**Student name: WEN WEN**

**Student number: 6689880**

**University of Ottawa**

**ASSIGNMENT 1:**

**Title: Weather Detecting Sensor Network Operation Simulator**

**PartⅠ**

**Problem Description:** People nowadays using wireless sensor networks to detect the weather and record the changes of the local climate. Such kind of sensor networks contains a lot of sensors which could work in harsh environments and those sensors surely have to transit the data they collected to the centralized server. To make sure the data which are collected by sensors are reliable, we calculate the average value of data that collected by different sensors in the same time, which we marked as value1, and then we calculate the average value of value1 collected in different time, by doing this, we could also detecting the abruptness of temperature change by calculating the variance of all the data we collected.

**Components:**

This simulation system consists of three major components: time slot, H\_average and D\_average. And H\_average and D\_average also contain some subcomponents.

**H\_average** is used to calculate the average temperature and the abruptness of temperature change for one hour, and together with the time slot, it could generate an output every minute. To achieve those functions, H\_average also contain four smaller components, each of them has different functions.

**D\_average** is used to calculate the average temperature and the abruptness of temperature change for a single day.

Plus: **Time slot** is used to control the time advance in order to send outputs at designed time points. Assume the time slot is a build-in pulse generator which will generate digital signal sequence in the order of “01010101”, and every bit lasts one second. And because Time slot and atomic model Time makes all the outputs to be generated in particular time points, so there is no need to control time advance in the later steps.

**Frame:**

Time slot

**TOP model:**

Detector Simulator[top]

Sensor network

Terminal device

H\_average

D\_average

**Coupled Model:**

**H\_average:**

Out2

Out1

In1

C

Time

In2

B

A

**D\_average:**

D

Out2

Out1

In

E

**The behavior of each coupled model and atomic model:**

1. When the sensor network generated enough figures and transmitted them to the reader, the reader, in this case is the reader simulator, will first send the signals to the coupled model H\_average, and in the coupled model, the data will be calculated first by atomic model A, assume there are 10 sensors, and in one minute, they could generate 10 figures to represent the current temperature of the local area, and A will generate the summary of the 10 figures. What is more, together with atomic model Time, the summary value will be generated at the end of each minute.
2. The summary value will be used in two atomic models. In atomic model B, the summary value will first be divided by 10, then stored in a array, when there is 60 values in the array, which means one hour has passed, the atomic model B will summarize all the 60 values and divide the summary value by 60 to get the average temperature of that hour. The send out the average temperature. In atomic model C, the summary value will also be divided by 10, then be stored in an array. When atomic model C received the output from atomic model B, C starts to calculate the variance of the 60 values. The variance we got could represent the abruptness of temperature change within one hour.
3. One of the outputs, average temperature of an hour, is the input of coupled model D\_average. When we got the average value of 24 average temperatures of 24 continuously hours, we could roughly got the average temperature of the whole day. This calculation is done by atomic model D. Also, the variance of the 24 values, which could be calculated in atomic model E associate with the output of atomic model D, shows the abruptness of temperature change.

**PartⅡ**

**Formal specifications:**

The formal specifications <S, X, Y, δint, δext, λ, ta> for the atomic models are defined as follows:

**Time:**

S = {passive, active}

X = {in}

Y = {out}

δint (active) = passive

δext (in, passive) = active

δext (in, active) = active

λ (active)

{

Send 1 to output port; //used to activate the output of atomic model A

}

ta(passive)=INFINITY

ta(active)=preparationTime



A:

S = {passive, active}

X = {in1, in2}

Y = {out}

δint (active) = passive

δext (in1, in2, passive)

{

If(in1&& n<10)

Read S; n=n+1; //n is the number of data which have been added

passive;

If(in1&& n=10)

active;

if(in2)

read T;

}

δext (in1, in2, active) = active

λ (active)

{

If(T==1&&active)

Send output S;

}

ta(passive)=INFINITY

ta(active)=Zero



B:

S = {passive, active}

X = {in}

Y = {out}

δint (active) = passive

δext (in, passive)

{

If(in)

Read A;

N=N+1;

passive

If(N==60)

active

δext (in, active) = active

λ (active)

{

If(istate=true)

B= (average value of 60 As) /10

Send output B;

}

ta(passive)=INFINITY

ta(active)=Zero



C:

S = {passive, active}

X = {in1, in2}

Y = {out}

δint (active) = passive

δext (in1, in2, passive)

{

if(in1)

Read S[N];

T[N]=S[N]/10;

N++;

If(N==60)

If(in2)

Read AVG;

for i=0 to 59 step 60

{

D=D+(T[i]-AVG)2/60

}

}

δext (in1, in2, active) = active

λ (active)

{

If(N==60)

//N stands for the number of data received by input port 1 and also the size of the array

{send out output D;}

}

ta(passive)=INFINITY

ta(active)=Zero



D:

S = {passive, active}

X = {in}

Y = {out}

δint (active) = passive

δext (in, passive)

{

If(in)

Sum=Sum+(value for port in);

N++;

If(N==24)

AVG=Sum/24;

}

δext (in, active) = active

λ (active)

{

If(N==24) //N stands for 24 hours of a day

{

Send output AVG;

}

}

ta(passive)=INFINITY

ta(active)=Zero



E:

S = {passive, active}

X = {in1, in2}

Y = {out}

δint (active) = passive

δext (in1, in2, passive)

{

If(in2)

Read S[N];

If(N=24)

If(in1)

Read AVG;

For i=0 to 23 step 24

{

D=D+(S[i]-AVG)2/24;

}

}

δext (in1, in2, active) = active

λ (active)

{

If(N==24)

//N stands for the number of data received by input port 2 and also the size of the array

{send out output D;}

}

ta(passive)=INFINITY

ta(active)=Zero



The formal specifications <X, Y, D, {Mi}, {Ii}, {Zij}, SELECT > for coupled model H\_average and D\_average:

H\_average:

X = {in1, in2};

Y = {out1, out2};

D = {Time, A, B, C};

I(Time) = A;

I(A) = { B, C};

I(B) = { C, self};

I(C) = self;

Z(Time) = A;

Z(A) = B; Z(A) = C;

Z(B) = C; Z(B) = self;

Z(C) = self;

SELECT: ;

D\_average:

X = {in};

Y= {out1, out2};

D ={D, E}

I(D) = E;

I(E) = self;

Z(D) = E;

Z(E) = self;

SELECT:

Reader Simulator:

X = {in1, in2};

Y = {out1, out2, out3, out4};

D = {H\_average, D\_average};

I(H\_average) = {D\_average, self};

I(D\_average) = self;

Z(H\_average) = D\_average; Z(H\_average) = self;

Z(D\_average) = self;

SELECT:

**Test strategies:**

Using black box strategy, create *.ev* files, which carry all the input information we need, to each atomic and coupled model, then we run the simulation to see whether the outputs match our expectation.

**PartⅢ**

**Test content:**

1. Time:

Because of the delay and the collision, the preparation time is set as 59 seconds, and when a positive pulse came from the time slot at 00:00:01:00, after 59 seconds, there will be a output generated at 00:01:00:00. And the time interval of two incoming pulses must longer than one second, and if the time accuracy is highly expected, for example, we could set the preparation time as 59.9999 and the time interval could be 0.0001.

The .ev file is indicated as follows:

00:00:01:00 in 1

00:00:05:00 in 1

00:00:09:00 in 1

00:01:01:00 in 1

00:02:01:00 in 1

00:03:01:00 in 1

00:04:01:00 in 1

00:05:01:00 in 1

The outputs are as follows:

00:01:00:000 out 1

00:02:00:000 out 1

00:03:00:000 out 1

00:04:00:000 out 1

00:05:00:000 out 1

00:06:00:000 out 1

1. A

The function of atomic model A is to calculate the summary of 10 data generated by sensors in the sensor network every minute. Since the data is generated synchronous and the transmission time is relatively short, all the data will come at the first 1 or 2 seconds. To examine the correctness of A, the inputs sometimes come with a time interval of 1 second, sometimes 1 ms and even 0. And the input port in2, receives the outputs come from atomic model Time.

The .ev file is defined as follows:

00:00:00:01 in1 13

00:00:00:02 in1 14

00:00:00:03 in1 12

00:00:00:04 in1 15

00:00:00:05 in1 13

00:00:00:06 in1 12

00:00:00:07 in1 11

00:00:00:08 in1 12

00:00:00:09 in1 12

00:00:00:10 in1 12

00:01:01:00 in1 13

00:01:01:00 in1 14

00:01:01:00 in1 12

00:01:02:00 in1 15

00:01:02:00 in1 13

00:01:03:00 in1 12

00:01:03:00 in1 11

00:01:04:00 in1 12

00:01:05:00 in1 12

00:01:06:00 in1 12

00:01:00:00 in2 1

00:02:00:00 in2 1

The outputs are as follows:

00:01:00:000 out 126

00:02:00:000 out 126

1. B

Atomic model B is used to calculate average value. The time advance is controlled by Time, and the inputs are the outputs of atomic model A, so the inputs will come with a time interval of 1 minute, and only after 60 inputs, there will be an output generated.

The .ev file is as follows:

00:01:00:00 in 125

00:02:00:00 in 130

00:03:00:00 in 132

00:04:00:00 in 134

00:05:00:00 in 126

00:06:00:00 in 125

00:07:00:00 in 130

00:08:00:00 in 132

00:09:00:00 in 134

00:10:00:00 in 126

00:11:00:00 in 125

00:12:00:00 in 130

00:13:00:00 in 132

00:14:00:00 in 134

00:15:00:00 in 126

00:16:00:00 in 125

00:17:00:00 in 130

00:18:00:00 in 132

00:19:00:00 in 134

00:20:00:00 in 126

00:21:00:00 in 125

00:22:00:00 in 130

00:23:00:00 in 132

00:24:00:00 in 134

00:25:00:00 in 126

00:26:00:00 in 125

00:27:00:00 in 130

00:28:00:00 in 132

00:29:00:00 in 134

00:30:00:00 in 126

00:31:00:00 in 125

00:32:00:00 in 130

00:33:00:00 in 132

00:34:00:00 in 134

00:35:00:00 in 126

00:36:00:00 in 125

00:37:00:00 in 130

00:38:00:00 in 132

00:39:00:00 in 134

00:40:00:00 in 126

00:41:00:00 in 125

00:42:00:00 in 130

00:43:00:00 in 132

00:44:00:00 in 134

00:45:00:00 in 126

00:46:00:00 in 125

00:47:00:00 in 130

00:48:00:00 in 132

00:49:00:00 in 134

00:50:00:00 in 126

00:51:00:00 in 125

00:52:00:00 in 130

00:53:00:00 in 132

00:54:00:00 in 134

00:55:00:00 in 126

00:56:00:00 in 125

00:57:00:00 in 130

00:58:00:00 in 132

00:59:00:00 in 134

00:60:00:00 in 126

The output is as follows:

01:00:00:000 out 12.94

The result is the same with my expectation.

1. C

Atomic model C is used to calculate variances. Variance could represent the abruptness of the temperature change. The bigger the variance value is, the greater the temperature fluctuates.

The .ev file is as follows:

00:01:00:00 in1 125

00:02:00:00 in1 130

00:03:00:00 in1 132

00:04:00:00 in1 134

00:05:00:00 in1 126

00:06:00:00 in1 125

00:07:00:00 in1 130

00:08:00:00 in1 132

00:09:00:00 in1 134

00:10:00:00 in1 126

00:11:00:00 in1 125

00:12:00:00 in1 130

00:13:00:00 in1 132

00:14:00:00 in1 134

00:15:00:00 in1 126

00:16:00:00 in1 125

00:17:00:00 in1 130

00:18:00:00 in1 132

00:19:00:00 in1 134

00:20:00:00 in1 126

00:21:00:00 in1 125

00:22:00:00 in1 130

00:23:00:00 in1 132

00:24:00:00 in1 134

00:25:00:00 in1 126

00:26:00:00 in1 125

00:27:00:00 in1 130

00:28:00:00 in1 132

00:29:00:00 in1 134

00:30:00:00 in1 126

00:31:00:00 in1 125

00:32:00:00 in1 130

00:33:00:00 in1 132

00:34:00:00 in1 134

00:35:00:00 in1 126

00:36:00:00 in1 125

00:37:00:00 in1 130

00:38:00:00 in1 132

00:39:00:00 in1 134

00:40:00:00 in1 126

00:41:00:00 in1 125

00:42:00:00 in1 130

00:43:00:00 in1 132

00:44:00:00 in1 134

00:45:00:00 in1 126

00:46:00:00 in1 125

00:47:00:00 in1 130

00:48:00:00 in1 132

00:49:00:00 in1 134

00:50:00:00 in1 126

00:51:00:00 in1 125

00:52:00:00 in1 130

00:53:00:00 in1 132

00:54:00:00 in1 134

00:55:00:00 in1 126

00:56:00:00 in1 125

00:57:00:00 in1 130

00:58:00:00 in1 132

00:59:00:00 in1 134

00:60:00:00 in1 126

01:00:00:00 in2 12.94

The output is as follows:

01:00:00:000 out 0.1184

1. D

The outputs of the last coupled model, more specific, the outputs of atomic model B, will become the income of atomic model D and E. D is used to calculate the average value of inputs, it will generate output after 24 inputs calculated. So, the output will be generated once every 24 hours.

The .ev file is as follows:

01:0:0:0 in 12.5

02:0:0:0 in 13

03:0:0:0 in 13.2

04:0:0:0 in 13.4

05:0:0:0 in 12.6

06:0:0:0 in 12.5

07:0:0:0 in 13

08:0:0:0 in 13.2

09:0:0:0 in 13.4

10:0:0:0 in 12.6

11:0:0:0 in 12.5

12:0:0:0 in 13

13:0:0:0 in 13.2

14:0:0:0 in 13.4

15:0:0:0 in 12.6

16:0:0:0 in 12.5

17:0:0:0 in 13

18:0:0:0 in 13.2

19:0:0:0 in 13.4

20:0:0:0 in 12.6

21:0:0:0 in 12.5

22:0:0:0 in 13

23:0:0:0 in 13.2

24:0:0:0 in 13.4

25:0:0:0 in 12.6

26:0:0:0 in 12.5

27:0:0:0 in 13

28:0:0:0 in 13.2

29:0:0:0 in 13.4

30:0:0:0 in 12.6

31:0:0:0 in 12.5

32:0:0:0 in 13

33:0:0:0 in 13.2

34:0:0:0 in 13.4

35:0:0:0 in 12.6

36:0:0:0 in 12.5

37:0:0:0 in 13

38:0:0:0 in 13.2

39:0:0:0 in 13.4

40:0:0:0 in 12.6

41:0:0:0 in 12.5

42:0:0:0 in 13

43:0:0:0 in 13.2

44:0:0:0 in 13.4

45:0:0:0 in 12.6

46:0:0:0 in 12.5

47:0:0:0 in 13

48:0:0:0 in 13.2

The outputs are as follows:

24:00:00:000 out 12.9542

48:00:00:000 out 12.9208

1. E

This atomic model has the same function with atomic model C, they all used to calculate the variance of the inputs, but C also has to do average of each input, because A only provides summaries of data.

The .ev file is indicated below:

01:00:00:00 in2 12.5

02:00:00:00 in2 13

03:00:00:00 in2 13.2

04:00:00:00 in2 13.4

05:00:00:00 in2 12.6

06:00:00:00 in2 12.5

07:00:00:00 in2 13

08:00:00:00 in2 13.2

09:00:00:00 in2 13.4

10:00:00:00 in2 12.6

11:00:00:00 in2 12.5

12:00:00:00 in2 13

13:00:00:00 in2 13.2

14:00:00:00 in2 13.4

15:00:00:00 in2 12.6

16:00:00:00 in2 12.5

17:00:00:00 in2 13

18:00:00:00 in2 13.2

19:00:00:00 in2 13.4

20:00:00:00 in2 12.6

21:00:00:00 in2 12.5

22:00:00:00 in2 13

23:00:00:00 in2 13.2

24:00:00:00 in2 13.4

25:00:00:00 in2 12.6

26:00:00:00 in2 12.5

27:00:00:00 in2 13

28:00:00:00 in2 13.2

29:00:00:00 in2 13.4

30:00:00:00 in2 12.6

31:00:00:00 in2 12.5

32:00:00:00 in2 13

33:00:00:00 in2 13.2

34:00:00:00 in2 13.4

35:00:00:00 in2 12.6

36:00:00:00 in2 12.5

37:00:00:00 in2 13

38:00:00:00 in2 13.2

39:00:00:00 in2 13.4

40:00:00:00 in2 12.6

41:00:00:00 in2 12.5

42:00:00:00 in2 13

43:00:00:00 in2 13.2

44:00:00:00 in2 13.4

45:00:00:00 in2 12.6

46:00:00:00 in2 12.5

47:00:00:00 in2 13

48:00:00:00 in2 13.2

24:00:00:000 in1 12.9542

48:00:00:000 in1 12.9208

The outputs are:

24:00:00:000 out 0.342748

48:00:00:000 out 0.356939

1. H\_average

This coupled model contains atomic model Time, A, B, and C. It has two input ports, one is linked to sensor network, the other one is linked to time slot. In order to make the simulation simple, I use number loop to be the inputs, so the variance I expect here is 0, and the average value is the same as the average value of {11,12,12,14,12,13,14,15,12,11}, which is 12.6.

The .ev file is as follows:

00:00:01:00 in1 11

00:00:02:00 in1 12

00:00:03:00 in1 12

00:00:04:00 in1 14

00:00:05:00 in1 12

00:00:06:00 in1 13

00:00:07:00 in1 14

00:00:08:00 in1 15

00:00:09:00 in1 12

00:00:10:00 in1 11

00:01:01:00 in1 11

00:01:02:00 in1 12

00:01:03:00 in1 12

00:01:04:00 in1 14

00:01:05:00 in1 12

00:01:06:00 in1 13

00:01:07:00 in1 14

00:01:08:00 in1 15

00:01:09:00 in1 12

00:01:10:00 in1 11

00:02:01:00 in1 11

……

00:59:09:00 in1 12

00:59:10:00 in1 11

00:00:01:00 in2 1

00:01:01:00 in2 1

00:02:01:00 in2 1

00:03:01:00 in2 1

……

00:58:01:00 in2 1

00:59:01:00 in2 1

The outputs are:

01:00:00:000 out1 12.6

01:00:00:000 out2 0

1. D\_average

This coupled model only has one input, because the inputs will come in particular time points, there is no need to set up a time control specially for this coupled model. The simulation next is a simulation of two days situation and the total simulation time is “48 hours”. Every day, it will generate two outputs, one is the average temperature of the local area, and the other one is the fluctuation of temperature of each hour.

The .ev file is as follows:

01:0:0:0 in 12.5

02:0:0:0 in 13

03:0:0:0 in 13.2

04:0:0:0 in 13.4

05:0:0:0 in 12.6

06:0:0:0 in 12.5

07:0:0:0 in 13

08:0:0:0 in 13.2

09:0:0:0 in 13.4

10:0:0:0 in 12.6

11:0:0:0 in 12.5

12:0:0:0 in 13

13:0:0:0 in 13.2

14:0:0:0 in 13.4

15:0:0:0 in 12.6

16:0:0:0 in 12.5

17:0:0:0 in 13

18:0:0:0 in 13.2

19:0:0:0 in 13.4

20:0:0:0 in 12.6

21:0:0:0 in 12.5

22:0:0:0 in 13

23:0:0:0 in 13.2

24:0:0:0 in 13.4

25:0:0:0 in 12.6

26:0:0:0 in 12.5

27:0:0:0 in 13

28:0:0:0 in 13.2

29:0:0:0 in 13.4

30:0:0:0 in 12.6

31:0:0:0 in 12.5

32:0:0:0 in 13

33:0:0:0 in 13.2

34:0:0:0 in 13.4

35:0:0:0 in 12.6

36:0:0:0 in 12.5

37:0:0:0 in 13

38:0:0:0 in 13.2

39:0:0:0 in 13.4

40:0:0:0 in 12.6

41:0:0:0 in 12.5

42:0:0:0 in 13

43:0:0:0 in 13.2

44:0:0:0 in 13.4

45:0:0:0 in 12.6

46:0:0:0 in 12.5

47:0:0:0 in 13

48:0:0:0 in 13.2

Outputs:

24:00:00:000 out1 12.9542

24:00:00:000 out2 0.342726

48:00:00:000 out1 12.9208

48:00:00:000 out2 0.356962

1. Top model

The simulation for top model is not complicated, but if I want to get the temperature information of 24 hours, the simulation will take too much time, and I tried, but the speed of my computer became very slow, so I have to simplified the model, I assume there are 5 sensors in the sensor network instead of 10, and every 10 minutes, the sensors will transmit samples to the reader only once instead of one time per minute. And I also assume that one day has only two hours instead of 24 hours, just to reduce the work of my computer. Even I did those simplifies, the simulation could also examine the correctness of my top model, because the logic is not changed.

The .ev file is as follows:

00:00:01:00 in1 13

00:00:01:00 in1 14

00:00:01:00 in1 12

00:00:01:00 in1 15

00:00:01:00 in1 13

00:01:01:00 in1 12

00:01:01:00 in1 11

00:01:01:00 in1 12

00:01:01:00 in1 12

00:01:01:00 in1 12

……

00:00:01:00 in2 1

00:10:01:00 in2 1

00:20:01:00 in2 1

00:30:01:00 in2 1

00:40:01:00 in2 1

00:50:01:00 in2 1

01:00:01:00 in2 1

01:10:01:00 in2 1

01:20:01:00 in2 1

01:30:01:00 in2 1

01:40:01:00 in2 1

01:50:01:00 in2 1

The outputs are:

01:00:00:000 out1 126

01:00:00:000 out2 0

02:00:00:000 out1 126

02:00:00:000 out2 0

02:00:00:000 out3 126

02:00:00:000 out4 0

So all the model I created are logically correct, and they all passed the simulation without errors.

Conclusion:

This simulator I create could simulate the normal situation of temperature detecting. And it is useful for many individuals and companies. For example, the government could place a sensor which could detect temperature to all the apartments and houses, the sensor network will collect huge information and transmit them to the centralized server. With the association of this kind of system, the government could easily monitor the temperature of each home. In Canada, there are some issues about the average indoor temperature. And the scientists place different kinds of sensors into different places, for example, if scientists want to know the temperature and the pressure in the deep sea, they could not diving every day in one particular ocean area, and the figures they want to get are fluctuated. For this situation, a sensor network is supposed to be used. Scientists could throw the sensors into the sea, and if the signal strength is strong enough, all the things the scientists need to do are sitting on the ship and checking the log of the sensors.

And there are also some places of this model could be improved and developed such as the time slot, it could be replaced by a DEVS graphs atomic model which could keep generating time intervals after receiving an activation signal. Also, during the final simulation, the huge volume of data became a great issue, if I want to finish the simulation of one single day, it may take me about 5 hours to send in the inputs and generate the outputs. What is more, this simulation is already simplified, and there are only 10 sensors in the sensor network, while in real cases, the numbers of sensors could be extremely large. Besides, the huge volume of data could cause some serious problems other than calculating speed. For example, there will be traffic conjunction and big time delay. The data reliability is also a great problem, because when the sensor network is extremely large, some data generated by sensors would lost, and the lost of data will affect the final result, so there should be a protocol which could make sure all the data we needed is transmitted successfully.