System Dependability Lab

Exercises on Safety Assessment of Dynamic Systems

ENNAOUI Hamza(L2), DAI Guohao(L1)

Thursday 1st December, 2022

Question 1: Basic modeling components

1. Complete the class **RepairableComponent** :

```
class RepairableComponent
	// inherits from nonrepairableComponent
	extends NonRepairableComponent;
	// Adding the repair event to make the component repairable
	event evRepair (delay = exponential(pMeu));
	parameter Real pMeu = 0.025;
	transition
		evRepair: not vsWorking -> vsWorking := true;
end
```

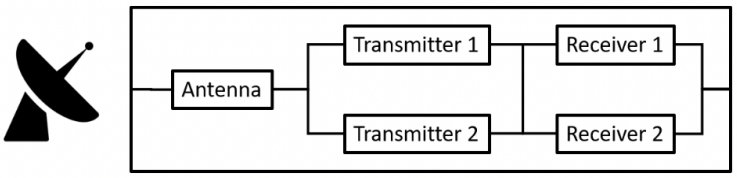
2. Complete the class **NonRepairableInOutComponent**:

```
class NonRepairableInOutComponent
	extends NonRepairableComponent;
	// Input and Output booleans
	Boolean vfInput, vfOutput (reset = false);
	assertion
		vfOutput := if vsWorking then vfInput else false;
end
```

3. Complete the class **RepairableInOutComponent** :

```
class RepairableInOutComponent
	extends RepairableComponent;
	// Input and Output booleans
	Boolean vfInput, vfOutput (reset = false);
	assertion
		vfOutput := if vsWorking then vfInput else false;
end
```

Question 2.1 : Reliability Block Diagrams



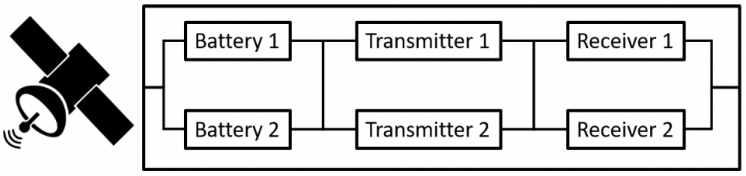
In this section, we complete and verify the class **GroundStationSubSystem** to represent the reliability block diagram given by figure.

The model is shown as below:

```
/* Radar subsystem
*  represented by a block diagram modeling pattern with repairable components
*/

class GroundStationSubSystem
	// Parameters

	// Components
	RepairableInOutComponent Antenna;
	RepairableInOutComponent Transmitter1;
	RepairableInOutComponent Transmitter2;
	RepairableInOutComponent Receiver1;
	RepairableInOutComponent Receiver2;
	// output boolean
	Boolean vfOutput ( reset = false );

	// Connections
	assertion
		Antenna.vfInput := true;
		// both transmitters take their input from the antenna
		Transmitter1.vfInput := Antenna.vfOutput;
		Transmitter2.vfInput := Antenna.vfOutput;
		// receivers take their input either from transmitter 1 or 2 due to
parallelisation
		Receiver1.vfInput := Transmitter1.vfOutput or Transmitter2.vfOutput;
```

```
            Receiver2.vfInput := Transmitter1.vfOutput or Transmitter2.vfOutput;
            // the final output is either from receiver1 or 2
            vfOutput := Receiver1.vfOutput or Receiver2.vfOutput;
end
```

Question 2.2 : Satellite reliability block diagram



In this section, we complete and verify the class **SatelliteSubSystem** to represent the reliability block diagram given by figure.

The model is shown as below:

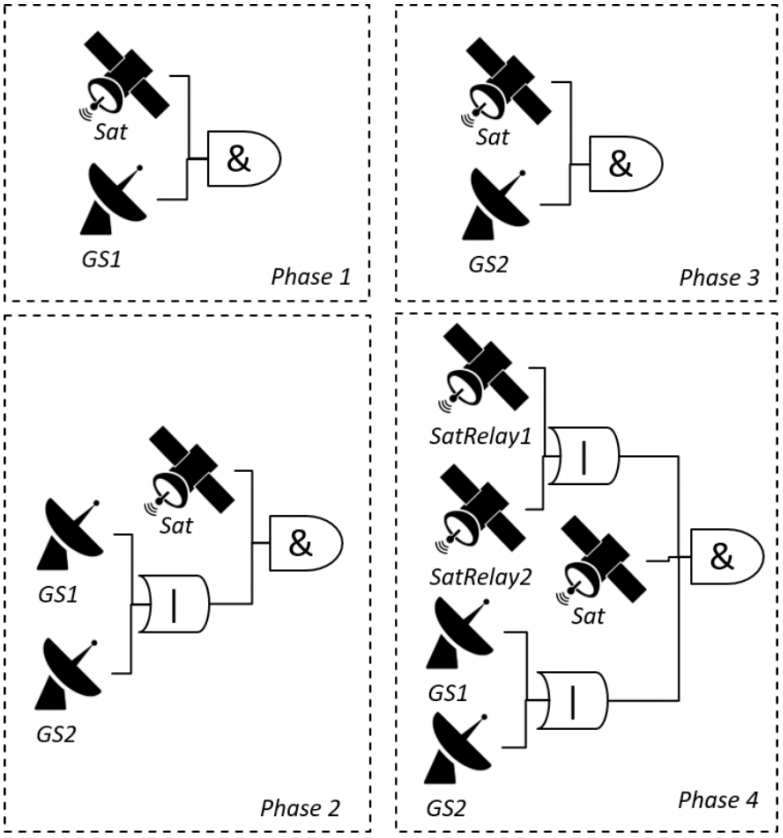
```
/*  Satellite subsystem
*     represented by a block diagram modeling pattern with non repairable
components
*/

class SatelliteSubSystem

      // Components
      NonRepairableInOutComponent Battery1;
      NonRepairableInOutComponent Battery2;
      NonRepairableInOutComponent Transmitter1;
      NonRepairableInOutComponent Transmitter2;
      NonRepairableInOutComponent Receiver1;
      NonRepairableInOutComponent Receiver2;

      Boolean vfOutput( reset = false );

      // Connections
      assertion
            Battery1.vfInput := true;
            Battery2.vfInput := true;
            // transmitters take their input either from battery1 or 2 due to
parallelisation
            Transmitter1.vfInput := Battery1.vfOutput or Battery2.vfOutput;
            Transmitter2.vfInput := Battery1.vfOutput or Battery2.vfOutput;

            // Receivers take their input either from transmitter1battery1 or 2
due to  parallelisation
            Receiver1.vfInput := Transmitter1.vfOutput or
Transmitter2.vfOutput;
```

```
            Receiver2.vfInput := Transmitter1.vfOutput or
Transmitter2.vfOutput;
            vfOutput := Receiver1.vfOutput or Receiver2.vfOutput;
end
```

Question 3: Static phased mission system modeling and assessment

In this section, we define the class **SatelliteSubSystem** which are subsystems of the satellite communication system used in different phases.

The following figure shows the components of the system used in each phase.



The model is shown as below:

```
/*
* Phased mission system:
*       Satellite Communication System
*/

block System
       // Radar subsystems
       GroundStationSubSystem GS1, GS2;

       // Relay satellite subsystems
```

```
	SatelliteSubSystem Sat;
	SatelliteSubSystem SatRelay1, SatRelay2;

	Boolean vfWorking (reset = false);

	// Phases modeling
	block PhaseController
		Integer vsPhase (init = 1);
	end

	// Subsystem used during the 1st phase
	block Phase1
		embeds main.GS1 as GS1;
		embeds main.Sat as Sat;
		Boolean vfWorking ( reset = false );
		assertion
			vfWorking := GS1.vfOutput and Sat.vfOutput;
	end

	// Subsystem used during the 2nd phase
	block Phase2
		embeds main.GS1 as GS1;
		embeds main.GS2 as GS2;
		embeds main.Sat as Sat;
		Boolean vfWorking ( reset = false );
		assertion
			vfWorking := (GS1.vfOutput or GS2.vfOutput) and
Sat.vfOutput;
	end

	// Subsystem used during the 3rd phase
	block Phase3
		embeds main.GS2 as GS2;
		embeds main.Sat as Sat;
		Boolean vfWorking( reset = false );
		assertion
			vfWorking := Sat.vfOutput and GS2.vfOutput;
	end

	// Subsystem used during the 4th phase
	block Phase4
		embeds main.GS1 as GS1;
		embeds main.GS2 as GS2;
		embeds main.SatRelay1 as SatRelay1;
		embeds main.SatRelay2 as SatRelay2;
		embeds main.Sat as Sat;
		Boolean vfWorking( reset = false );
		assertion
			vfWorking := (GS1.vfOutput or GS2.vfOutput) and Sat.vfOutput
and (SatRelay1.vfOutput or SatRelay2.vfOutput);
```

```
	end

	assertion
		vfWorking := if (PhaseController.vsPhase == 1) then
Phase1.vfWorking
				else if (PhaseController.vsPhase == 2) then
Phase2.vfWorking
				else if (PhaseController.vsPhase == 3) then
Phase3.vfWorking
				else if (PhaseController.vsPhase == 4) then
Phase4.vfWorking
				else false;
end
```

1. Compute Minimal Cuts Set for the Failure Condition (FC) for each phase (1,2,3,4) :

phase 1 :

1 GS1.Antenna.evFailure
2 GS1.Transmitter1.evFailure GS1.Transmitter2.evFailure
2 GS1.Receiver1.evFailure GS1.Receiver2.evFailure
2 Sat.Battery1.evFailure Sat.Battery2.evFailure
2 Sat.Transmitter1.evFailure Sat.Transmitter2.evFailure
2 Sat.Receiver1.evFailure Sat.Receiver2.evFailure

phase 2 :

2 GS1.Antenna.evFailure GS2.Antenna.evFailure
2 Sat.Battery1.evFailure Sat.Battery2.evFailure
2 Sat.Transmitter1.evFailure Sat.Transmitter2.evFailure
2 Sat.Receiver1.evFailure Sat.Receiver2.evFailure
3 GS1.Antenna.evFailure GS2.Transmitter1.evFailure GS2.Transmitter2.evFailure
3 GS1.Antenna.evFailure GS2.Receiver1.evFailure GS2.Receiver2.evFailure
3 GS1.Transmitter1.evFailure GS1.Transmitter2.evFailure GS2.Antenna.evFailure
3 GS1.Receiver1.evFailure GS1.Receiver2.evFailure GS2.Antenna.evFailure
4 GS1.Transmitter1.evFailure GS1.Transmitter2.evFailure GS2.Transmitter1.evFailure
GS2.Transmitter2.evFailure
4 GS1.Transmitter1.evFailure GS1.Transmitter2.evFailure GS2.Receiver1.evFailure
GS2.Receiver2.evFailure
4 GS1.Receiver1.evFailure GS1.Receiver2.evFailure GS2.Transmitter1.evFailure
GS2.Transmitter2.evFailure
4 GS1.Receiver1.evFailure GS1.Receiver2.evFailure GS2.Receiver1.evFailure
GS2.Receiver2.evFailure

phase 3 :

1 GS2.Antenna.evFailure
2 Sat.Battery1.evFailure Sat.Battery2.evFailure
2 Sat.Transmitter1.evFailure Sat.Transmitter2.evFailure
2 Sat.Receiver1.evFailure Sat.Receiver2.evFailure
2 GS2.Transmitter1.evFailure GS2.Transmitter2.evFailure
2 GS2.Receiver1.evFailure GS2.Receiver2.evFailure

phase 4 :

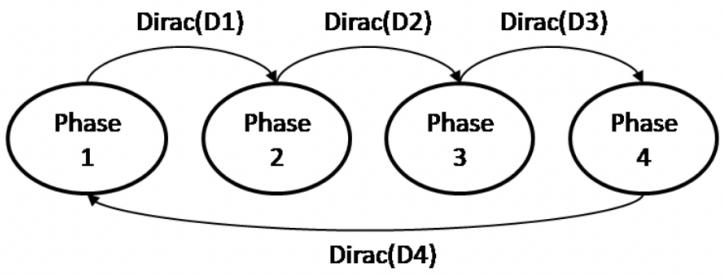
```
2  GS1.Antenna.evFailure  GS2.Antenna.evFailure
2  Sat.Battery1.evFailure  Sat.Battery2.evFailure
2  Sat.Transmitter1.evFailure  Sat.Transmitter2.evFailure
2  Sat.Receiver1.evFailure  Sat.Receiver2.evFailure
3  GS1.Antenna.evFailure  GS2.Transmitter1.evFailure  GS2.Transmitter2.evFailure
3  GS1.Antenna.evFailure  GS2.Receiver1.evFailure  GS2.Receiver2.evFailure
3  GS1.Transmitter1.evFailure  GS1.Transmitter2.evFailure  GS2.Antenna.evFailure
3  GS1.Receiver1.evFailure  GS1.Receiver2.evFailure  GS2.Antenna.evFailure
4  GS1.Transmitter1.evFailure  GS1.Transmitter2.evFailure  GS2.Transmitter1.evFailure
GS2.Transmitter2.evFailure
4  GS1.Transmitter1.evFailure  GS1.Transmitter2.evFailure  GS2.Receiver1.evFailure
GS2.Receiver2.evFailure
4  GS1.Receiver1.evFailure  GS1.Receiver2.evFailure  GS2.Transmitter1.evFailure
GS2.Transmitter2.evFailure
4  GS1.Receiver1.evFailure  GS1.Receiver2.evFailure  GS2.Receiver1.evFailure
GS2.Receiver2.evFailure
4  SatRelay1.Battery1.evFailure  SatRelay1.Battery2.evFailure  SatRelay2.Battery1.evFailure
SatRelay2.Battery2.evFailure
4  SatRelay1.Battery1.evFailure  SatRelay1.Battery2.evFailure
SatRelay2.Transmitter1.evFailure  SatRelay2.Transmitter2.evFailure
4  SatRelay1.Battery1.evFailure  SatRelay1.Battery2.evFailure  SatRelay2.Receiver1.evFailure
SatRelay2.Receiver2.evFailure
4  SatRelay1.Transmitter1.evFailure  SatRelay1.Transmitter2.evFailure
SatRelay2.Battery1.evFailure  SatRelay2.Battery2.evFailure
4  SatRelay1.Transmitter1.evFailure  SatRelay1.Transmitter2.evFailure
SatRelay2.Transmitter1.evFailure  SatRelay2.Transmitter2.evFailure
4  SatRelay1.Transmitter1.evFailure  SatRelay1.Transmitter2.evFailure
SatRelay2.Receiver1.evFailure  SatRelay2.Receiver2.evFailure
4  SatRelay1.Receiver1.evFailure  SatRelay1.Receiver2.evFailure  SatRelay2.Battery1.evFailure
SatRelay2.Battery2.evFailure
4  SatRelay1.Receiver1.evFailure  SatRelay1.Receiver2.evFailure
SatRelay2.Transmitter1.evFailure  SatRelay2.Transmitter2.evFailure
4  SatRelay1.Receiver1.evFailure  SatRelay1.Receiver2.evFailure
SatRelay2.Receiver1.evFailure  SatRelay2.Receiver2.evFailure
```

2. What are the most critical components of the system?Justify your answer.

Answer :

The most critical components are the antennas because they compose a minimal cut set. So if the antenna alone breaks the whole system breaks.

Question 4: Dynamic phased mission system modeling and assessment



In this section, we change phases of the satellite communication system by the given durations in the table.

Phase duration	D1=D3=2h, D2=1h, D4=7h

The model is shown as below:

4.2

```
/* Radar subsystem
* represented by a block diagram modeling pattern with repairable components
*
*/
class GroundStationSubSystem


// Components
	// the antenna
	RepairableInOutComponent Antenna;
	// the transmitters
	RepairableInOutComponent Transmitter1;
	RepairableInOutComponent Transmitter2;
	// the receivers
	RepairableInOutComponent Receiver1;
	RepairableInOutComponent Receiver2;

	// the output
	Boolean vfOutput ( reset = false );

	// Connections
	assertion
		Antenna.vfInput := true;
		// transmitter take their input from antenna
		Transmitter1.vfInput := Antenna.vfOutput;
		Transmitter2.vfInput := Antenna.vfOutput;

		Receiver1.vfInput := Transmitter1.vfOutput or
Transmitter2.vfOutput;
		Receiver2.vfInput := Transmitter1.vfOutput or
Transmitter2.vfOutput;
		// vfOutput is just the output of one of the receivers
		vfOutput := Receiver1.vfOutput or Receiver2.vfOutput;
end
```

4.5

```
/*
* Phased mission system:
* Satellite Communication System
```

```
*
*/
block System
	// Radar subsystems
	GroundStationSubSystem GS1, GS2;
	SatelliteSubSystem Sat;

	// Relay satellite subsystems
	SatelliteSubSystem SatRelay1, SatRelay2;

	Boolean vfWorking (reset = false);

	// Phases modeling
	block PhaseController
		Integer vsPhase (init = 1);
		// parameters
		// Durations as reals
		parameter Real D1 = 2.0;
		parameter Real D2 = 1.0;
		parameter Real D3 = 2.0;
		parameter Real D4 = 7.0;

		// events
		// we declare here every event to transition from phase to phase
		event evPhase1_2 (delay = Dirac(D1));
		event evPhase2_3 (delay = Dirac(D2));
		event evPhase3_4 (delay = Dirac(D3));
		event evPhase4_1 (delay = Dirac(D4));


		transition
		// definition of every transition
			evPhase1_2: vsPhase == 1 -> vsPhase := 2;
			evPhase2_3: vsPhase == 2 -> vsPhase := 3;
			evPhase3_4: vsPhase == 3 -> vsPhase := 4;
			evPhase4_1: vsPhase == 4 -> vsPhase := 1;
	end

	// Subsystem used during the 1st phase
	block Phase1

			 // Call for the GS1 and Sats
		embeds main.GS1 as GS1;
		embeds main.Sat as Sat;

		Boolean vfWorking ( reset = false );
		assertion
		// either GS1's or Sat's output
			vfWorking := GS1.vfOutput and Sat.vfOutput;
	end
```

```
	// Subsystem used during the 2nd phase
	block Phase2

		embeds main.GS1 as GS1;
		embeds main.GS2 as GS2;
		embeds main.Sat as Sat;
		Boolean vfWorking ( reset = false );
		assertion
		// vsWorking
			vfWorking := (GS1.vfOutput or GS2.vfOutput) and
Sat.vfOutput;
	end

	// Subsystem used during the 3rd phase
	block Phase3

		embeds main.GS2 as GS2;
		embeds main.Sat as Sat;
		Boolean vfWorking( reset = false );
		assertion

			vfWorking := Sat.vfOutput and GS2.vfOutput;
	end

	// Subsystem used during the 4th phase
	block Phase4

		embeds main.GS1 as GS1;
		embeds main.GS2 as GS2;
		embeds main.SatRelay1 as SatRelay1;
		embeds main.SatRelay2 as SatRelay2;
		embeds main.Sat as Sat;
		Boolean vfWorking( reset = false );
		assertion

			vfWorking := (GS1.vfOutput or GS2.vfOutput) and Sat.vfOutput
and (SatRelay1.vfOutput or SatRelay2.vfOutput);
	end

	assertion

		vfWorking := if (PhaseController.vsPhase == 1) then
Phase1.vfWorking
				else if (PhaseController.vsPhase == 2) then
Phase2.vfWorking
				else if (PhaseController.vsPhase == 3) then
Phase3.vfWorking
				else if (PhaseController.vsPhase == 4) then
Phase4.vfWorking
```

```
                                else false;

        // Observer oFailed
        observer Boolean oFailed = not vfWorking;

end
```

Question 5: Common cause failures

```
/* Satellite subsystem
*       represented by a block diagram modeling pattern with non repairable
components
*/

class SatelliteSubSystem

        // Components
        NonRepairableInOutComponent Battery1;
        NonRepairableInOutComponent Battery2;
        NonRepairableInOutComponent Transmitter1;
        NonRepairableInOutComponent Transmitter2;
        NonRepairableInOutComponent Receiver1;
        NonRepairableInOutComponent Receiver2;

        Boolean vfOutput( reset = false );

        // Here we define the common cause failure of batteries in the satellite
        event batteries_fail;
        transition
                batteries_fail: ?Battery1.evFailure & ?Battery2.evFailure;

        // Connections
        assertion
                Battery1.vfInput := true;
                Battery2.vfInput := true;

                Transmitter1.vfInput := Battery1.vfOutput or Battery2.vfOutput;
                Transmitter2.vfInput := Battery1.vfOutput or Battery2.vfOutput;
                Receiver1.vfInput := Transmitter1.vfOutput or
Transmitter2.vfOutput;
                Receiver2.vfInput := Transmitter1.vfOutput or
Transmitter2.vfOutput;

                vfOutput := Receiver1.vfOutput or Receiver2.vfOutput;
end
```