SysML Hands-On Exercises

Exercise 6.2
SysML Activity Diagrams

MagicDraw

August 2018

OBJECTIVES

The objectives of this exercise are to

- Create an Activity and Activity Diagram to model autonomous flight onboard the UAV.
- Use actions, object flows, control flows, accept event and send signal actions, and interruptible regions.

This process is intended to represent modeling one part of the behavior of the UAV.

PREPARATION

1. This exercise assumes the student has Cameo System Modeler 19.0 (or MagicDraw 19.0 with SysML plug-in) installed correctly on his or her machine with a valid license for use.
2. The student should load the Part 6 course materials onto the computer, specifically Exercise 6.2 Starter UAV.mdzip and Exercise 6.2 Final UAV.mdzip.
3. The student should view the video Introduction to SysML Part 6 Exercise 6.2 in its entirety before attempting the exercise.

NOTES AND CAUTIONS

We recommend that the student watch the video demonstration of this exercise in its entirety before beginning their own work. The video includes background and explanatory material that is not repeated in the written instructions.

We also recommend that the student read the material carefully. The most common source of error is confusion between blocks, packages and diagrams, some of which have similar names. When the student is not sure what an element is, either in the browser or in a diagram, select that element and look in the Properties tab for the gray label that identifies the element type. Also, be careful in reading the instructions in realizing when an instruction should be carried out in the browser or in a diagram.

EXERCISE

6.2.1 Start Cameo System Modeler

6.2.2 Open Exercise 6.2 Starter UAV.mdzip

6.2.3 Create an Activity and Activity Diagram

- Right-click on the **UAV Behavior** package in the browser.
- Select Create Element → Activity.
- Name the activity **Fly Preprogrammed Course**.
- Right-click on the **Fly Preprogrammed Course** activity and select Create Diagram → SysML Activity Diagram.
- Drag the other four activities in the **UAV Behavior** package into the diagram and arrange them roughly as in Figure 1.
- Name the actions and rearrange the pins as shown in Figure 2.

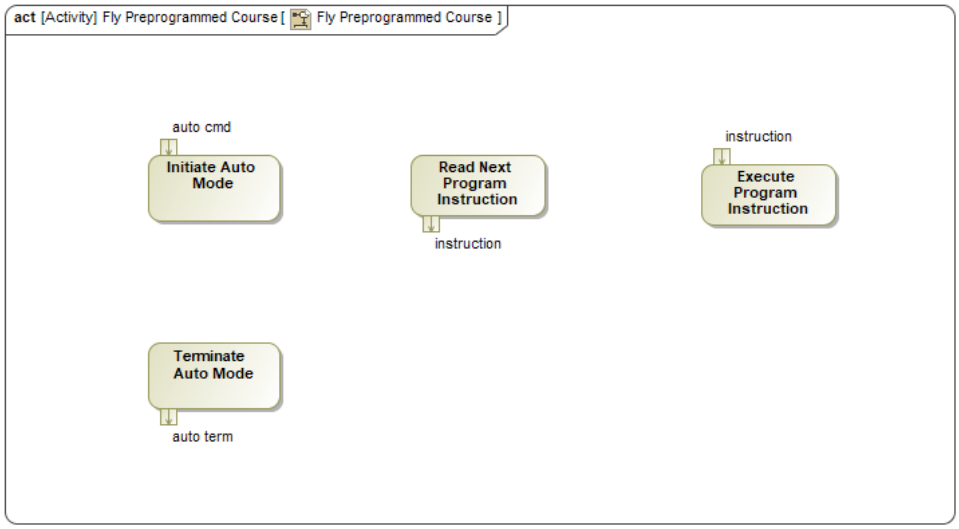

Figure 1 ***Fly Preprogrammed Course*** *diagram, first stage*

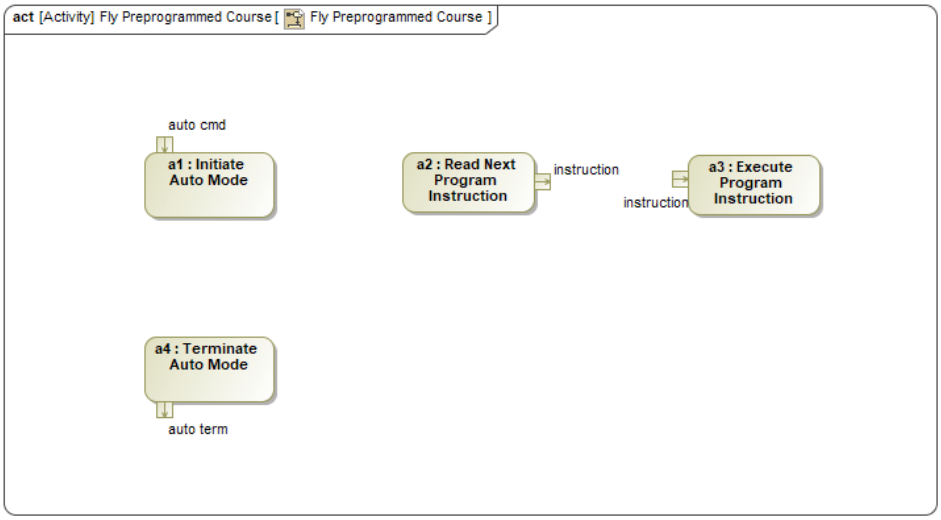

Figure 2 ***Fly Preprogrammed Course*** *diagram, second stage*

- Click on Initial Node in the Diagram Toolbar under the SysML Activity Diagram heading, then click to the left of **a1**.
- In the same way, place an Activity Final symbol to the left of **a4**.
- In the same way, place a Merge Node symbol to the right of **a1**.
- In the same way, place a Decision Node symbol to the left of **a2**.
- Double-click the Decision Node to open the Specification window. Name it **Prgm Instruction Available?** See Figure 3.
- Click on Accept Event Action in the Diagram Toolbar under the SysML Activity Diagram heading, then click somewhere above **a1.**
- Double-click the Accept Event Action to open the Specification window. Name it **Auto Cmd.** Set the Event Type under Trigger to CallEvent. Set Operation under Trigger to **auto()**, an Operation of the **UAV** block, by dragging the Operation from the containment browser onto the field in the action Specification Window. See Figure 4.
- Click the **Auto Cmd** symbol just created and show its Floating Toolbar. Click on the bottom icon, Output Pin.
- Click on Send Signal Action in the Diagram Toolbar under the SysML Activity Diagram heading, then click somewhere below **a4.**
- Double-click the Send Signal Action to open the Specification window. Name it **Auto Termination Signal.** Set Signal to **auto termination**, a Signal found under the **Pilot Station** block. See Figure 5.
- Click the **Auto Termination Signal** symbol just created and show its Floating Toolbar. Click on the Input Pin icon. Choose New Target Pin.
- Click on Accept Event Action in the Diagram Toolbar under the SysML Activity Diagram heading, then click somewhere below **a2.**
- Double-click the Accept Event Action to open the Specification window. Name it **Abort Auto Cmd.** Set the Event Type under Trigger to CallEvent. Set Operation under Trigger to **abort auto()**, an Operation of the **UAV** block. See Figure 6.

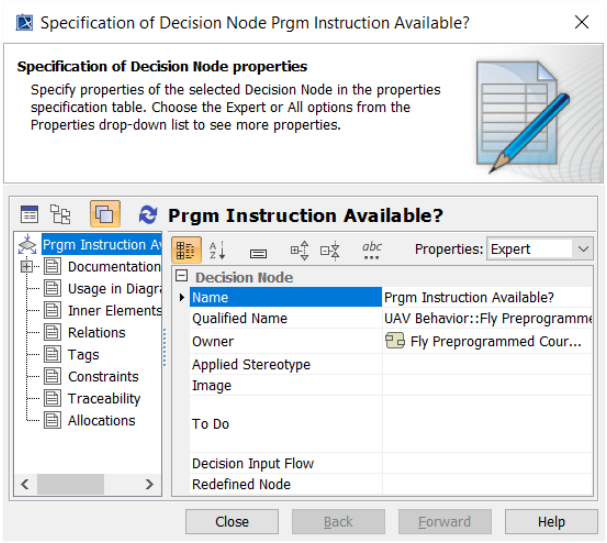

Figure 3 Specifying Decision Node

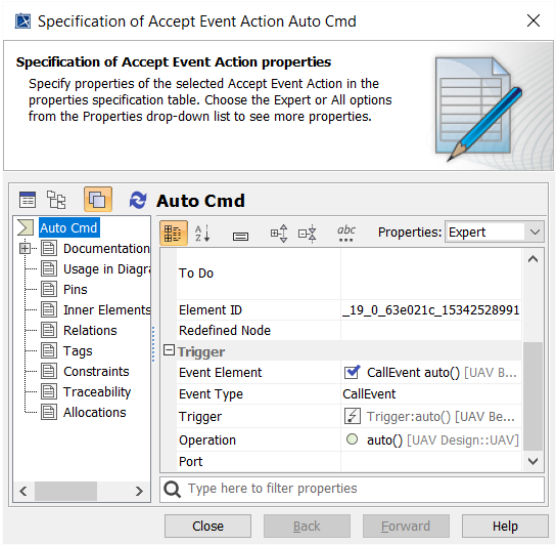

Figure 4 Specifying first Accept Event Action

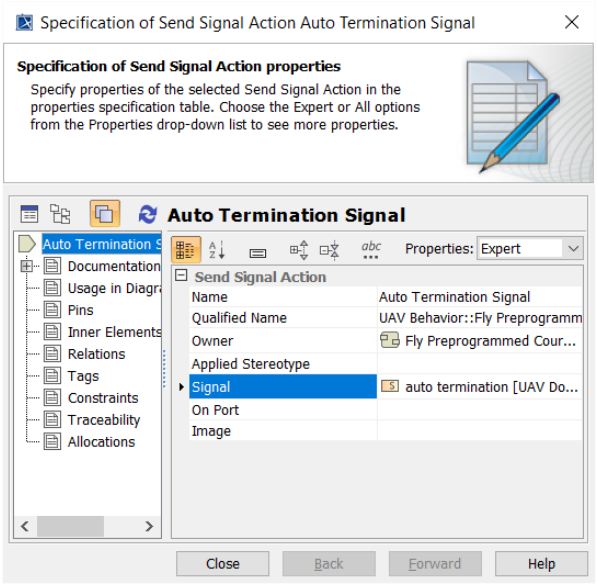

Figure 5 Specifying Send Signal Action

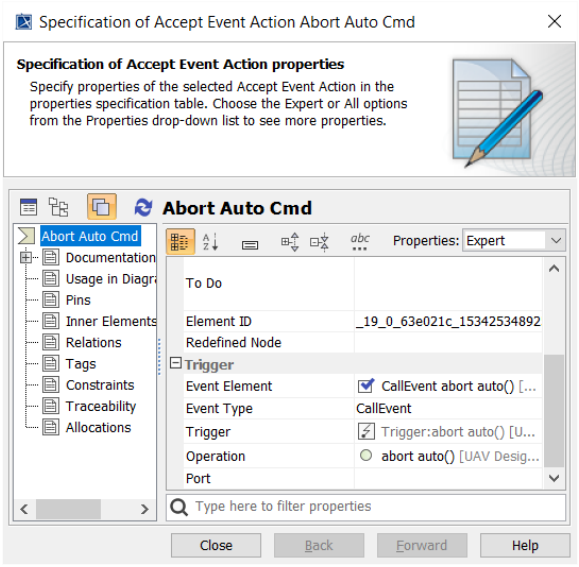

Figure 6 Specifying second Accept Event Action

- Click on Activity Final in the Diagram Toolbar under the SysML Activity Diagram heading, then click somewhere below **a3**.
- The diagram should now appear similar to Figure 7.

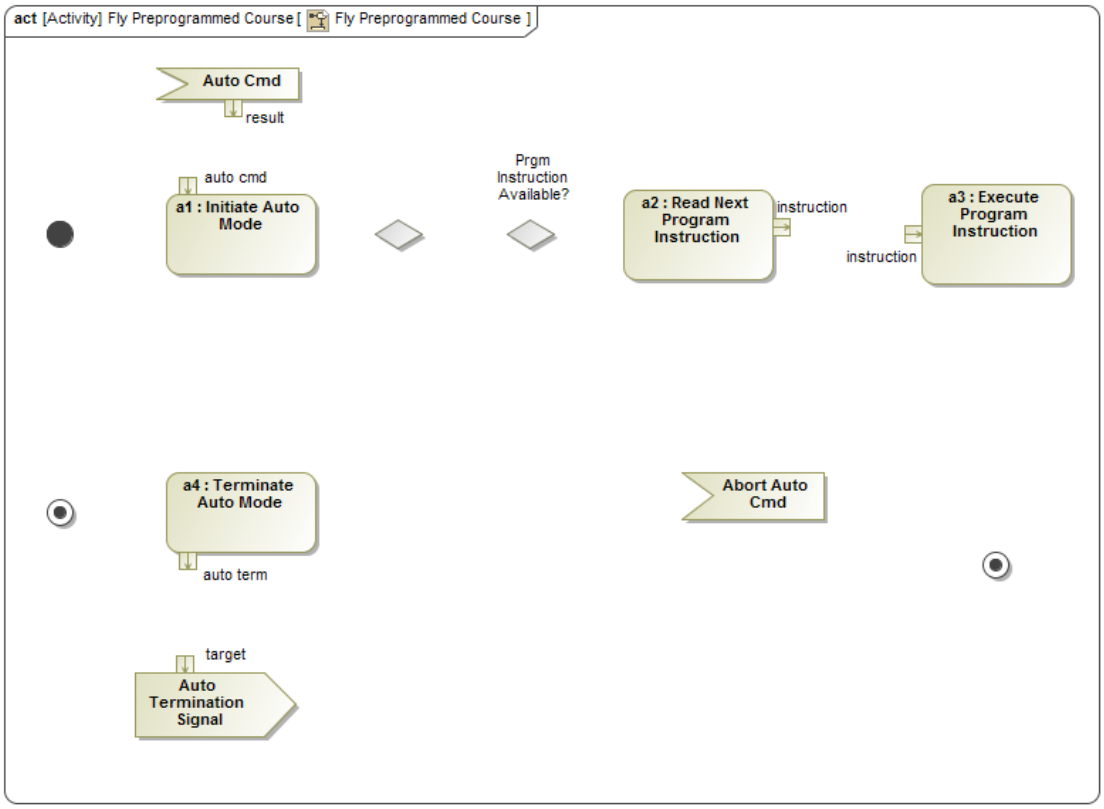

Figure 7 ***Fly Preprogrammed Course*** *diagram, third stage*

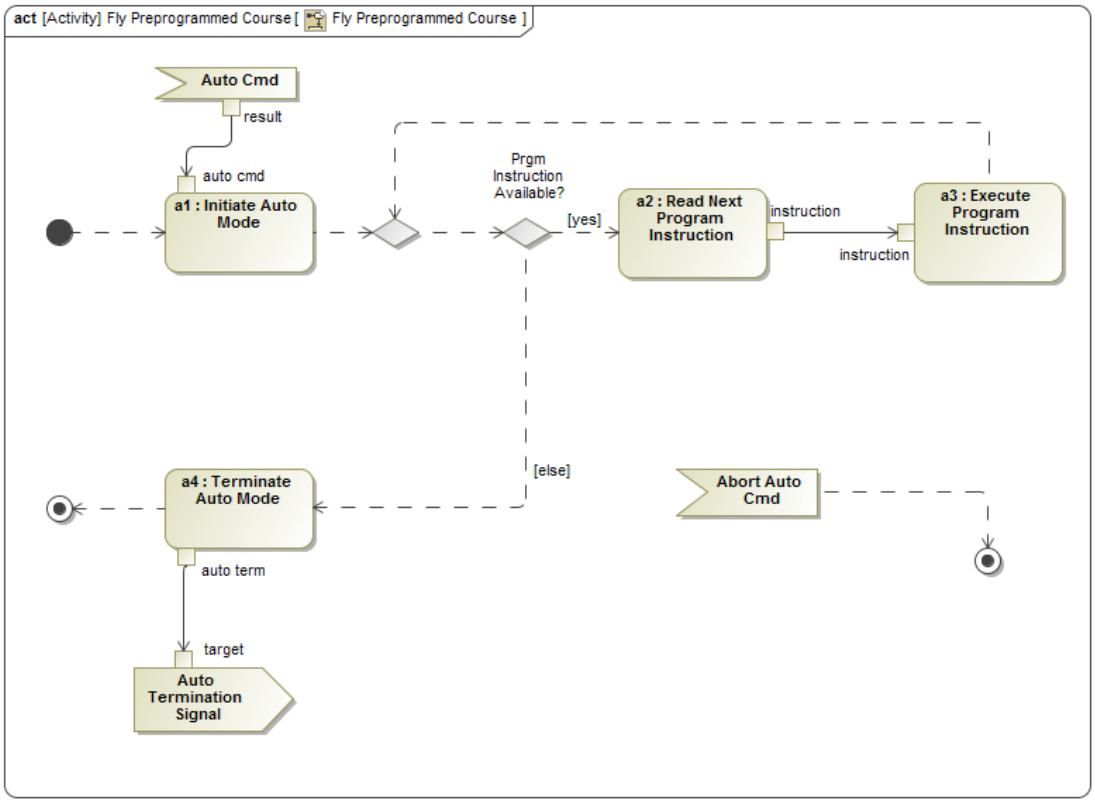

Figure 8 ***Fly Preprogrammed Course*** *diagram, fourth stage*

6.2.4 Add Object Flows and Control Flows

- Add three object flows to the diagram: from **Auto Cmd** to **a1**; from **a2** to **a3**; from **a4** to **Auto Termination Signal.**
- Add eight control flows to the diagram: from Initial Node to **a1**; from **a1** to the Merge Node; from the Merge Node to the Decision Node; from the Decision Node to **a2**; from **a3** to the Merge Node, from the Decision Node to **a4**; from **a4** to the Activity Final; and from **Abort Auto Cmd** to Activity Final.
- Double-click the control flow from the Decision Node to **a2**. In the Specification window, enter “yes” for Guard.
- Double-click the control flow from the Decision Node to **a4**. In the Specification window, enter “else” for Guard.
- The diagram should appear similar to Figure 8.

6.2.5 Add Interruptible Activity Region

- Click on Interruptible Activity Region in the Diagram Toolbar under the SysML Activity Diagram heading, then click somewhere above and to the left of **a2** and drag it down to somewhere below and to the right of **Abort Auto Cmd**.
- Enter the name **Emergency** at the top of the region.
- Shade the region by right-clicking on the region boundary, selecting Symbol Properties, and checking Use Fill Color.
- The final diagram should appear similar to Figure 9.
- Save and close the project.

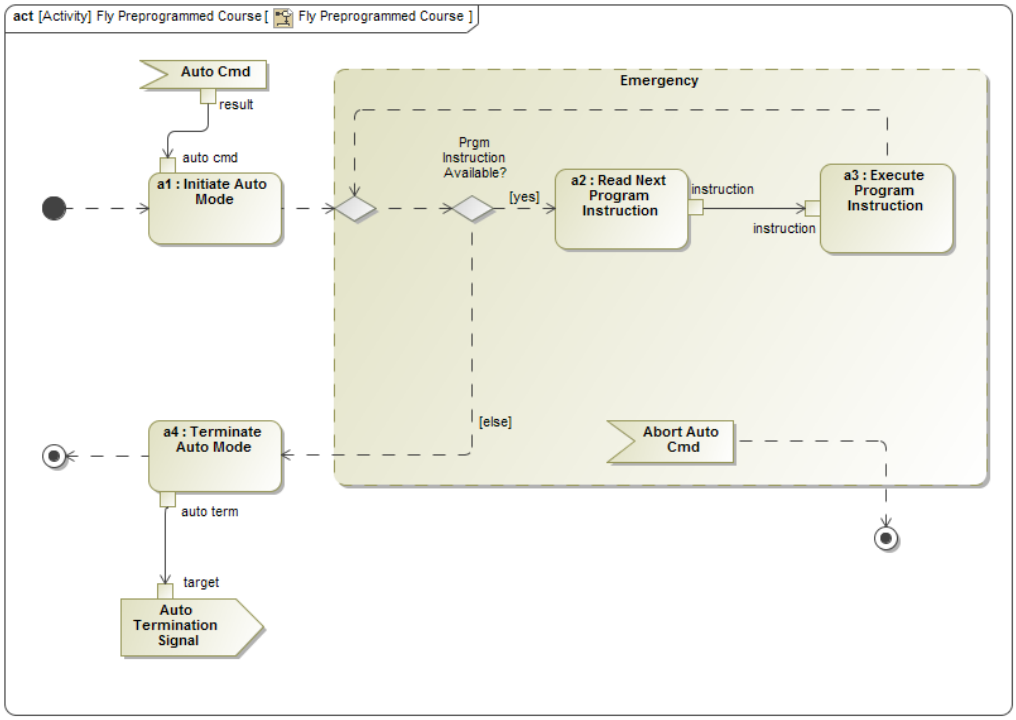

Figure 9 ***Fly Preprogrammed Course*** *diagram, final stage*