**SYSC 5104: Methodologies for Discrete-Event Modeling and Simulation**

**Instructor: Prof. Gabriel A. Wainer**

**Powered Compact Shelving System**

**Assignment 1**

**Prepared By: Zaheenabanu Somani**

**Student Id: 101054500**

**Part I:**

Powered Compact shelving systems are readily used in offices, industries and educational institutions to increase storage space by closely packing each unit when access is not required. It consists of:

1. AisleCheck: Pressing MOVE button on adjacent aisle will check the consecutive aisles and open the desired aisle.
2. Walk-in detection: The entry sensors detect any object in an open aisle and locks open the aisle.
3. Walk-out detection: The aisle is unlocked when exit sensor is interrupted.
4. Counter: When walk-in is detected counter increases to 1 and when walk-out is detected counter decreases to 0.

The real time system of powered compact shelving system works as follows:

On pressing the move button (left/right/reset) on the adjacent aisle of the desired aisle state of the consecutive aisles is checked. When an object interrupts the entry sensor the counter increases to 1 and the aisle is locked open and cannot be closed. During this time if another aisle is operated the aisle checker will detect an open aisle and the operation will be stopped.

If an aisle is empty the time to open the aisle is calculated. The motor will move in the button (left/right) direction and the desired aisle opens in calculated time.

On exit from the aisle when the exit sensor is interrupted the counter decreases to 0 and the aisle unlocks. After an aisle has been unlocked RESET button adjacent to the open aisle is used to initialize the movement of units.

**Figure 1. Powered Compact Shelving System Model**

onExit

Entry\_Sensor

Exit\_Sensor

decr

incr

AisleCheck

onReset

update

aisleOpen

Timer

stop

Aisle\_unlocked

Calc\_Aisles

Counter

aisleLocked

walk-out

walk-in

reset

moveRight

moveLeft

Figure 1 consists of a top model, one structured model: AisleCheck and six atomic models: Entry\_Sensor, Exit\_Sensor, Counter, Aisle\_unlocked, Calc\_Aisles, Timer.

**PART II:**

The input output ports and links for the system are as follows:

[Top]

components : aisleCheck calcAisles@CalcAisles timer@Timer

out : aisleLocked stop aisleOpen

in : moveLeft moveRight reset walk\_out walk\_in

Link : moveLeft moveLeft@aisleCheck

Link : moveRight moveRight@aisleCheck

Link : reset reset@aisleCheck

Link : aisleLocked@aisleCheck aisleLocked

Link : stop@aisleCheck stop

Link : onReset@aisleCheck onReset@calcAisles

Link : update@calcAisles update@timer

Link : aisleOpen@timer aisleOpen

Link : walk\_in@timer walk\_in

Link : walk\_in walk\_in@aisleCheck

Link : walk\_out walk\_out@aisleCheck

[calcAisles]

components : calcAisles@CalcAisles

out : update

in : onReset

Link : onReset onReset@calcAisles

Link : update@calcAisles update

[timer]

components : timer@Timer

out : aisleOpen walk\_in

in : update

Link : update update@timer

Link : aisleOpen@timer aisleOpen

Link : walk\_in@timer walk\_in

[aisleCheck]

components : counter@Counter entrySensor@EntrySensor exitSensor@ExitSensor aisleUnlocked@AisleUnlocked

out : aisleLocked onReset stop

in : moveLeft moveRight reset walk\_in walk\_out

Link : moveLeft moveLeft@counter

Link : moveRight moveRight@counter

Link : walk\_in walk\_in@entrySensor

Link : incr@entrySensor incr@counter

Link : walk\_out walk\_out@exitSensor

Link : decr@exitSensor decr@counter

Link : aisleLocked@counter aisleLocked

Link : onExit@counter onExit@aisleUnlocked

Link : reset reset@aisleUnlocked

Link : stop@aisleUnlocked stop

Link : onReset@aisleUnlocked onReset

[counter]

components : counter@Counter

out : aisleLocked onExit

in : moveLeft moveRight incr decr

Link : moveLeft moveLeft@counter

Link : moveRight moveRight@counter

Link : incr incr@counter

Link : decr decr@counter

Link : aisleLocked@counter aisleLocked

Link : onExit@counter onExit

[entrySensor]

components : entrySensor@EntrySensor

out : incr

in : walk\_in

Link : walk\_in walk\_in@entrySensor

Link : incr@entrySensor incr

[exitSensor]

components : exitSensor@ExitSensor

out : decr

in : walk\_out

Link : walk\_out walk\_out@exitSensor

Link : decr@exitSensor decr

[aisleUnlocked]

components : aisleUnlocked@AisleUnlocked

out : onReset stop

in : reset onExit

Link : reset reset@aisleUnlocked

Link : onExit onExit@aisleUnlocked

Link : onReset@aisleUnlocked onReset

Link : stop@aisleUnlocked stop

**FORMAL SPECIFICATIONS**

Powered Compact Shelving System consists of one coupled: AisleCheeck and six atomic models: Entry\_Sensor, Exit\_Sensor, Counter, Aisle\_unlocked, Calc\_Aisles, Timer. AisleCheck consists of four atomic models: Entry\_Sensor, Exit\_Sensor, Counter, Aisle\_unlocked.

The formal specifications of these models are given below. More information about states and time advance function changes; can be seen from “Part-2-C State Diagrams” section of this report.

**1. Counter:**

Counter = < S, X, Y, δint, δext, λ, ta >

X = {moveLeft, moveRight, incr, decr}

Y = {aisleLocked, onExit}

State Variables: {phase, ta, ifLocked, ifExit}

Internal Transfer Function:

δint(S2 )= S1, δint(S3 )= S1, δint(S4 )= S1, δint(S5 )= S1

External Transfer Function:

If counter is 0, δext(S1, moveLeft/moveRight)=S2

If counter is 1, δext(S1, moveLeft/moveRight)=S3

When counter is 0, δext(S1, incr)=S4, counter = 1

When counter is 1, δext(S1, decr)=S5, counter = 0

Output Function:

λ(S2->S1) =onExit, λ(S3->S1) =aisleLocked, λ(S4->S1) =aisleLocked, λ(S3->S1) =onExit

**2. EntrySensor:**

EntrySensor = < S, X, Y, δint, δext, λ, ta >

X = {walk\_in}

Y = {incr}

S = {phase, ta, ifincr}

Internal Transfer Function:

δint(S2 )= S1

External Transfer Function:

δext(S1, walk\_in)=S2

Output Function:

λ(S2->S1) =incr

**3. ExitSensor:**

ExitSensor = < S, X, Y, δint, δext, λ, ta >

X = {walk\_out}

Y = {decr}

S = {phase, ta, ifdecr}

Internal Transfer Function:

δint(S2 )= S1

External Transfer Function:

δext(S1, walk\_out)=S2

Output Function:

λ(S2->S1) = decr

**4. AisleUnlocked:**

AisleUnlocked = < S, X, Y, δint, δext, λ, ta >

X = {reset, onExit}

Y = {onReset, stop}

S = {phase, ta, ifReset, ifExit}

Internal Transfer Function:

δint(S2 )= S1, δint(S3 )= S1

External Transfer Function:

δext(S1, onExit)=S2

δext(S2, reset)=S3

Output Function:

λ(S2->S1) = stop, λ(S3->S1) = onReset

**5. Calcuate Aisles:**

CalcAisles = < S, X, Y, δint, δext, λ, ta >

X = {onReset}

Y = {update}

S = { phase ta, ifupdate}

Internal Transfer Function:

δint(S2 )= S1

External Transfer Function:

δext(S1, onReset)=S2

Output Function:

λ(S2->S1) = update

**6. Timer:**

Timer = < S, X, Y, δint, δext, λ, ta >

X = {update}

Y = {aisleOpen, walk\_in}

S = {phase, ta, ifaisleOpen}

Internal Transfer Function:

δint(S2 )= S1

External Transfer Function:

δext(S1, update)=S2

Output Function:

λ(S2->S1) = aisleOpen

**7. AisleCheck:**

AisleCheck = < X, Y, D, {Counter, EntrySensor, ExitSensor, AisleUnlocked}, {Ii}, {Zij}, select >

X = {moveLeft, moveRight, reset, walk\_in, walk\_out}

Y = {aisleLocker, onReset, stop}

I(Counter) = {AisleLocked, AisleUnlocked}

Zij:

If moveLeft/moveRight and counter = 0 then send output: stop.

If moveLeft/moveRight and counter = 0 and reset, then send output: ifReset.

If moveLeft/moveRight and counter = 1, then send output: aisleLocked

**1. STATE DIAGRAM FOR COUNTER:**

# S3

Phase= Active

Ta(s) = 0

IfExit = F

IfLocked = T

(Y: aisleLocked)

(X: moveLeft/moveRight counter=1)

(X: moveLeft /moveRight counter =0)

# S2

Phase= Active

Ta(s) = 0

IfExit = T

IfLocked = F

# S1

Phase= Passive

Ta(s) = inf

IfExit = F

IfLocked = F

(Y: onExit)

(Y: onExit)

# S5

Phase= Active

Ta(s) = 0

IfExit = T

IfLocked = F

(X: incr )

counter = 0

(Y: aisleLocked)

(X: decr )

counter = 1

# S4

Phase= Active

Ta(s) = 0

IfExit = F

IfLocked = T

**2. STATE DIAGRAM FOR ENTRYSENSOR:**

# S2

Phase= Active

Ta(s) = 0

Ifincr = T

# S1

Phase= Passive

Ta(s) = Inf

Ifincr = F

(X: walk\_in)

Y: incr (counter = 1)

**3. STATE DIAGRAM FOR EXITSENSOR:**

# S2

Phase= Active

Ta(s) = 0

Ifdecr = T

# S1

Phase= Passive

Ta(s) = Inf

Ifdecr = F

(X: walk\_out)

Y: decr (counter = 0)

**4. STATE DIAGRAM FOR AISLEUNLOCKED:**

(X: onExit)

# S2

Phase= Active

Ta(s) = 0

IfExit = T

IfReset = F

# S1

Phase= Passive

Ta(s) = inf

IfExit = F

IfReset = F

(Y: stop)

(X: reset)

(Y: onReset)

# S3

Phase= Active

Ta(s) = 0

IfExit = T

IfReset = T

**5. STATE DIAGRAM FOR CALCAISLES:**

# S2

Phase= Active

Ta(s) = 0

Ifupdate = T

# S1

Phase= Passive

Ta(s) = Inf

Ifupdate = F

(X: onReset)

(Y: update)

**6. STATE DIAGRAM FOR TIMER:**

# S2

Phase= Active

Ta(s) = 0

IfaisleOpen = T

# S1

Phase= Passive

Ta(s) = Inf

IfaisleOpen = F

(X: update)

(Y: aisleOpen)

**TEST STRATEGIES AND RESULTS:**

First all atomic models are tested. Then the coupled model AisleCheck, which contains four atomic models, is tested. Finally the Powered Compact Shelving System as a whole model is tested.

**Test Cases:**

In order to check if the model behaves like the real powered compact shelving system, the following test cases are prepared:

Test Case 1:

If model input is “moveLeft/moveRight”, and counter is 0 the system will stop. If it is already open, counter is 1 and aisle is locked open(aisleLocked).

Test Case 2:

If model input is “moveLeft/moveRight” and “reset”, and counter is 0 the system will open. If it is already open, counter is 1 and aisle is locked open(aisleLocked).

Test Case 3:

If the input is walk\_in and the counter is 0, the counter will increase to 1 and the aisle is locked.

Test Case 4:

If the input is walk\_out and the counter is 1, the counter will decrease to 0, on reset the aisle will open.

**TEST RESULTS:**

**1. Counter:**

Input File:

00:00:11:00 moveRight 1

00:01:11:00 moveRight 1

00:02:00:00 moveRight 1

00:02:15:00 moveRight 1

00:02:50:00 moveRight 1

00:03:20:00 moveRight 1

00:03:40:00 moveRight 1

Output File:

00:00:11:000 onexit 1

00:01:11:000 onexit 1

00:02:00:000 onexit 1

00:02:15:000 onexit 1

00:02:50:000 onexit 1

00:03:20:000 onexit 1

00:03:40:000 onexit 1

**2. EntrySensor:**

Input File:

00:00:08:00 walk\_in 1

00:00:26:00 walk\_in 1

00:01:05:00 walk\_in 1

00:01:30:00 walk\_in 1

00:02:08:00 walk\_in 1

00:02:26:00 walk\_in 1

00:03:00:00 walk\_in 1

Output File:

00:00:08:000 incr 1

00:00:26:000 incr 1

00:01:05:000 incr 1

00:01:30:000 incr 1

00:02:08:000 incr 1

00:02:26:000 incr 1

00:03:00:000 incr 1

**3. ExitSensor:**

Input File:

00:00:08:00 walk\_in 1

00:00:26:00 walk\_in 1

00:01:05:00 walk\_in 1

00:01:30:00 walk\_in 1

00:02:08:00 walk\_in 1

00:02:26:00 walk\_in 1

00:03:00:00 walk\_in 1

Output File:

00:00:10:000 decr 1

00:00:30:000 decr 1

00:01:10:000 decr 1

00:01:45:000 decr 1

00:02:10:000 decr 1

00:02:30:000 decr 1

00:03:00:000 decr 1

**4. AisleUnlocked:**

Input File:

00:01:00:00 reset 1

00:03:10:00 onExit 2

Output File:

00:01:00:000 onreset 1

00:03:10:000 stop 1

**5. CalcAisles:**

Input File:

00:01:10:00 onReset 1

00:02:00:00 onReset 1

00:02:20:00 onReset 1

00:02:45:00 onReset 1

00:03:15:00 onReset 1

00:03:50:00 onReset 1

00:04:00:00 onReset 1

Output File:

00:01:10:000 update 1

00:02:00:000 update 1

00:02:20:000 update 1

00:02:45:000 update 1

00:03:15:000 update 1

00:03:50:000 update 1

00:04:00:000 update 1

**6. Timer:**

Input File:

00:04:20:00 update 1

Output File:

00:04:45:000 aisleopen 1

**7. AisleCheck:**

Input File:

00:00:10:00 moveLeft 1

00:01:00:00 reset 2

00:02:20:00 walk\_in 3

00:02:50:00 moveRight 2

Output File:

00:00:10:000 stop 1

00:01:00:000 onreset 1

00:02:50:000 aislelocked 1

**8. PCSS:**

Input File:

00:00:10:00 moveLeft 1

00:01:10:00 reset 2

00:02:30:00 walk\_in 3

00:02:50:00 moveRight 4

00:03:15:00 walk\_out 5

00:03:45:00 reset 6

Output File:

00:00:10:000 stop 1

00:01:35:000 aisleopen 1

00:02:50:000 aislelocked 1

00:04:25:000 aisleopen 1

00:04:45:000 aislelocked 1

**TEST RESULTS:**

All test cases passed. The model works as expected.