The EFSM is the tuple S = (Q, Σ1, Σ2, q0, V, Λ),

where

Q = {dormant, init, idle, monitoring, safe_shutdown, error_diagnosis, final}

Σ1 = {kill, start, init_ok, begin_monitoring, moni_crash, init_crash, idle_crash, retry_init, idle_rescue, moni_rescue, shutdown, sleep}

Σ2 = {retry++, moni_err_msg, idle_err_msg, init_err_msg, retry=0}

q0 : dormant

V : retry = {0, 1,2,3}

$\Lambda_{unrefined}$ ={

1. → dormant

2. dormant$\xrightarrow{kill}$ final

3. dormant$\xrightarrow{start}$ init

4. init$\xrightarrow{init_ok}$ idle

5. init$\xrightarrow{init_{crash}/\ init_err_msg}$ error_diagnosis

6. init$\xrightarrow{kill}$ final

7. idle$\xrightarrow{begin_monitoring}$ monitoring

8. idle$\xrightarrow{idle_crash/\ idle_err_msg}$ error_diagnosis

9. idle$\xrightarrow{kill}$ final

10. monitoring$\xrightarrow{kill}$ final

11. monitoring$\xrightarrow{moni_crash/\ moni_err_msg}$ error_diagnosis

12. error_diagnosis$\xrightarrow{kill}$ final

13. error_diagnosis$\xrightarrow{moni_rescue}$ monitoring

14. error_diagnosis$\xrightarrow{retry_init[retry \leq 3]/retry{+}{+}}$ init

15. error_diagnosis$\xrightarrow{idle_rescue}$ idle

16. error_diagnosis$\xrightarrow{shutdown[retry>3]/retry=0}$ safe_shutdown

17. safe_shutdown $\xrightarrow{kill}$ final

18. safe_shutdown$\xrightarrow{sleep}$ $dormant$

}

The EFSM of the init state is the tuple S = (Q, Σ1, Σ2, q0, V, Λ),

where

Q = {boot_hw, senchk, tchk, psichk, ready }

Σ1 = {hw_ok, sen_ok, t_ok, psi_ok}

Σ2 = {}

q0 : boot_hw

V = {}

Λ_{refined} ={

1. → boot_hw
2. boot_hw$\xrightarrow{hw_ok}$ senchk
3. senchk$\xrightarrow{sen_ok}$ tchk
4. tchk$\xrightarrow{t_ok}$ psichk
5. psichk$\xrightarrow{psi_ok}$ ready

}

The EFSM of the refined monitoring state is the tuple S = (Q, Σ1, Σ2, q0, V, Λ), where

Q = {monidle, regulate_environment, lockdown}

Σ1 = {verify_contagion, contagion_alert,_no_contagion, after_100ms, purge_succ}

Σ2 = {inlockdown=false, inlockdown=true, set contagion, FACILITY_CRIT_MESG}

q0 : monidle

V = {inlockdown{true, false}}

Λ_{refined} ={

1. → monidle
2. monidle$\xrightarrow{no_contagion}$ regulate_environment

3. monidle$\xrightarrow{contagion_alert/FACILITY_CRIT_MSG,inlockdown=true}$ lockdown
4. monidle$\xrightarrow{verify_contagion/set\ contagion}$ monidle
5. regulate_environment$\xrightarrow{after_100ms}$ monidle
6. lockdown$\xrightarrow{purge_succ/inlockdown=false}$ monidle

}