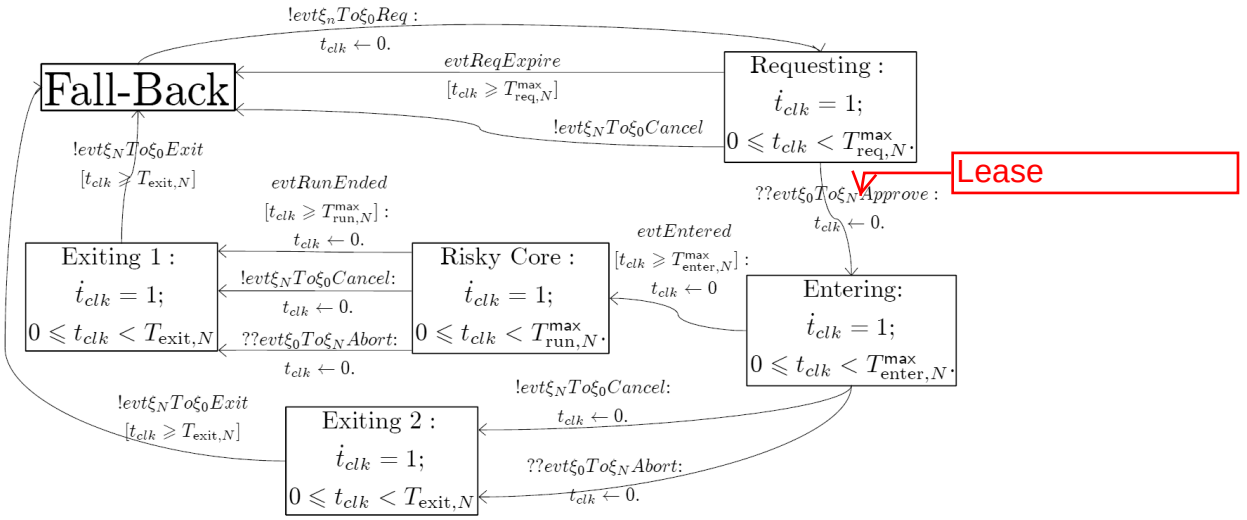

Fig. 9. Diagram of Hybrid Automaton A_{initzr}, the Design Pattern for Initializer

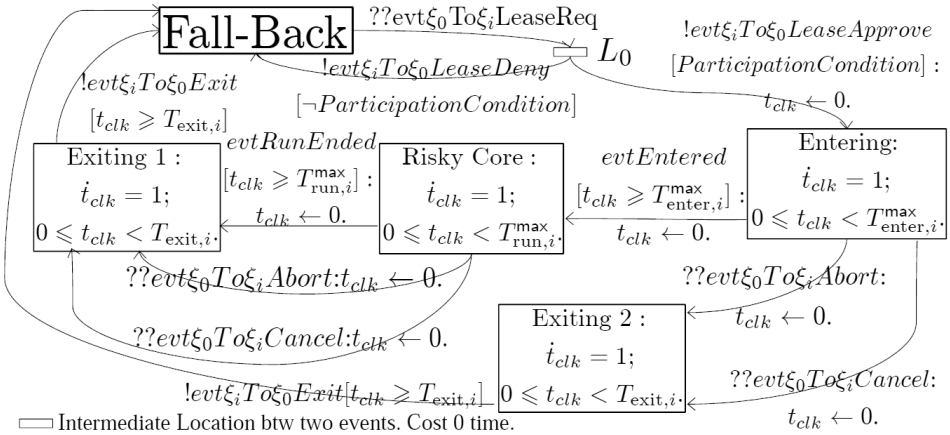

Fig. 10. Diagram of Hybrid Automaton $A_{\text{ptcpnt},i}$, the Design Pattern for the ith Participant.

$t_0+(i-1)T^{\max}_{\text{wait}}$ (see Fig. 8, 9, 10). After that, ξ_i's maximal stay outside of "Fall-Back" is $T^{\max}_{\text{enter},i}+T^{\max}_{\text{run},i}+T_{\text{exit},i}$ (see Fig. 9, 10). Because of Condition c4, this means by $t_0+T^{\max}_{\text{LS1}} \leq t_1$, ξ_i should have returned "Fall-Back". Due to the same reason as in Case 1, after the return, ξ_i should have stayed in "Fall-Back" till t_1.

Due to Case 1 and 2, ξ_i ($i = 1 \sim N$) should have returned to its "Fall-Back" location at $t_1 \in [t_0, t_0 + T^{\max}_{\text{wait}} + T^{\max}_{\text{LS1}}]$. ■

Then we have the following lemma.

Lemma 2 (Single Visit between Resets): Let t_0, t_1 ($t_0 < t_1$) be the time instances that two consecutive $evt\xi_0To\xi_1LeaseReq$ happen; or let t_0 be the last time that $evt\xi_0To\xi_1LeaseReq$ happens and $t_1 = \infty$. Either way,

$$\forall t \in (t_0, t_1) \cdot \text{no } evt\xi_0To\xi_1LeaseReq \text{ happens at } t. \quad (3)$$

We then have

Claim 2.1: Given $i \in \{1, 2, \ldots, N-1\}$, then throughout interval $[t_0, t_1)$, ξ_i can respectively enter its location "L0", "Entering", "Risky Core", "Exiting 1", and "Exiting 2" (see Fig. 10) for at the most once.

Claim 2.2: Throughout interval $[t_0, t_1)$, ξ_N can respectively enter its location "Entering", "Risky Core", "Exiting 1", and "Exiting 2" (see Fig. 9) for at the most once.

Proof: Due to Lemma 1, we have $\forall i \in \{0, 1, \ldots, N\} \cdot \ell_i(t_0^-) =$ "Fall-Back".

Given $i \in \{1, 2, \ldots, N-1\}$, for ℓ_i to enter "L0" twice in interval $[t_0, t_1)$, $evt\xi_0To\xi_iLeaseReq$ must happen twice (see Fig. 10), which implies $evt\xi_0To\xi_1LeaseReq$ happen twice (see Fig. 8), which implies $\exists t_2 \in (t_0, t_1) \cdot evt\xi_0To\xi_1LeaseReq$ happens at t_2. This contradicts Formula (3).

Same reasoning also applies to location "Entering", "Risky Core", "Exiting 1", and "Exiting 2". Hence Claim 2.1 is proven.

Same way we can prove Claim 2.2, where $evt\xi_0To\xi_NLeaseApprove$ replaces $evt\xi_0To\xi_iLeaseReq$. ■

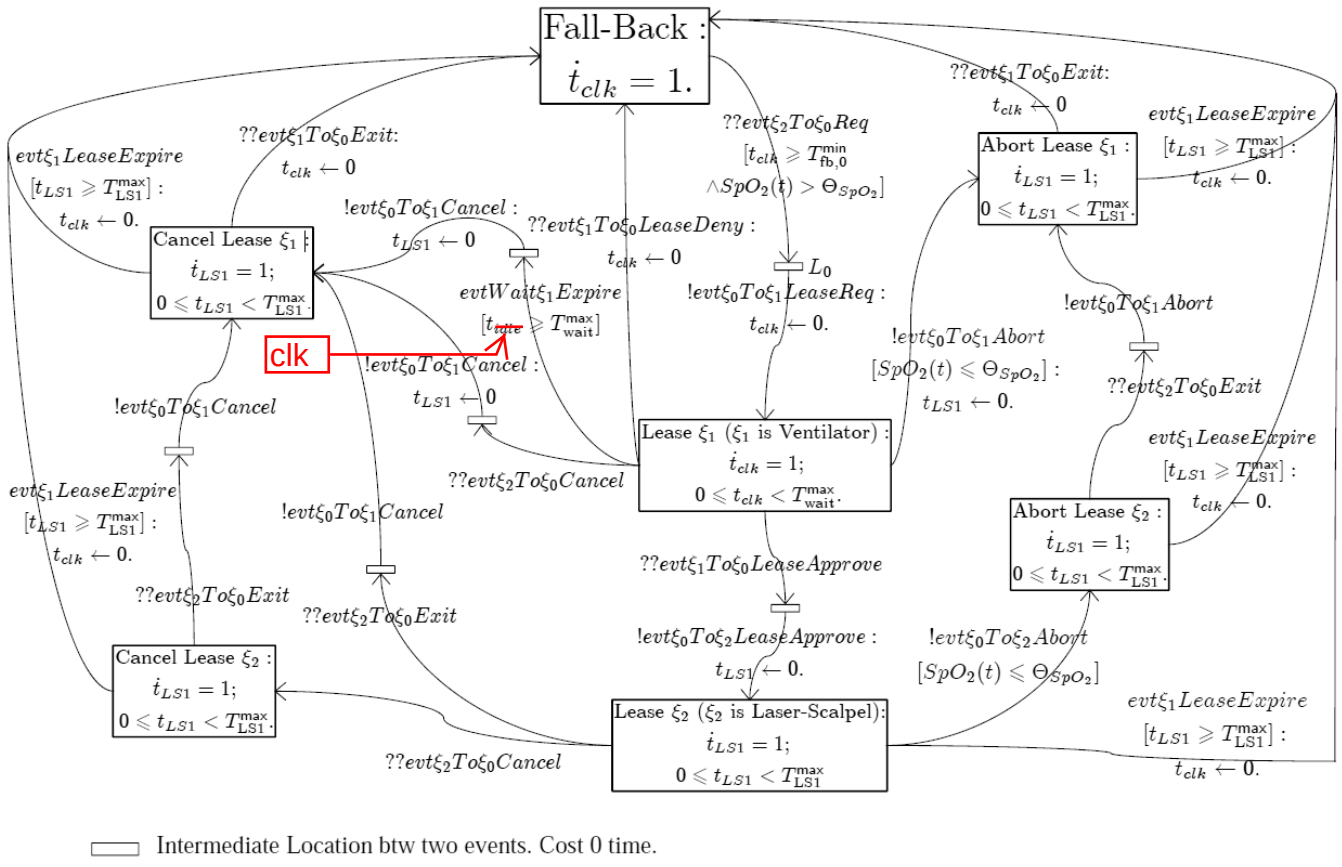

Fig. 11. Laser Tracheotomy Supervisor Detailed Design. Note entity ξ_1 refers to the ventilator, and ξ_2 refers to the laser-scalpel.

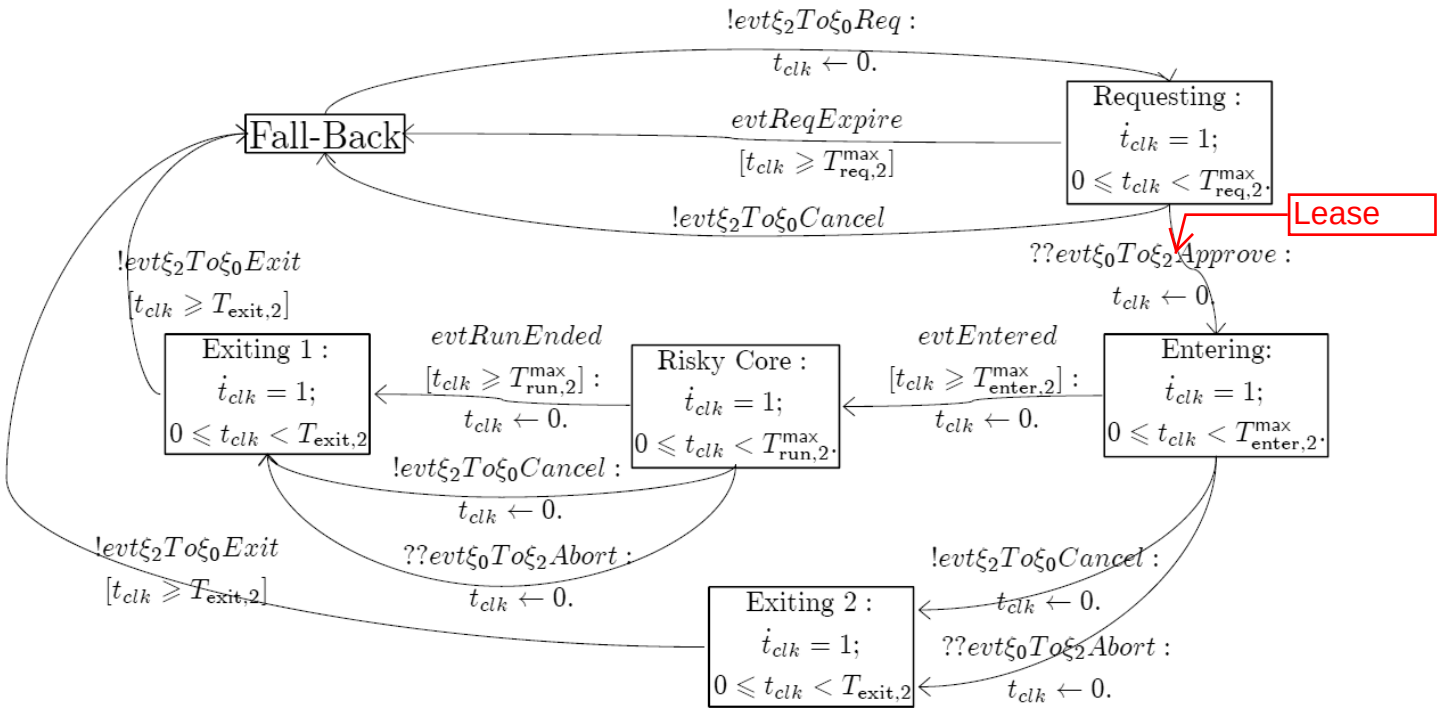

Fig. 12. Laser Tracheotomy Laser-Scalpel Detailed Design. Note the laser-scalpel emits and only emits laser when dwelling in location "Risky Core".

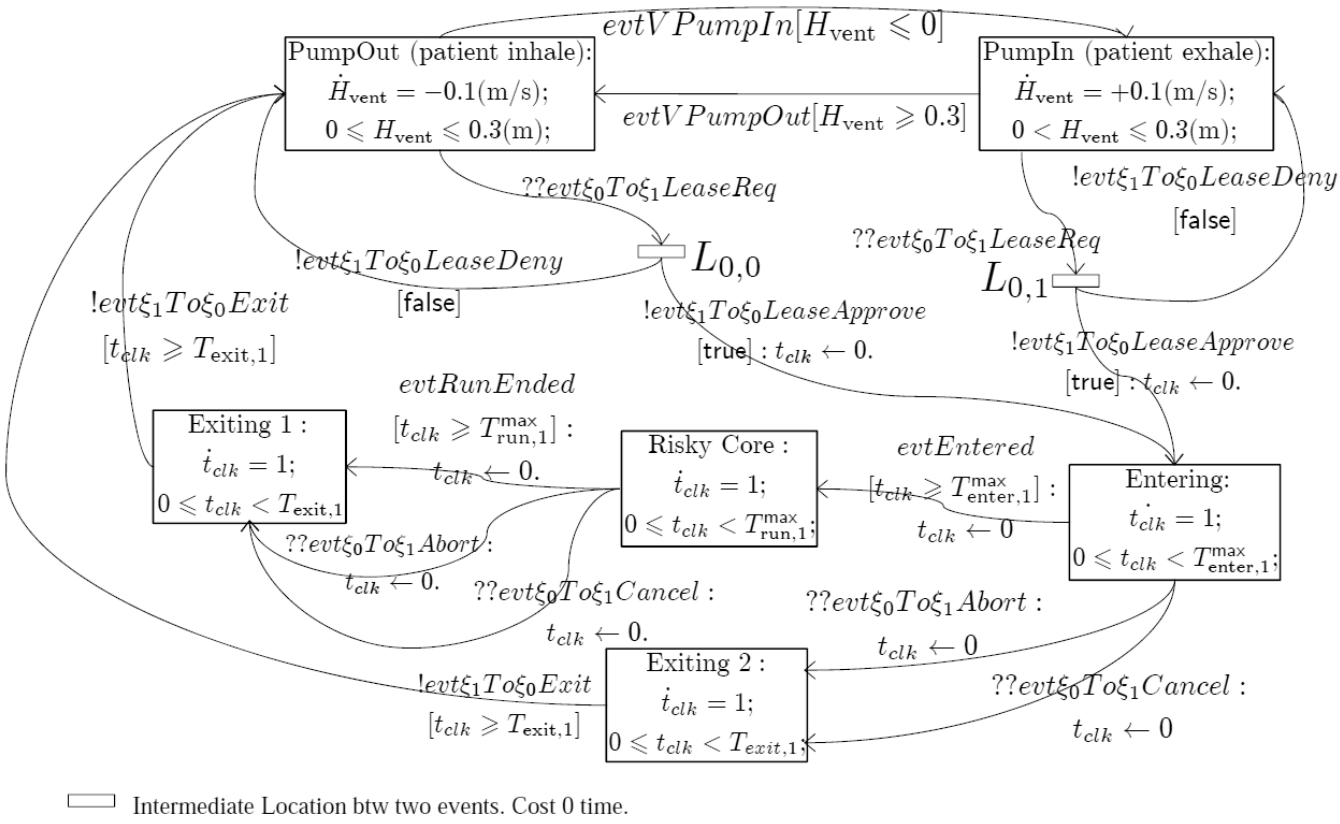

Fig. 13. Laser Tracheotomy Ventilator Detailed Design, by elaborating Participant Design Pattern. Note outside of location "PumpOut" and "PumpIn", H_{vent} remains unchanging; hence in "Entering", "Exiting 1", "Exiting 2", and "Risky Core", the ventilator is always pausing.

Furthermore, we have the following lemma.

Lemma 3 (PTE Compliance): Let t_0, t_1 ($t_0 < t_1$) be the time instances when two consecutive $evt\xi_0To\xi_1LeaseReq$ happen; or let t_0 be the last time $evt\xi_0To\xi_1LeaseReq$ happens and $t_1 = \infty$. In both cases, $\forall t \in [t_0, t_1)$, Ineq. (1) sustains.

Proof: Without loss of generality, let us first focus on entity ξ_i and ξ_{i-1} (where $i \in \{2, 3, \ldots, N-1\}$).

If ξ_i never entered risky-locations throughout interval $[t_0, t_1)$, then the PTE ordering of $\xi_{i-1} < \xi_i$ trivially sustains.

Otherwise, there must be a maximal interval $[t_2, t_3) \subseteq [t_0, t_1)$ that ℓ_i stays in risky-locations. That is, $evtEntered$ happens at t_2, $evt\xi_iTo\xi_0Exit$ happens at $t_3 < t_1$ (no matter $t_1 < \infty$ or $t_1 = \infty$, we know $t_3 < t_1$ due to Lemma 1), and $\forall t \in [t_2, t_3) \cdot \ell_i(t) \in \{$"Risky Core", "Exiting 1"$\}$.

Due to Lemma 2, $[t_2, t_3)$ is the only interval in $[t_0, t_1)$ that ℓ_i stays in risky-locations. Due to Lemma 1, $\ell_i(t_0^-) =$ "Fall-Back". By exhaustively examing all possible paths in $A_{\text{ptcpnt},i}$ of Fig. 10, we have

$$t_0 + T^{\max}_{\text{enter},i} \leq \quad t_2 \quad \leq t_0 + (i-1)T^{\max}_{\text{wait}} + T^{\max}_{\text{enter},i}; \text{ (4)}$$
$$\text{and } t_2 + T_{\text{exit},i} \leq \quad t_3 \quad \leq t_2 + T^{\max}_{\text{run},i} + T_{\text{exit},i}. \text{ (5)}$$

Now let us check the duration that ξ_{i-1} may stay in its risky-locations within $[t_0, t_1)$.

By exhaustively examining all possible paths in $A_{\text{ptcpnt},i}$ of Fig. 10, for ξ_i's $evtEntered$ to happen at t_2, $evt\xi_iTo\xi_0LeaseApprove$ must happen at

$$t_4 = t_2 - T^{\max}_{\text{enter},i}. \quad (6)$$

Note $t_4 \geq t_0$ because of Ineq. (4). According to A_{supvsr} of Fig. 8, because $\ell_0(t_0^-) =$ "Fall-Back", for $evt\xi_iTo\xi_0LeaseApprove$ to happen at $t_4 \in [t_0, t_2)$, $evt\xi_{i-1}To\xi_0LeaseApprove$ must have happened at some time instance t_5, where

$$t_0 \leq t_5 \leq t_4. \quad (7)$$

Due to Condition c5,

$$\begin{aligned} & t_5 + T^{\max}_{\text{enter},i-1} + T^{\min}_{\text{risky:}i-1\to i} \\ < \quad & t_4 + T^{\max}_{\text{enter},i} = t_2 \quad \text{(due to Eq. (6))}. \end{aligned} \quad (8)$$

On the otherhand, because after t_0, the first occurrence of $evt\xi_iTo\xi_0Exit$ happens at t_3, by exhaustively checking all possible paths in A_{supvsr} of Fig. 8, this implies the following proposition.

Proposition 1: No $evt\xi_0To\xi_jAbort$ nor $evt\xi_0To\xi_jCancel$ ever happened during $[t_0, t_3)$ ($\forall j = i-1, i-2, \ldots, 1$). □

Because of Proposition 1, Ineq. (7)(8) imply that after ℓ_{i-1} enters location "Entering" at t_5, it enters "Risky Core", i.e. risky-locations, at

$$t_6 = t_5 + T^{\max}_{\text{enter},i-1}. \quad (9)$$

Due to Ineq. (8), we have

$$t_6 < t_2 - T^{sfmin}_{\text{risky:}i-1\to i} < t_3. \tag{10}$$

On the other hand, because ξ_i's location "L0" has 0 dwelling time, and $evt\xi_i To\xi_0 LeaseApprove$ happens at t_4, so $evt\xi_0 To\xi_i LeaseReq$ must happen at t_4. We hence have

$$t_5 \geq t_4 - T^{\mathsf{max}}_{\text{wait}}. \tag{11}$$

For otherwise, ξ_0 would have given up receiving $evt\xi_{i-1} To\xi_0 LeaseApprove$ at $t_5 + T^{\mathsf{max}}_{\text{wait}}$ and would not have sent $evt\xi_0 To\xi_i LeaseReq$ at t_4.

Ineq. (11) and Condition c6 together imply

$$\begin{aligned} & t_5 + T^{\mathsf{max}}_{\text{enter},i-1} + T^{\mathsf{max}}_{\text{run},i-1} \\ > \;& t_4 + T^{\mathsf{max}}_{\text{enter},i} + T^{\mathsf{max}}_{\text{run},i} + T_{\text{exit},i} \\ = \;& t_2 + T^{\mathsf{max}}_{\text{run},i} + T_{\text{exit},i} \quad \text{(due to Eq. (6))} \\ \geq \;& t_3. \quad \text{(due to Ineq. (5))} \end{aligned} \tag{12}$$

Due to Proposition 1, during $[t_5, t_3)$, ξ_{i-1} never receives $evt\xi_0 To\xi_{i-1} Abort$ or $evt\xi_0 To\xi_{i-1} Cancel$. This fact, combined with Ineq. (12), implies the earliest time instance after t_5 that ξ_{i-1} may receive $evt\xi_0 To\xi_{i-1} Abort$ or $evt\xi_0 To\xi_{i-1} Cancel$ is t_3. Let $t_7 \in [t_0, t_1)$ be the time instance that ℓ_{i-1} exits risky-locations, then

$$\begin{aligned} t_7 \;\geq\; & \mathsf{min}\{t_5 + T^{\mathsf{max}}_{\text{enter},i-1} + T^{\mathsf{max}}_{\text{run},i-1} + T_{\text{exit},i-1}, \\ & \quad t_3 + T_{\text{exit},i-1}\} \\ = \;& t_3 + T_{\text{exit},i-1} \quad \text{(due to Ineq. (12))} \\ > \;& t_3 + T^{\mathsf{min}}_{\text{safe:}i\to i-1} \quad \text{(due to Condition c7)} \end{aligned} \tag{13}$$

Note Ineq. (10)(13) implies $t_6 < t_7$; and no matter $t_1 < \infty$ or $t_1 = \infty$, Lemma 1 implies $t_7 < t_1$. Therefore, to summarize, ξ_{i-1} must have visited risky-locations for once during interval $[t_0, t_1)$; and the vist starts at t_6, and ends at t_7, where t_6 compiles with Ineq. (10), and t_7 complies with Ineq. (13). In other words, the PTE ordering of $\xi_{i-1} < \xi$ sustains during interval $[t_0, t_1)$.

Using the same approach, we can also prove $\xi_{N-1} < \xi_N$ during $[t_0, t_1)$. ■

With the above lemmas, we can prove Theorem 1 as follows.

Throughout the execution of hybrid system $\mathcal{H}$, if $evt\xi_0 To\xi_1 LeaseReq$ never happens, as all entities are initialized from "Fall-Back" locations, the PTE safety rules trivially sustain.

If infinite number of $evt\xi_0 To\xi_1 LeaseReq$ happen. Then before the first $evt\xi_0 To\xi_1 LeaseReq$, all entities stay in "Fall-Back" locations, PTE safety rules trivially sustain. After that, between any two consecutive $evt\xi_0 To\xi_1 LeaseReq$, due to Lemma 1, PTE Safety Rule 1 sustains, and a (loose) risky-locations continuous dwelling time bound is $T^{\mathsf{max}}_{\text{wait}} + T^{\mathsf{max}}_{\text{LS1}}$; due to Lemma 3, PTE Safety Rule 2 also sustains. Therefore, PTE safety rules sustain.

The same proving approach can be applied to the scenario where finite number of $evt\xi_0 To\xi_1 LeaseReq$ happen (we need to check the special case: the interval between the last occurrence of $evt\xi_0 To\xi_1 LeaseReq$ and time ∞; but the same proving approach can be applied, and the conclusion is the same). ■