

**CARLETON UNIVERSITY**

**SYSC 5104 A**

**METHODOLOGIES FOR DISCRETE EVENT MODELING AND SIMULATION**

**Fall 2019**

**Assignment # 1**

**Student Academic Registration System (SARS)**

**Student: Shreya Taneja**

**Student Number: 101132722**

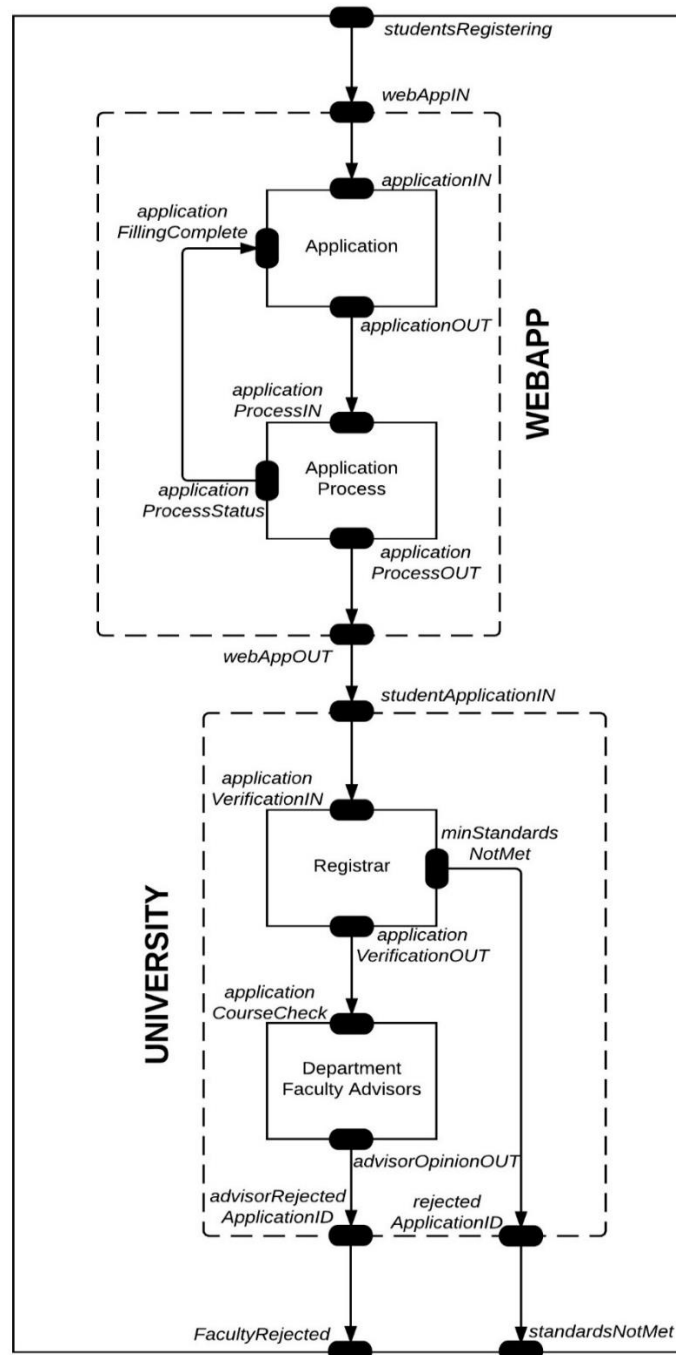
**Email: [shreyataneja@cmail.carleton.ca](mailto:shreyataneja@cmail.carleton.ca)**

## **INTRODUCTION**

The objective of the assignment is modeling of discrete event model using DEVS formalism and using CADMIUM simulation tool to execute discrete-event-simulations. Here we are adapting an existing CD++ model which we are testing and executing using CADMIUM. The formal specification of the model is defined and is then reimplemented using the CADMIUM simulation tool. The chosen DEVS model is Student Academic Registration System (SARS). Each model, sub model was defined in existing CD++ model which are executed and tested using CADMIUM. The conceptual model description for SARS is given as per the CD++ implementation.

## Conceptual Model Description

### Model Structure:



## **Problem to be Solved:**

Student registration system model consists of mainly two major blocks. In this simulation model, student's GPA, preferred courses and addition marks are assumed based on *probability* to simulate some real-world scenarios through web portal by filling the application. After filling the application There's a chance that the application might be with drawn, so application process takes 70% probability of submitting it.

The data is pushed to University's block where it is queued for registrar's approval. Once the registrar checks for minimum standards met by student based on probability it is either been accepted or rejected. If it has been accepted, the application is forwarded to Faculty advisor.

If the faculty finds the candidate suitable for the program and has a good GPA suitable for the position, he will accept it. The GPA too is calculated completely on probability and once it is rejected its marked as advisor rejected applicationID.

Student registration system keeps track of students, courses they have opted for and the status. Upon successful submission, a student ID is automatically assigned to the student for future reference purposes. Finally, within the registration period the Advisor do advising for the students by approving/rejecting requested course by the student.

Using the CADMIUM tool we obtain the simulation result in the output ports for all the applications submitted by the students that either not met by the minimum requirements or are rejected by the faculty.

## **Conceptual Model Description:**

This model represents a Student Academic Registration System. The model is composed of two sub models and both of the 2 sub models are coupled model and composed of two atomic models each.

The components of this application and their functions are as follow:

- WEBAPP model

This model is used to make all the students applications queue up, be processed in order and distribute those applications who want to apply for the designated university through university online process.

### **Components:**

- application
- application process

**Behavior of components:**

- application: This atomic model is used to make all the applications holding for filling procedures in a queue. Once it receives the feedback from the application process model considering submitting documents and submitting transcripts, it will let the next applications to be filled.
- applicationProcess: It distributes the new applications to the university. If he successfully filled the applications and chosen to submit the application.

- UNIVERSITY model

This model is aimed to make applications go through a series of check process (Administrations registrar, department faculty / advisor check) before accepting the application. As the result, not so good student applications are selected out.

**Components:**

- registrar
- faculty

**Behavior of components:**

- registrar: It could check whether the student met the minimum requirements of the university or if the applications are full and no more space is available of new students.
- faculty: It could check if the students course options is suitable for his/her goals and want to give the admission or not.

## **PART 2:**

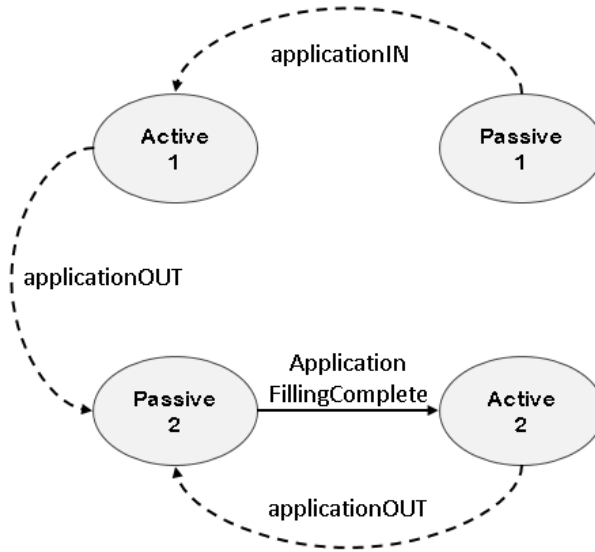
As you can see in figure 1, the SARS – Student Academic Registering System simulator has 1 input and 2 outputs. The *studentsregistering* means how many applications including the ones who have submitted the applications online and the ones that are still pending are go to university. The two outputs are showing the unsatisfied applications who fails in one of the series of checks, that is, registrars *minStandards*, advisors *advisorOpinion*. In terms of general overview, The application submission is made of two components: online submission and university verification. In addition, WEBAPP and UNIVERSITY models are decompressed to 2 components each for better understanding. WEBAPP has application and *applicationProcess* unit or atomic models connected in a series with a feedback links. Where as for the couple model UNIVERSITY is split in o two main atomic models, administrations block or most commonly known as *registrar* and the departments faculty who act as an advisors named as *advisor*, these are connected in series and each as their own output. These models are taken based on assumption that they have the similar behavior which can check if the application fit the standards of the university.

### **Formal Specifications of atomic models**

The more concrete DEVS formalism with port specifications is as follows:

$$\text{DEVS} = ( X, Y, S, \delta_{\text{ext}}, \delta_{\text{int}}, \lambda, t_a )$$

## Application:



$X = \{applicationIN, applicationFillingComplete\}$

$Y = \{1, studentNumber\}$

$S = \{“active1”, “active2”, “passive1”, “passive2”\}$

$\delta_{ext} (“passive1”, totalApplicationsCount) = “active1”$

$\delta_{ext} (“passive2”, applicationFillingComplete) = “active2”$

$\delta_{int} (“active1”, 1) = “passive1”$

$\delta_{int} (“active2”, applicationNum++) = “passive2”$

$\lambda (“active1”, applicationTime) = 1$

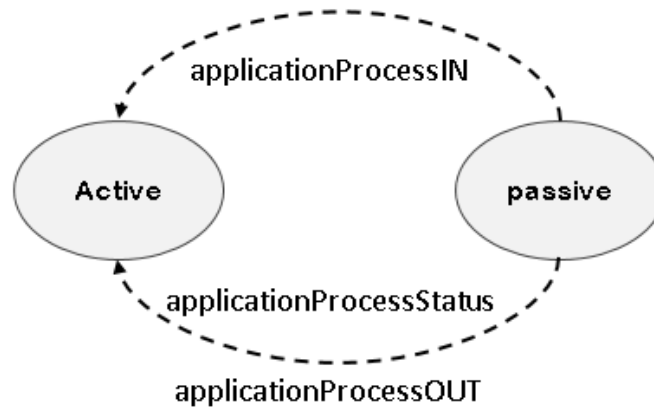
$\lambda (“active2”, applicationTime) = applicationNum$

$ta (“active1”, applicationTime) = 5$

$ta (“active2”, applicationTime) = 5$

$ta (“passive”) = \infty$

## applicationProcess



$X = \{applicationProcessIn\}$

$Y = \{applicationProcessOUT, applicationProcessStatus\}$

$S = \{"passive", "active"\}$

$\delta_{ext} ("passive", applicationNum) = "active"$

$\delta_{int} ("active") = "passive"$

$\lambda ("active", applicationProcessTime)$

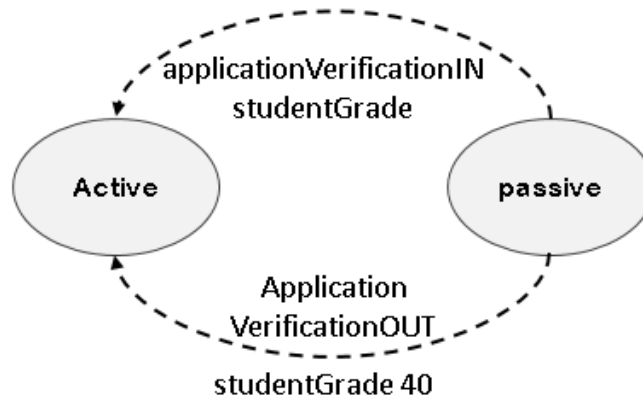
$= applicationNum$  if  $outport = applicationProcessStatus$

$= applicationProcessno$  if  $outport = applicationProcessOUT$

$ta ("active", applicationProcessTime) = 5$



registrar



$S = \{\text{"passive"}, \text{"active"}\}$

$X = \{\text{applicationVerificationIN}\}$

$Y = \{\text{applicationVerificationOUT}, \text{minStandardsNotMet}\}$

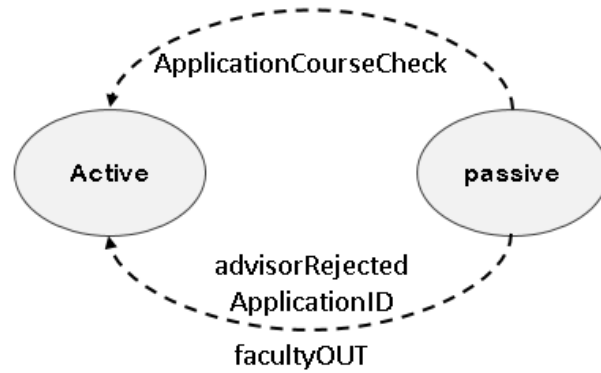
$\delta_{\text{ext}}(\text{"passive"}, \text{studentGrade}) = \text{"active"}$

$\delta_{\text{int}}(\text{"active"}) = \text{"passive"}$

$\lambda(\text{"active"}, \text{registrarTime}) = \text{studentGrade} (\geq 40) \text{ if } \text{outport} = \text{applicationVerificationOUT}$   
 $= \text{studentGrade} (< 40) \text{ if } \text{outport} = \text{minStandardNotMet}$

$ta(\text{"active"}, \text{registrarTime}) = 5$

## Faculty



$S = \{\text{"passive"}, \text{"active"}\}$

$X = \{\text{applicationCourseCheck}\}$

$Y = \{\text{facultyOut}, \text{advisorOpinionOUT}\}$

$\delta_{\text{ext}}(\text{"passive"}, \text{checkApplicationNum}) = \text{"active"}$

$\delta_{\text{int}}(\text{"active"}) = \text{"passive"}$

$\lambda(\text{"active"}, \text{facultyTime}) = \text{checkApplicationNum (satisfies) if outport = facultyOut}$

$\quad = \text{checkApplicationNum (not) if outport = advisorOpinionOUT}$

$ta(\text{"active"}, \text{facultyTime}) = 10$

### **Formal specifications of coupled models**

$N = (X, Y, D, EIC, EOC, IC)$  for the coupled model webApp and university Simulator are defined as follows:

- **webApp**

$X = \{webAppIN\}$

$Y = \{webAppOUT\}$

$D = \{application, applicationProcess\}$

$EIC = \{((webApp, "webAppIN"), (application, "applicationIN"))\}$

$EOC = \{(applicationProcess, "applicationProcessOUT"), (webApp, "webAppOUT")\}$

$IC = \{((application, "applicationOUT"), (applicationProcess, "applicationProcessIN")), ((applicationProcess, "applicationProcessStatus"), (application, "applicationFillingComplete"))\}$

- **university**

$X = \{studentApplicationIN\}$

$Y = \{rejectedApplicationID, advisorRejectedApplicationID\}$

$D = \{registrar, faculty\}$

$EIC = \{(university, "studentApplicationIN"), (registrar, "applicationVerificationIN")\}$

$EOC = \{((registrar, "minStandardsNotMet"), (university, "rejectedapplicationID")), ((faculty, "advisorOpinionOUT"), (university, "advisorRejectedApplicationID"))\}$

$IC = \{((registrar, "minStandardsNotMet"), (faculty, "applicationCourseCheck"))\}$

### **TOP MODEL : SARS**

$X = \{studentsRegistering\}$

$Y = \{facultyRejected, standardsNotMet\}$

$D = \{WEBAPP, UNIVERSITY\}$

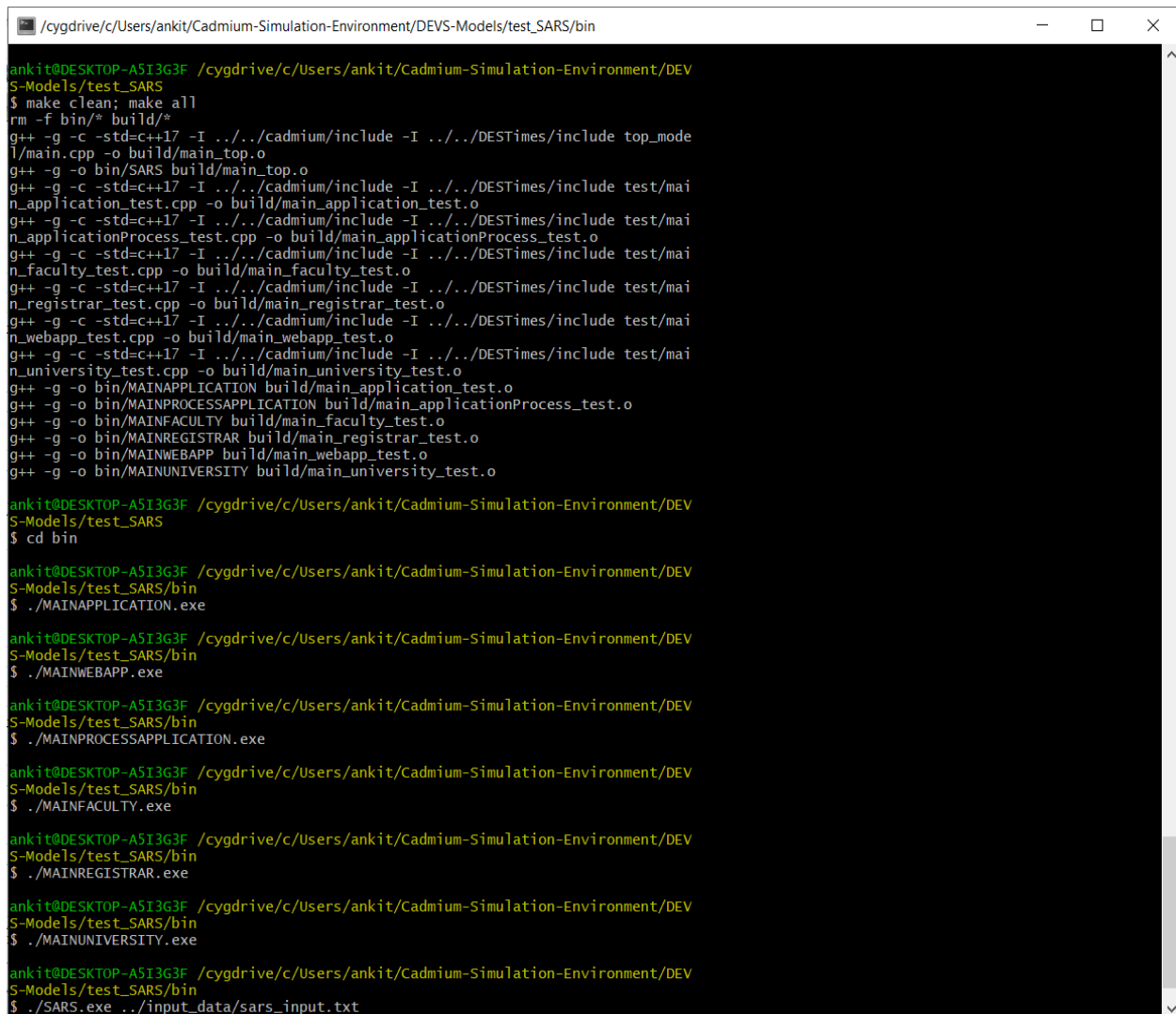
$EIC = \{(SARS, "studentsRegistering"), (WEBAPP, "webAppIN")\}$

EOC = {((UNIVERSITY, "advisorRejectedApplicationID"), (SARS, "facultyRejected")),  
 ((UNIVERSITY, "rejectedApplicationID"), (SARS, "standardsNotMet"))}  
 IC = {((WEBAPP, "webAppOUT"), (UNIVERSITY, "studentApplicationIN"))}

## Test Strategies:

The atomic models and coupled models will be tested using the "black box" testing method. Test cases are created by adding different combinations of inputs from the text files, run the simulation (./MAIN\_MODEL.exe) and check whether the outputs in the output file (.text) are what we expected.

Following is a screenshot of steps to compile and execute the given model in Cadmium.



```

/cygdrive/c/Users/ankit/Cadmium-Simulation-Environment/DEVS-Models/test_SARS/bin
ankit@DESKTOP-A5I3G3F /cygdrive/c/Users/ankit/Cadmium-Simulation-Environment/DEV
S-Models/test_SARS
$ make clean; make all
rm -f bin/* build/*
g++ -g -c -std=c++17 -I .././cadmium/include -I .././DESTimes/include top_mode
l/main.cpp -o build/main_top.o
g++ -g -o bin/SARS build/main_top.o
g++ -g -c -std=c++17 -I .././cadmium/include -I .././DESTimes/include test/mai
n_application_test.cpp -o build/main_application_test.o
g++ -g -c -std=c++17 -I .././cadmium/include -I .././DESTimes/include test/mai
n_applicationProcess_test.cpp -o build/main_applicationProcess_test.o
g++ -g -c -std=c++17 -I .././cadmium/include -I .././DESTimes/include test/mai
n_faculty_test.cpp -o build/main_faculty_test.o
g++ -g -c -std=c++17 -I .././cadmium/include -I .././DESTimes/include test/mai
n_registrar_test.cpp -o build/main_registrar_test.o
g++ -g -c -std=c++17 -I .././cadmium/include -I .././DESTimes/include test/mai
n_webapp_test.cpp -o build/main_webapp_test.o
g++ -g -c -std=c++17 -I .././cadmium/include -I .././DESTimes/include test/mai
n_university_test.cpp -o build/main_university_test.o
g++ -g -o bin/MAINAPPLICATION build/main_application_test.o
g++ -g -o bin/MAINPROCESSAPPLICATION build/main_applicationProcess_test.o
g++ -g -o bin/MAINFACULTY build/main_faculty_test.o
g++ -g -o bin/MAINREGISTRAR build/main_registrar_test.o
g++ -g -o bin/MAINWEBAPP build/main_webapp_test.o
g++ -g -o bin/MAINUNIVERSITY build/main_university_test.o

ankit@DESKTOP-A5I3G3F /cygdrive/c/Users/ankit/Cadmium-Simulation-Environment/DEV
S-Models/test_SARS
$ cd bin

ankit@DESKTOP-A5I3G3F /cygdrive/c/Users/ankit/Cadmium-Simulation-Environment/DEV
S-Models/test_SARS/bin
$ ./MAINAPPLICATION.exe

ankit@DESKTOP-A5I3G3F /cygdrive/c/Users/ankit/Cadmium-Simulation-Environment/DEV
S-Models/test_SARS/bin
$ ./MAINWEBAPP.exe

ankit@DESKTOP-A5I3G3F /cygdrive/c/Users/ankit/Cadmium-Simulation-Environment/DEV
S-Models/test_SARS/bin
$ ./MAINPROCESSAPPLICATION.exe

ankit@DESKTOP-A5I3G3F /cygdrive/c/Users/ankit/Cadmium-Simulation-Environment/DEV
S-Models/test_SARS/bin
$ ./MAINFACULTY.exe

ankit@DESKTOP-A5I3G3F /cygdrive/c/Users/ankit/Cadmium-Simulation-Environment/DEV
S-Models/test_SARS/bin
$ ./MAINREGISTRAR.exe

ankit@DESKTOP-A5I3G3F /cygdrive/c/Users/ankit/Cadmium-Simulation-Environment/DEV
S-Models/test_SARS/bin
$ ./MAINUNIVERSITY.exe

ankit@DESKTOP-A5I3G3F /cygdrive/c/Users/ankit/Cadmium-Simulation-Environment/DEV
S-Models/test_SARS/bin
$ ./SARS.exe ./input_data/sars_input.txt

```

### **PART 3:**

To verify the atomic models, test cases are created to test these models. Please refer the project models for more in-depth analysis on the coding.

#### **Atomic model “application”**

The input of application should be a positive integer represents the number of applications who will get on the university admission. No matter how large this integer is, as long as it is a positive integer, it should start to send from 1 indicates the first application. After sending, it should wait for the feedback at the port “applicationFillingComplete”. Once it receives the feedback and then it could send the next application (2, 3, 4 ...)

So, the input files for this atomic model is as follows, because we have to inputs into this application atomic model, we have 2 input files:

**File 1:** application\_input\_applicationIN .text

00:00:10:00 11

**File 2:** application\_input\_applicationFillingComplete .text

00:00:16:00 1

00:00:22:00 2

**00:00:25:00 3**

00:00:32:00 4

00:00:40:00 5

**00:00:42:00 6**

00:00:52:00 7

00:00:58:00 8

00:01:04:00 9

00:01:10:00 10

00:01:16:00 11

The 2 bold events (line 3 and 8) should not generate outputs because the “applicationTime” is 5 which means it needs 5 seconds to execute the application time. If at that time, the feedback is arrived in advance, buffer will ignore it at current and handles the new one. Take the line 5 as

an example, at 32 second; the input receives 4 but 3 has not finished, so buffer ignores the on-going one and handles 4. One should be noticed is that the elapsed time is changed from 4 to 0, so the output is 5 at 37 second, same as that of cd++ model testing. The output file shows the expected result as follows:

## 1. application\_test\_output\_messages.txt

```
File Edit Format View Help
00:00:15:000
[Application_defs::applicationOUT: {1}] generated by model application1
00:00:16:000
[cdmium::basic_models::pdevs::iestream_input_defs<int>::out: {1}] generated by model input_reader_applicationFillingComplete
00:00:16:000
[Application_defs::applicationOUT: {1}] generated by model application1
00:00:21:000
[Application_defs::applicationOUT: {2}] generated by model application1
00:00:22:000
[cdmium::basic_models::pdevs::iestream_input_defs<int>::out: {2}] generated by model input_reader_applicationFillingComplete
00:00:22:000
[Application_defs::applicationOUT: {2}] generated by model application1
00:00:25:000
[cdmium::basic_models::pdevs::iestream_input_defs<int>::out: {3}] generated by model input_reader_applicationFillingComplete
00:00:25:000
[Application_defs::applicationOUT: {3}] generated by model application1
00:00:30:000
[Application_defs::applicationOUT: {4}] generated by model application1
00:00:32:000
[cdmium::basic_models::pdevs::iestream_input_defs<int>::out: {4}] generated by model input_reader_applicationFillingComplete
00:00:32:000
[Application_defs::applicationOUT: {4}] generated by model application1
00:00:37:000
[Application_defs::applicationOUT: {5}] generated by model application1
00:00:40:000
[cdmium::basic_models::pdevs::iestream_input_defs<int>::out: {5}] generated by model input_reader_applicationFillingComplete
00:00:40:000
[Application_defs::applicationOUT: {5}] generated by model application1
00:00:42:000
[cdmium::basic_models::pdevs::iestream_input_defs<int>::out: {6}] generated by model input_reader_applicationFillingComplete
00:00:42:000
[Application_defs::applicationOUT: {6}] generated by model application1
00:00:47:000
[Application_defs::applicationOUT: {7}] generated by model application1
```

## 2. application\_test\_output\_state.txt

---

```

State for model input_reader_applicationIN is next time: inf
State for model input_reader_applicationFillingComplete is next time: 00:00:16:000
State for model application1 is applicationNum: 1 & totalApplicationNum: 11
00:00:15:000
State for model input_reader_applicationIN is next time: inf
State for model input_reader_applicationFillingComplete is next time: 00:00:16:000
State for model application1 is applicationNum: 1 & totalApplicationNum: 11
00:00:16:000
State for model input_reader_applicationIN is next time: inf
State for model input_reader_applicationFillingComplete is next time: 00:00:06:000
State for model application1 is applicationNum: 1 & totalApplicationNum: 11
00:00:16:000
State for model input_reader_applicationIN is next time: inf
State for model input_reader_applicationFillingComplete is next time: 00:00:06:000
State for model application1 is applicationNum: 2 & totalApplicationNum: 11
00:00:21:000
State for model input_reader_applicationIN is next time: inf
State for model input_reader_applicationFillingComplete is next time: 00:00:06:000
State for model application1 is applicationNum: 2 & totalApplicationNum: 11
00:00:22:000
State for model input_reader_applicationIN is next time: inf
State for model input_reader_applicationFillingComplete is next time: 00:00:03:000
State for model application1 is applicationNum: 2 & totalApplicationNum: 11
00:00:22:000
State for model input_reader_applicationIN is next time: inf
State for model input_reader_applicationFillingComplete is next time: 00:00:03:000
State for model application1 is applicationNum: 3 & totalApplicationNum: 11
00:00:25:000
State for model input_reader_applicationIN is next time: inf
State for model input_reader_applicationFillingComplete is next time: 00:00:07:000
State for model application1 is applicationNum: 3 & totalApplicationNum: 11
00:00:25:000
State for model input_reader_applicationIN is next time: inf
State for model input_reader_applicationFillingComplete is next time: 00:00:07:000
State for model application1 is applicationNum: 4 & totalApplicationNum: 11
00:00:30:000
State for model input_reader_applicationIN is next time: inf
State for model input_reader_applicationFillingComplete is next time: 00:00:07:000
State for model application1 is applicationNum: 4 & totalApplicationNum: 11
00:00:32:000
State for model input_reader_applicationIN is next time: inf
State for model input_reader_applicationFillingComplete is next time: 00:00:08:000

```

---

- **Atomic model “applicationProcess”**

The input of the applicationProcess should be an integer as well, and then it handles the data it received, making it BIT AND with 1. If the result is 1, it will send from the port “applicationProcessStatus” and the port “applicationProcessOUT”, otherwise it will send from the port “applicationProcessStatus” only.

So, the input file for this atomic model is as follows:

applicationProcess\_input.text

00:00:10:00 1  
**00:00:16:00 2**  
**00:00:19:00 6**  
00:00:28:00 7  
00:00:34:00 8  
00:00:35:00 16  
00:01:28:00 27  
00:02:34:00 38  
**00:02:36:00 39**  
00:04:34:00 43  
00:05:34:00 53  
00:06:20:00 65  
00:07:34:00 77  
00:08:00:00 88

The 3 bold events (line 3, 6 and 9) should not generate outputs because the “applicationProcessTime” is 5 which means it needs 5 seconds to execute the application. If at that time, a new input arrived in advance, application/buffer will ignore it because it is in the “active” mode at current. Take the line 3 as an example, at 19 seconds; the input receives 6 but 2 has not finished, so buffer ignores the 6 directly rather than handle it, which is opposite to the “application” model. One should be noticed is that the elapsed time does not change although the new data is coming, so the input 2 is sent at 21(03+09+09) second without changing. The output file shows the expected result as follows:



1.

applicationProcess\_test\_output\_messages - Notepad

File Edit Format View Help

00:00:00:000

00:00:00:000

[cadmium::basic\_models::pdevs::iestream\_input\_defs<int>::out: {}] generated by model input\_reader\_applicationProcessIN

00:00:10:000

[cadmium::basic\_models::pdevs::iestream\_input\_defs<int>::out: {1}] generated by model input\_reader\_applicationProcessIN

00:00:15:000

[ApplicationProcess\_defs::applicationProcessOUT: {1}, ApplicationProcess\_defs::applicationProcessStatus: {1}] generated by model application1

00:00:16:000

[cadmium::basic\_models::pdevs::iestream\_input\_defs<int>::out: {2}] generated by model input\_reader\_applicationProcessIN

00:00:19:000

[cadmium::basic\_models::pdevs::iestream\_input\_defs<int>::out: {6}] generated by model input\_reader\_applicationProcessIN

00:00:24:000

[ApplicationProcess\_defs::applicationProcessOUT: {2}, ApplicationProcess\_defs::applicationProcessStatus: {2}] generated by model application1

00:00:28:000

[cadmium::basic\_models::pdevs::iestream\_input\_defs<int>::out: {7}] generated by model input\_reader\_applicationProcessIN

00:00:33:000

[ApplicationProcess\_defs::applicationProcessOUT: {7}, ApplicationProcess\_defs::applicationProcessStatus: {7}] generated by model application1

00:00:34:000

[cadmium::basic\_models::pdevs::iestream\_input\_defs<int>::out: {8}] generated by model input\_reader\_applicationProcessIN

00:00:35:000

[cadmium::basic\_models::pdevs::iestream\_input\_defs<int>::out: {16}] generated by model input\_reader\_applicationProcessIN

00:00:40:000

[ApplicationProcess\_defs::applicationProcessOUT: {8}, ApplicationProcess\_defs::applicationProcessStatus: {8}] generated by model application1

00:00:45:000

[ApplicationProcess\_defs::applicationProcessOUT: {8}, ApplicationProcess\_defs::applicationProcessStatus: {8}] generated by model application1

00:00:50:000


[ApplicationProcess\_defs::applicationProcessOUT: {8}, ApplicationProcess\_defs::applicationProcessStatus: {8}] generated by model application1

00:00:55:000

[ApplicationProcess\_defs::applicationProcessOUT: {8}, ApplicationProcess\_defs::applicationProcessStatus: {8}] generated by model application1

00:01:00:000

2.

 applicationProcess\_test\_output\_state - Notepad

File Edit Format View Help

```
00:00:00:000
State for model input_reader_applicationProcessIN is next time: 00:00:00:000
State for model application1 is ApplicationProcessOUT: 0
00:00:00:000
State for model input_reader_applicationProcessIN is next time: 00:00:10:000
State for model application1 is ApplicationProcessOUT: 0
00:00:10:000
State for model input_reader_applicationProcessIN is next time: 00:00:06:000
State for model application1 is ApplicationProcessOUT: 1
00:00:15:000
State for model input_reader_applicationProcessIN is next time: 00:00:06:000
State for model application1 is ApplicationProcessOUT: 1
00:00:16:000
State for model input_reader_applicationProcessIN is next time: 00:00:03:000
State for model application1 is ApplicationProcessOUT: 2
00:00:19:000
State for model input_reader_applicationProcessIN is next time: 00:00:09:000
State for model application1 is ApplicationProcessOUT: 2
00:00:24:000
State for model input_reader_applicationProcessIN is next time: 00:00:09:000
State for model application1 is ApplicationProcessOUT: 2
00:00:28:000
State for model input_reader_applicationProcessIN is next time: 00:00:06:000
State for model application1 is ApplicationProcessOUT: 7
00:00:33:000
State for model input_reader_applicationProcessIN is next time: 00:00:06:000
State for model application1 is ApplicationProcessOUT: 7
00:00:34:000
State for model input_reader_applicationProcessIN is next time: 00:00:01:000
State for model application1 is ApplicationProcessOUT: 8
00:00:35:000
State for model input_reader_applicationProcessIN is next time: 00:00:53:000
State for model application1 is ApplicationProcessOUT: 8
00:00:40:000
State for model input_reader_applicationProcessIN is next time: 00:00:53:000
State for model application1 is ApplicationProcessOUT: 8
00:00:45:000
State for model input_reader_applicationProcessIN is next time: 00:00:53:000
```

- **Atomic model “Registrar”**

The input of this model is integers indicates the student applications. Then it handles the data it received, making it BIT AND with 30 or 40. If the result is larger than 25, it will send from the port “minStandardsNotMet”, otherwise it will be sent from the port “applicationVerificationOUT” only.

So, the test file “registrar\_input.text” is created as follows:

00:00:00:00 22

00:00:20:00 23

00:00:25:00 24

00:01:10:00 26

**00:02:10:00 56**

00:02:20:00 122

00:02:30:00 125

00:02:40:00 167

00:02:50:00 178

**00:03:00:00 179**

00:04:20:00 223

00:04:22:00 225

00:04:50:00 245

00:05:10:00 256

00:05:30:00 278

The output should generate different outputs depending on different integers. It is similar to the “applicationProcess” model when the new data is input but the former one does not be handled yet, so the new one will just be dropped, which makes sense in reality as well. The output files show the expected result as follows:


1.

Registrar\_test\_output\_messages - Notepad

File Edit Format View Help

```
00:00:00:000
00:00:00:000
[cadmium::basic_models::pdevs::iestream_input_defs<int>::out: {}] generated by model input_reader_registrar
00:00:00:000
[cadmium::basic_models::pdevs::iestream_input_defs<int>::out: {22}] generated by model input_reader_registrar
00:00:09:000
[Registrar_defs::minStandardsNotMet: {}, Registrar_defs::applicationVerificationOUT: {22}] generated by model registrar1
00:00:18:000
[Registrar_defs::minStandardsNotMet: {}, Registrar_defs::applicationVerificationOUT: {22}] generated by model registrar1
00:00:20:000
[cadmium::basic_models::pdevs::iestream_input_defs<int>::out: {23}] generated by model input_reader_registrar
00:00:25:000
[cadmium::basic_models::pdevs::iestream_input_defs<int>::out: {24}] generated by model input_reader_registrar
00:00:34:000
[Registrar_defs::minStandardsNotMet: {}, Registrar_defs::applicationVerificationOUT: {23}] generated by model registrar1
00:00:43:000
[Registrar_defs::minStandardsNotMet: {}, Registrar_defs::applicationVerificationOUT: {23}] generated by model registrar1
00:00:52:000
[Registrar_defs::minStandardsNotMet: {}, Registrar_defs::applicationVerificationOUT: {23}] generated by model registrar1
00:01:01:000
[Registrar_defs::minStandardsNotMet: {}, Registrar_defs::applicationVerificationOUT: {23}] generated by model registrar1
00:01:10:000
[cadmium::basic_models::pdevs::iestream_input_defs<int>::out: {26}] generated by model input_reader_registrar
[Registrar_defs::minStandardsNotMet: {}, Registrar_defs::applicationVerificationOUT: {23}] generated by model registrar1
00:01:19:000
[Registrar_defs::minStandardsNotMet: {26}, Registrar_defs::applicationVerificationOUT: {}] generated by model registrar1
00:01:28:000
[Registrar_defs::minStandardsNotMet: {26}, Registrar_defs::applicationVerificationOUT: {}] generated by model registrar1
00:01:37:000
[Registrar_defs::minStandardsNotMet: {26}, Registrar_defs::applicationVerificationOUT: {}] generated by model registrar1
00:01:46:000
[Registrar_defs::minStandardsNotMet: {26}, Registrar_defs::applicationVerificationOUT: {}] generated by model registrar1
00:01:55:000
[Registrar_defs::minStandardsNotMet: {26}, Registrar_defs::applicationVerificationOUT: {}] generated by model registrar1
00:02:04:000
[Registrar_defs::minStandardsNotMet: {26}, Registrar_defs::applicationVerificationOUT: {}] generated by model registrar1
00:02:10:000
```

2.

 Registrar\_test\_output\_state - Notepad

File Edit Format View Help

```
00:00:00:000
State for model input_reader_registrar is next time: 00:00:00:000
State for model registrar1 is studentGrade: 0
00:00:00:000
State for model input_reader_registrar is next time: 00:00:00:000
State for model registrar1 is studentGrade: 0
00:00:00:000
State for model input_reader_registrar is next time: 00:00:20:000
State for model registrar1 is studentGrade: 22
00:00:09:000
State for model input_reader_registrar is next time: 00:00:20:000
State for model registrar1 is studentGrade: 22
00:00:18:000
State for model input_reader_registrar is next time: 00:00:20:000
State for model registrar1 is studentGrade: 22
00:00:20:000
State for model input_reader_registrar is next time: 00:00:05:000
State for model registrar1 is studentGrade: 23
00:00:25:000
State for model input_reader_registrar is next time: 00:00:45:000
State for model registrar1 is studentGrade: 23
00:00:34:000
State for model input_reader_registrar is next time: 00:00:45:000
State for model registrar1 is studentGrade: 23
00:00:43:000
State for model input_reader_registrar is next time: 00:00:45:000
State for model registrar1 is studentGrade: 23
00:00:52:000
State for model input_reader_registrar is next time: 00:00:45:000
State for model registrar1 is studentGrade: 23
00:01:01:000
State for model input_reader_registrar is next time: 00:00:45:000
State for model registrar1 is studentGrade: 23
00:01:10:000
State for model input_reader_registrar is next time: 00:01:00:000
State for model registrar1 is studentGrade: 26
00:01:19:000
State for model input_reader_registrar is next time: 00:01:00:000
State for model registrar1 is studentGrade: 26
00:01:28:000
```

- **Atomic model “faculty”**

This model has the similar function to the “registrar”. The input of this model is integers indicates the student applications. Then it handles the data it received, making it BIT AND with 63. If the result is 49 which indicates the application does not satisfy the standard, but not all the unsatisfied applications are sent from the “advisorRejectedApplicationID” output, some of them will have 20% chance to escape the check implementing by random function, which reflects the true fact. Due to the low chance, so I put many integers which should be possibly sent from the “advisorOpinionOUT”.

So, the input file “faculty\_input.txt” is created as follows:

```
00:00:00:00 46
00:03:00:00 47
00:04:00:00 48
00:05:00:00 49
00:06:00:00 51
00:07:00:00 53
00:08:00:00 113
00:09:00:00 177
00:10:00:00 241
```

The result of BIT AND algorithm should decide which port the application should be sent from, so most of them (49, 113, 177, 241, 305, 369) should be sent from the port “advisorOpinionOUT”, while there is 20% possibility that they could be sent from the port “faultyOUT”. The output is not deterministic due to the random function. The following is an example of the output file. In this output file, the event with bold fonts means this application escapes the rigorous background checks if you have an amazing recommendation from some famous and well-known folks.


1.

faculty\_test\_output\_messages - Notepad

File Edit Format View Help

```
[Faculty_defs::facultyOUT: {}, Faculty_defs::advisorOpinionOUT: {49}] generated by model faculty1
00:05:50:000
[Faculty_defs::facultyOUT: {}, Faculty_defs::advisorOpinionOUT: {49}] generated by model faculty1
00:06:00:000
[cadmium::basic_models::pdevs::iestream_input_defs<int>::out: {51}] generated by model input_reader_faculty
[Faculty_defs::facultyOUT: {}, Faculty_defs::advisorOpinionOUT: {49}] generated by model faculty1
00:06:10:000
[Faculty_defs::facultyOUT: {51}, Faculty_defs::advisorOpinionOUT: {}] generated by model faculty1
00:06:20:000
[Faculty_defs::facultyOUT: {51}, Faculty_defs::advisorOpinionOUT: {}] generated by model faculty1
00:06:30:000
[Faculty_defs::facultyOUT: {51}, Faculty_defs::advisorOpinionOUT: {}] generated by model faculty1
00:06:40:000
[Faculty_defs::facultyOUT: {51}, Faculty_defs::advisorOpinionOUT: {}] generated by model faculty1
00:06:50:000
[Faculty_defs::facultyOUT: {51}, Faculty_defs::advisorOpinionOUT: {}] generated by model faculty1
00:07:00:000
[cadmium::basic_models::pdevs::iestream_input_defs<int>::out: {53}] generated by model input_reader_faculty
[Faculty_defs::facultyOUT: {51}, Faculty_defs::advisorOpinionOUT: {}] generated by model faculty1
00:07:10:000
[Faculty_defs::facultyOUT: {53}, Faculty_defs::advisorOpinionOUT: {}] generated by model faculty1
00:07:20:000
[Faculty_defs::facultyOUT: {53}, Faculty_defs::advisorOpinionOUT: {}] generated by model faculty1
00:07:30:000
[Faculty_defs::facultyOUT: {53}, Faculty_defs::advisorOpinionOUT: {}] generated by model faculty1
00:07:40:000
[Faculty_defs::facultyOUT: {53}, Faculty_defs::advisorOpinionOUT: {}] generated by model faculty1
00:07:50:000
[Faculty_defs::facultyOUT: {53}, Faculty_defs::advisorOpinionOUT: {}] generated by model faculty1
00:08:00:000
[cadmium::basic_models::pdevs::iestream_input_defs<int>::out: {113}] generated by model input_reader_faculty
[Faculty_defs::facultyOUT: {53}, Faculty_defs::advisorOpinionOUT: {}] generated by model faculty1
00:08:10:000
[Faculty_defs::facultyOUT: {}, Faculty_defs::advisorOpinionOUT: {113}] generated by model faculty1
00:08:20:000
[Faculty_defs::facultyOUT: {}, Faculty_defs::advisorOpinionOUT: {113}] generated by model faculty1
00:08:30:000
[Faculty_defs::facultyOUT: {}, Faculty_defs::advisorOpinionOUT: {113}] generated by model faculty1
00:08:40:000
```

2.

 faculty\_test\_output\_state - Notepad

File Edit Format View Help

00:03:20:000

State for model input\_reader\_faculty is next time: 00:01:00:000

State for model faculty1 is facultyOUT: 47

00:03:30:000

State for model input\_reader\_faculty is next time: 00:01:00:000

State for model faculty1 is facultyOUT: 47

00:03:40:000

State for model input\_reader\_faculty is next time: 00:01:00:000

State for model faculty1 is facultyOUT: 47

00:03:50:000

State for model input\_reader\_faculty is next time: 00:01:00:000

State for model faculty1 is facultyOUT: 47

00:04:00:000

State for model input\_reader\_faculty is next time: 00:01:00:000

State for model faculty1 is facultyOUT: 48

00:04:10:000

State for model input\_reader\_faculty is next time: 00:01:00:000

State for model faculty1 is facultyOUT: 48

00:04:20:000

State for model input\_reader\_faculty is next time: 00:01:00:000

State for model faculty1 is facultyOUT: 48

00:04:30:000

State for model input\_reader\_faculty is next time: 00:01:00:000

State for model faculty1 is facultyOUT: 48

00:04:40:000

State for model input\_reader\_faculty is next time: 00:01:00:000

State for model faculty1 is facultyOUT: 48

00:04:50:000

State for model input\_reader\_faculty is next time: 00:01:00:000

State for model faculty1 is facultyOUT: 48

00:05:00:000

State for model input\_reader\_faculty is next time: 00:01:00:000

State for model faculty1 is facultyOUT: 48

00:05:10:000

State for model input\_reader\_faculty is next time: 00:01:00:000

State for model faculty1 is facultyOUT: 48

00:05:20:000

State for model input\_reader\_faculty is next time: 00:01:00:000

State for model faculty1 is facultyOUT: 48

00:05:30:000

State for model input\_reader\_faculty is next time: 00:01:00:000

State for model faculty1 is facultyOUT: 48



- **Coupled Model “webApp”**

The coupled model webApp is the second top model which consists of “application” and “applicationProcess” atomic models. All applications go to the webApp model, and then go into the “application” which calculates how many universities will go boarding, after which they enter the “applicationprocess” notifies where student applications should go next ( the other coupled model “university”). At the same time, once the “applicationProcess” has handled one application, it will notify the application that it is empty now and next application could come on.

The input of this coupled model is “webappIN”, a positive integer indicating the number of applications who want to get on the university admission. The output is “webappOut” which indicates the number of applications who want to get admission and have successfully submitted the application with 70% probability.

The test file “webApp\_input.txt” is simple with only one line as follows. 15 applications want to get the admission.


00:00:00:00 15

The output should be a series of integer numbers which indicate student applications. As I simulate at the time 00:00:00:00, going through the applicationTime and applicationProcessTime for each application. the time of output should be at 00:00:10:00 for the first application. Sequentially, the second application ( the integer 2 ) implemented the probability of 70 % and then the result is 0/1 which means he has submitted the application or not, so this is not sent from the port “webAppOUT”. In this case he accepted, That is the reason why the second application is sent out at the 00:00:20:000 (after handling 2 applications) given by **ApplicationProcessOUT**. The following is the output file.

1.

```
webApp_output_messages - Notepad
File Edit Format View Help
00:00:00:000
00:00:00:000
[cadmium::basic_models::pdevs::iestream_input_defs<int>::out: {}] generated by model input_reader
00:00:00:000
[cadmium::basic_models::pdevs::iestream_input_defs<int>::out: {15}] generated by model input_reader
00:00:05:000
[Application_defs::applicationOUT: {1}] generated by model application1
00:00:10:000
[ApplicationProcess_defs::applicationProcessOUT: {1}, ApplicationProcess_defs::applicationProcessStatus: {1}] generated by model applicationProcess1
00:00:10:000
[Application_defs::applicationOUT: {1}] generated by model application1
00:00:15:000
[Application_defs::applicationOUT: {2}] generated by model application1
[ApplicationProcess_defs::applicationProcessOUT: {1}, ApplicationProcess_defs::applicationProcessStatus: {1}] generated by model applicationProcess1
00:00:15:000
[Application_defs::applicationOUT: {2}] generated by model application1
00:00:20:000
[Application_defs::applicationOUT: {3}] generated by model application1
[ApplicationProcess_defs::applicationProcessOUT: {2}, ApplicationProcess_defs::applicationProcessStatus: {2}] generated by model applicationProcess1
00:00:20:000
[Application_defs::applicationOUT: {3}] generated by model application1
00:00:25:000
[Application_defs::applicationOUT: {4}] generated by model application1
[ApplicationProcess_defs::applicationProcessOUT: {3}, ApplicationProcess_defs::applicationProcessStatus: {3}] generated by model applicationProcess1
00:00:25:000
[Application_defs::applicationOUT: {4}] generated by model application1
00:00:30:000
[Application_defs::applicationOUT: {5}] generated by model application1
[ApplicationProcess_defs::applicationProcessOUT: {4}, ApplicationProcess_defs::applicationProcessStatus: {4}] generated by model applicationProcess1
00:00:30:000
[Application_defs::applicationOUT: {5}] generated by model application1
00:00:35:000
[Application_defs::applicationOUT: {6}] generated by model application1
[ApplicationProcess_defs::applicationProcessOUT: {5}, ApplicationProcess_defs::applicationProcessStatus: {5}] generated by model applicationProcess1
00:00:35:000
[Application_defs::applicationOUT: {6}] generated by model application1
00:00:40:000
[Application_defs::applicationOUT: {7}] generated by model application1
[ApplicationProcess_defs::applicationProcessOUT: {6}, ApplicationProcess_defs::applicationProcessStatus: {6}] generated by model applicationProcess1
```

2.

 webApp\_output\_state - Notepad

File Edit Format View Help

00:00:00:000

State for model input\_reader is next time: inf

State for model application1 is applicationNum: 1 & totalApplicationNum: 15

State for model applicationProcess1 is ApplicationProcessOUT: 1

00:00:05:000

State for model input\_reader is next time: inf

State for model application1 is applicationNum: 1 & totalApplicationNum: 15

State for model applicationProcess1 is ApplicationProcessOUT: 1

00:00:10:000

State for model input\_reader is next time: inf

State for model application1 is applicationNum: 1 & totalApplicationNum: 15

State for model applicationProcess1 is ApplicationProcessOUT: 1

00:00:10:000

State for model input\_reader is next time: inf

State for model application1 is applicationNum: 2 & totalApplicationNum: 15

State for model applicationProcess1 is ApplicationProcessOUT: 1

00:00:15:000

State for model input\_reader is next time: inf

State for model application1 is applicationNum: 2 & totalApplicationNum: 15

State for model applicationProcess1 is ApplicationProcessOUT: 2

00:00:15:000

State for model input\_reader is next time: inf

State for model application1 is applicationNum: 3 & totalApplicationNum: 15

State for model applicationProcess1 is ApplicationProcessOUT: 2

00:00:20:000

State for model input\_reader is next time: inf

State for model application1 is applicationNum: 3 & totalApplicationNum: 15

State for model applicationProcess1 is ApplicationProcessOUT: 3

00:00:20:000

State for model input\_reader is next time: inf

State for model application1 is applicationNum: 4 & totalApplicationNum: 15

State for model applicationProcess1 is ApplicationProcessOUT: 3

00:00:25:000

State for model input\_reader is next time: inf

State for model application1 is applicationNum: 4 & totalApplicationNum: 15

State for model applicationProcess1 is ApplicationProcessOUT: 4

00:00:25:000

State for model input\_reader is next time: inf

State for model application1 is applicationNum: 5 & totalApplicationNum: 15

State for model applicationProcess1 is ApplicationProcessOUT: 4

00:00:30:000

State for model input\_reader is next time: inf

- **Coupled Model “university”**

The coupled model university is also the second top model which consists of “registrar” and “Faculty” atomic models. All student applications enter into the university model, and then go into the “registrar” which calculates whether the applications should be accepted or not, after which they enter the “faculty” atomic model to receive the corresponding final application check. In any of three check programs, the application does not obey the rule of university standards and sheer luck; the application will probably fail to secure a seat in that university. But some of them has the certain probability to escape such things, if they are lucky enough.

The input of this coupled model is “studentapplicationIN”, a positive integer indicating the number of student applications who want to get an admission. The outputs are “rejected applicationID” and “advisorRejectedApplicationID” which indicate the number of applications who do not satisfy the conditions.

The test file “university\_input.txt” is not simple as the “webapp” coupled model as follows.

```
00:00:00:00 22
00:00:20:00 23
00:00:25:00 24
00:01:10:00 26
00:02:10:00 56
00:02:20:00 122
00:02:30:00 125
00:02:40:00 167
00:02:50:00 178
00:03:00:00 179
00:04:20:00 223
00:04:22:00 225
00:04:50:00 245
00:05:10:00 256
00:05:30:00 278
```


1.

university\_output\_messages - Notepad

File Edit Format View Help

```
00:00:20:000
[cadmium::basic_models::pdevs::iestream_input_defs<int>::out: {23}] generated by model input_reader
00:00:25:000
[cadmium::basic_models::pdevs::iestream_input_defs<int>::out: {24}] generated by model input_reader
00:00:28:000
[Faculty_defs::facultyOUT: {22}, Faculty_defs::advisorOpinionOUT: {}] generated by model faculty1
00:00:34:000
[Registrar_defs::minStandardsNotMet: {}, Registrar_defs::applicationVerificationOUT: {23}] generated by model registrar1
00:00:43:000
[Registrar_defs::minStandardsNotMet: {}, Registrar_defs::applicationVerificationOUT: {23}] generated by model registrar1
00:00:52:000
[Registrar_defs::minStandardsNotMet: {}, Registrar_defs::applicationVerificationOUT: {23}] generated by model registrar1
00:01:01:000
[Registrar_defs::minStandardsNotMet: {}, Registrar_defs::applicationVerificationOUT: {23}] generated by model registrar1
00:01:10:000
[cadmium::basic_models::pdevs::iestream_input_defs<int>::out: {26}] generated by model input_reader
[Registrar_defs::minStandardsNotMet: {}, Registrar_defs::applicationVerificationOUT: {23}] generated by model registrar1
00:01:19:000
[Registrar_defs::minStandardsNotMet: {26}, Registrar_defs::applicationVerificationOUT: {}] generated by model registrar1
00:01:20:000
[Faculty_defs::facultyOUT: {23}, Faculty_defs::advisorOpinionOUT: {}] generated by model faculty1
00:01:28:000
[Registrar_defs::minStandardsNotMet: {26}, Registrar_defs::applicationVerificationOUT: {}] generated by model registrar1
00:01:30:000
[Faculty_defs::facultyOUT: {23}, Faculty_defs::advisorOpinionOUT: {}] generated by model faculty1
00:01:37:000
[Registrar_defs::minStandardsNotMet: {26}, Registrar_defs::applicationVerificationOUT: {}] generated by model registrar1
00:01:40:000
[Faculty_defs::facultyOUT: {23}, Faculty_defs::advisorOpinionOUT: {}] generated by model faculty1
00:01:46:000
[Registrar_defs::minStandardsNotMet: {26}, Registrar_defs::applicationVerificationOUT: {}] generated by model registrar1
00:01:50:000
[Faculty_defs::facultyOUT: {23}, Faculty_defs::advisorOpinionOUT: {}] generated by model faculty1
00:01:55:000
[Registrar_defs::minStandardsNotMet: {26}, Registrar_defs::applicationVerificationOUT: {}] generated by model registrar1
00:02:00:000
[Faculty_defs::facultyOUT: {23}, Faculty_defs::advisorOpinionOUT: {}] generated by model faculty1
```

2.

 university\_output\_state - Notepad

File Edit Format View Help

```
State for model faculty1 is facultyOUT: 56
State for model registrar1 is studentGrade: 125
00:02:39:000
State for model input_reader is next time: 00:00:10:000
State for model faculty1 is facultyOUT: 56
State for model registrar1 is studentGrade: 125
00:02:40:000
State for model input_reader is next time: 00:00:10:000
State for model faculty1 is facultyOUT: 56
State for model registrar1 is studentGrade: 167
00:02:49:000
State for model input_reader is next time: 00:00:10:000
State for model faculty1 is facultyOUT: 39
State for model registrar1 is studentGrade: 167
00:02:50:000
State for model input_reader is next time: 00:00:10:000
State for model faculty1 is facultyOUT: 39
State for model registrar1 is studentGrade: 178
00:02:59:000
State for model input_reader is next time: 00:00:10:000
State for model faculty1 is facultyOUT: 50
State for model registrar1 is studentGrade: 178
00:03:00:000
State for model input_reader is next time: 00:01:20:000
State for model faculty1 is facultyOUT: 50
State for model registrar1 is studentGrade: 179
00:03:09:000
State for model input_reader is next time: 00:01:20:000
State for model faculty1 is facultyOUT: 51
State for model registrar1 is studentGrade: 179
00:03:18:000
State for model input_reader is next time: 00:01:20:000
State for model faculty1 is facultyOUT: 51
State for model registrar1 is studentGrade: 179
00:03:27:000
State for model input_reader is next time: 00:01:20:000
State for model faculty1 is facultyOUT: 51
State for model registrar1 is studentGrade: 179
00:03:36:000
State for model input_reader is next time: 00:01:20:000
State for model faculty1 is facultyOUT: 51
State for model registrar1 is studentGrade: 179
```

<

- The top model “SARS”

The top model consists of 2 coupled models “webApp” and “university”. The input of the top model should be an integer (larger than 0) which indicates the total number of applications. The output should be student applications who fail in any of two check process. In order to get a larger samples to analyze, I set the “studentsRegistering” equal to 500. So the file sars\_input.txt is created as follows:


00:00:00:00 500

After going through the “webapp” model, total applications are divided into 2 parts- submitted and not submitted applications. student Applications are represented by the numbers. Then they receive a series of checks implemented by the algorithm BIT AND, and then some Applications will fail in one of two check process, but some will escape due to the random function

- 1.

```
SARS_output_messages - Notepad
File Edit Format View Help
00:00:00:000
00:00:00:000
[cadmium::basic_models::pdevs::istream_input_defs<int>::out: {}] generated by model input_reader
00:00:00:000
[cadmium::basic_models::pdevs::istream_input_defs<int>::out: {500}] generated by model input_reader
00:00:05:000
[Application_defs::applicationOUT: {1}] generated by model application1
00:00:10:000
[ApplicationProcess_defs::applicationProcessOUT: {1}, ApplicationProcess_defs::applicationProcessStatus: {1}] generated by model applicationProcess1
00:00:10:000
[Application_defs::applicationOUT: {1}] generated by model application1
00:00:15:000
[Application_defs::applicationOUT: {2}] generated by model application1
[ApplicationProcess_defs::applicationProcessOUT: {1}, ApplicationProcess_defs::applicationProcessStatus: {1}] generated by model applicationProcess1
00:00:15:000
[Application_defs::applicationOUT: {2}] generated by model application1
00:00:20:000
[Application_defs::applicationOUT: {3}] generated by model application1
[ApplicationProcess_defs::applicationProcessOUT: {2}, ApplicationProcess_defs::applicationProcessStatus: {2}] generated by model applicationProcess1
00:00:20:000
[Application_defs::applicationOUT: {3}] generated by model application1
00:00:25:000
[Application_defs::applicationOUT: {4}] generated by model application1
[ApplicationProcess_defs::applicationProcessOUT: {3}, ApplicationProcess_defs::applicationProcessStatus: {3}] generated by model applicationProcess1
00:00:25:000
[Application_defs::applicationOUT: {4}] generated by model application1
00:00:30:000
[Application_defs::applicationOUT: {5}] generated by model application1
[ApplicationProcess_defs::applicationProcessOUT: {4}, ApplicationProcess_defs::applicationProcessStatus: {4}] generated by model applicationProcess1
00:00:30:000
[Application_defs::applicationOUT: {5}] generated by model application1
00:00:35:000
[Application_defs::applicationOUT: {6}] generated by model application1
[ApplicationProcess_defs::applicationProcessOUT: {5}, ApplicationProcess_defs::applicationProcessStatus: {5}] generated by model applicationProcess1
00:00:35:000
[Application_defs::applicationOUT: {6}] generated by model application1
```

2.

 SARS\_output\_state - Notepad

File Edit Format View Help

00:00:45:000

State for model input\_reader is next time: inf

State for model application1 is applicationNum: 9 & totalApplicationNum: 500

State for model applicationProcess1 is ApplicationProcessOUT: 8

State for model faculty1 is facultyOUT: 0

State for model registrar1 is studentGrade: 1

00:00:50:000

State for model input\_reader is next time: inf

State for model application1 is applicationNum: 9 & totalApplicationNum: 500

State for model applicationProcess1 is ApplicationProcessOUT: 9

State for model faculty1 is facultyOUT: 0

State for model registrar1 is studentGrade: 1

00:00:50:000

State for model input\_reader is next time: inf

State for model application1 is applicationNum: 10 & totalApplicationNum: 500

State for model applicationProcess1 is ApplicationProcessOUT: 9

State for model faculty1 is facultyOUT: 0

State for model registrar1 is studentGrade: 1

00:00:55:000

State for model input\_reader is next time: inf

State for model application1 is applicationNum: 10 & totalApplicationNum: 500

State for model applicationProcess1 is ApplicationProcessOUT: 10

State for model faculty1 is facultyOUT: 0

State for model registrar1 is studentGrade: 1

00:00:55:000

State for model input\_reader is next time: inf

State for model application1 is applicationNum: 11 & totalApplicationNum: 500

State for model applicationProcess1 is ApplicationProcessOUT: 10

State for model faculty1 is facultyOUT: 0

State for model registrar1 is studentGrade: 1

00:01:00:000

State for model input\_reader is next time: inf

State for model application1 is applicationNum: 11 & totalApplicationNum: 500

State for model applicationProcess1 is ApplicationProcessOUT: 11

State for model faculty1 is facultyOUT: 0

State for model registrar1 is studentGrade: 1

00:01:00:000

State for model input\_reader is next time: inf

State for model application1 is applicationNum: 12 & totalApplicationNum: 500

State for model applicationProcess1 is ApplicationProcessOUT: 11

State for model faculty1 is facultyOUT: 0

State for model registrar1 is studentGrade: 1



**Conclusion:**

The SARS model simulates the process of student applications. In addition, the hierarchical test for both atomic models and coupled models generate the expected results. The output data is reasonable, and the time of output can be calculated according to the time set in each model. The information of registrar and faculty advisors of applications can be obtained by applicationID number (in binary) in my model. The execution results and the behavior of the model match the specifications of models.