Olivier Brochu Dufour (26964834), Maxwell Mowbray(7005334)

SOEN 331 As. 3 – 30/03/2015

A specification S is defined as a tuple S = (Q, ∑1, ∑2, q0, V, Λ) **where**

Q = {} the set of states

∑1 = {} the set of events

∑2 = {} the set of actions

q0 = the initial state

V = {} the set of variables (global)

Λ = {} the set of transitions between states

**Part I: Overall system**

S = (Q, ∑1, ∑2, q0, V, Λ)

Q = {dormant, init, idle, monitoring, error\_diagnosis, safe\_shutdown}

∑1 = {start, init\_ok, begin\_monitoring, monitor\_crash, moni\_rescue, idle\_crash, idle\_rescue, init\_crash, retry\_init, shutdown, sleep}

∑2 = {moni\_err\_msg, broadcast idle\_err\_msg, broadcast init\_err\_msg, retry++}

q0 : dormant

V : retry :  ℕ0 initialized to 0

Λ : Transition specifications

start

1. dormant init

init\_ok

2. init idle

begin\_monitoing

3. idle monitoring

monitor\_crash/moni\_err\_msg

4. monitoring error\_diagnosis

moni\_rescue

5. error\_diagnosis monitoring

idle\_crash/broadcast idle\_err\_msg

6. idle error\_diagnosis

idle\_rescue

7. error\_diagnosis idle

init\_crash/broadcast init\_err\_msg

8. init error\_diagnosis

retry\_init[retry<3]/retry++

9. error\_diagnosis init

shutdown[retry>=3]

10. error\_diagnosis safe\_shutdown

sleep

11. safe\_shutdown dormant

**Part II: Refine init**

S = (Q, ∑1, ∑2, q0, V, Λ)

Q = {boot\_hw, senchk, tchk, psichk, ready}

∑1 = {hw\_ok, senok, t\_ok, psi\_ok}

∑2 = {}

q0 : boot\_hw

V : {}

Λ : Transition specifications

hw\_ok

1. boot\_hw senchk

senok

2. senchk tchk

t\_ok

3. tchk psichk

psi\_ok

4. psichk ready

**Part III: Refine monitoring**

S = (Q, ∑1, ∑2, q0, V, Λ)

Q = {mon\_idle, regulate\_environment, lockdown}

∑1 = {no\_contagion, contagion\_alert, after\_100ms, purge\_succ}

∑2 = {broadcast FACILITY\_CRIT\_MESG}

q0 : regulate\_environment

V : inlockdown = {true, false}

Λ : Transition specifications

no\_contagion

1. mon\_idle regulate\_environment

after\_100ms

2. regulate\_environment mon\_idle

contagion\_alert / broadcast FACILITY\_CRIT\_MESG

3. mon\_idle lockdown

purge\_succ/inlockdown := false

4. lockdown mon\_idle

**Part IV: Refine lockdown**

S = (Q, ∑1, ∑2, q0, V, Λ)

Q = {prep\_vpurge, alt\_temp, alt\_psi, risk\_assess, safe\_status, lockdown\_exit}

∑1 = {initiate\_purge, tcyc\_comp, psicyc\_comp}

∑2 = {lock\_doors, unlock\_doors}

q0 : prep\_vpurge

V :

Λ : Transition specifications

Initiate\_purge/lock\_doors

1. prep\_vpurge alt\_temp

Initiate\_purge/lock\_doors

2. prep\_vpurge alt\_psi

tcyc\_comp

3. alt\_temp risk\_assess

psicyc\_comp

4. alt\_psi risk\_asssess

[risk > 1%]

5. risk\_assess prep\_vpurge

[risk < 1%]

6. risk\_assess lockdown\_exit

/unlock\_doors

7. safe\_status lockdown\_exit

**Part V: Refine error diagnosis (robust error handling)**

S = (Q, ∑1, ∑2, q0, V, Λ)

Q = {error\_rcv, applicable\_rescue, reset\_module\_data, error\_diagnosis\_exit }

∑1 = {apply\_protocol\_rescues, reset\_to\_stable}

∑2 = {}

q0 : error\_rcv

V : err\_protocol\_def = {true, false}

Λ : Transition specifications

[error\_protocol\_def = true]

1. error\_rcv applicable\_rescue

[error\_protocol\_def = false]

2. error\_rcv reset\_module\_data

apply\_protocol\_rescues

3. applicable\_rescue error\_diagnosis\_exit

reset\_to\_stable

4. reset\_module\_data error\_diagnosis\_exit