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}

$\Lambda_{\text{unrefined}}$ ={

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 = {}

$\Lambda_{refined}$ ={

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}

q0 : monidle

V = {inlockdown{true, false}}

$\Lambda_{refined}$ ={

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

}

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

where

Q = {prep_vpurge, alt_temp, alt_psi, safe_status, risk_assess}

Σ1 = {initiate_purge, tcyc_comp,_psicyc_comp, risk_action, evaluate_risk, perform_alteration}

Σ2 = {lock_doors, unlock_doors, set risk}

q0 : prep_vpurge

V = {risk}

$\Lambda_{refined}$ ={

1. → prep_vpurge
2. prep_vpurge $\xrightarrow{initiate\_purge/lock\_doors}$ alt_temp
3. prep_vpurge $\xrightarrow{initiate\_purge/lock\_doors}$ alt_psi
4. alt_temp $\xrightarrow{perform\_alteration}$ alt_temp
5. alt_temp $\xrightarrow{tcyc\_comp}$ risk_assess
6. alt_psi $\xrightarrow{perform\_alteration}$ alt_psi

7. alt_psi $\xrightarrow{tcyc\_comp}$ risk_assess
8. risk_assess $\xrightarrow{evaluate\_risk/set\ risk}$ risk_assess
9. risk_assess $\xrightarrow{risk\_action[risk \leq 1]/unlock\_doors,set\ risk}$ safe_status
10. risk_assess $\xrightarrow{risk\_action[risk > 1]/set\ risk}$ prep_vpurge

}