SysML Hands-On Exercises

Exercise 8.1
SysML State Machine Diagrams

MagicDraw

August 2018

OBJECTIVES

The objective of this exercise is to create a State Machine and State Machine Diagram to describe the Classifier Behavior of the UAV. This process captures the top-level behavior of the UAV and provides a framework to organize its lower-level behaviors.

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 8 course materials onto the computer, specifically Exercise 8.1 Starter UAV.mdzip and Exercise 8.1 Final UAV.mdzip.
3. The student should view the video Introduction to SysML Part 8 Exercise 8.1 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

8.1.1 Start Cameo System Modeler

8.1.2 Open Exercise 8.1 Starter UAV.mdzip

8.1.3 Create a State Machine and State Machine Diagram

- Right-click the **UAV** block in the browser and open the Specification window.
- In the left-hand box, select Behaviors. Click the Create button (see Figure 1) and choose State Machine from the list at the bottom.
- In the Specification window for the new state machine, name it **Flight States** (see Figure 2).
- Click the Back Button to return to the **UAV** block specification. Confirm that **Flight States** is listed as the Classifier Behavior.
- In the browser, right-click **Flight States** and select Create Diagram → SysML State Machine Diagram. The initial diagram will consist of an initial pseudostate and a state.

8.1.4 Create States and Do Behaviors

- Select Final State in the Diagram Toolbar (under the State Machine Diagram heading), then click inside the diagram frame.
- Select State in the Diagram Toolbar (under the State Machine Diagram heading), then click inside the diagram frame. Repeat twice.
- Name the states **Power On**, **Manual Flight**, **Autonomous Flight** and **Autonomous Landing**. Rearrange and resize the state symbols roughly as shown in Figure 3.
- Double-click the **Power On** state to open its Specification window. Scroll down to the section headed Entry. Select Behavior Type as Activity and Name as **Run Diagnostics**. See Figure 4. Hit Close.
- Add Do Behaviors to the remaining states as follows:
  - For **Manual Flight**, an activity named **Fly under Manual Control**
  - For **Autonomous Flight**, an activity named **Fly on Autopilot**
  - For **Autonomous Landing**, an activity named **Return to Base and Land**
- When finished, the diagram should look similar to Figure 6.

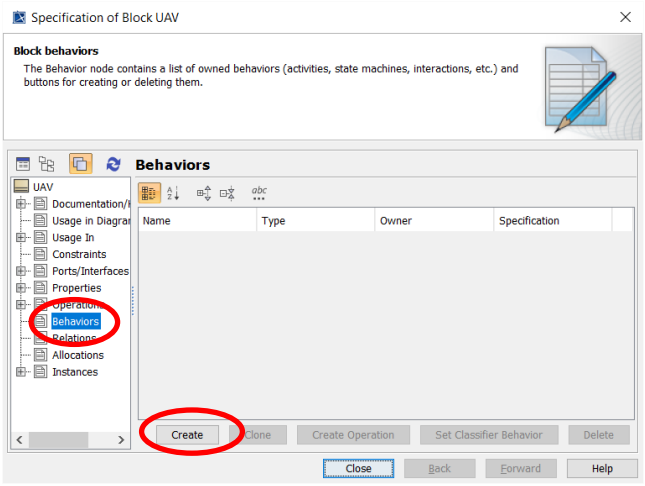

*Figure 1 Specification window, **UAV** block*

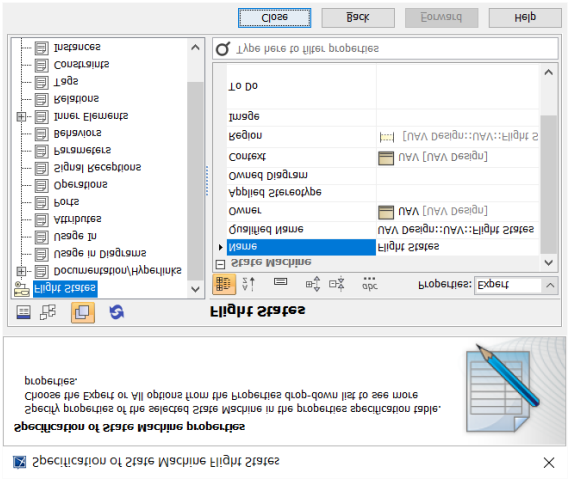

Figure 2 Specification window, new state machine

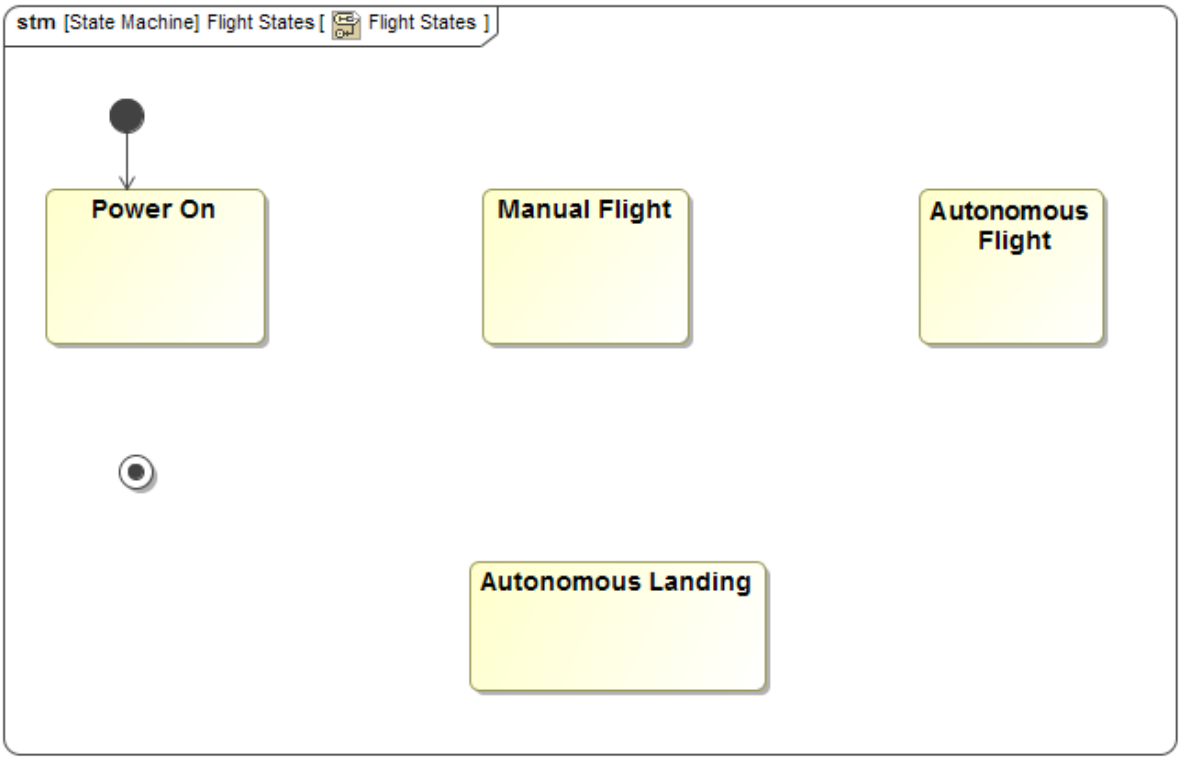

*Figure 3 **Flight States** state machine diagram, first stage*

8.1.5 **Create Transitions**

- Click on Power On state in diagram and launch Floating Toolbar. Select the arrow icon (Transition) and drag the end of the arrow to **Manual Flight**.
- Double-click on the transition arrow and open its Specification window.

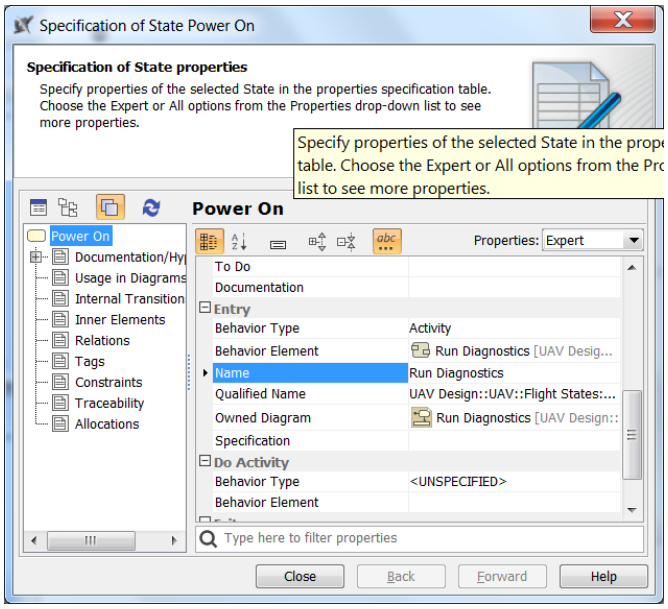

*Figure 4 Specification window, **Power On** state*

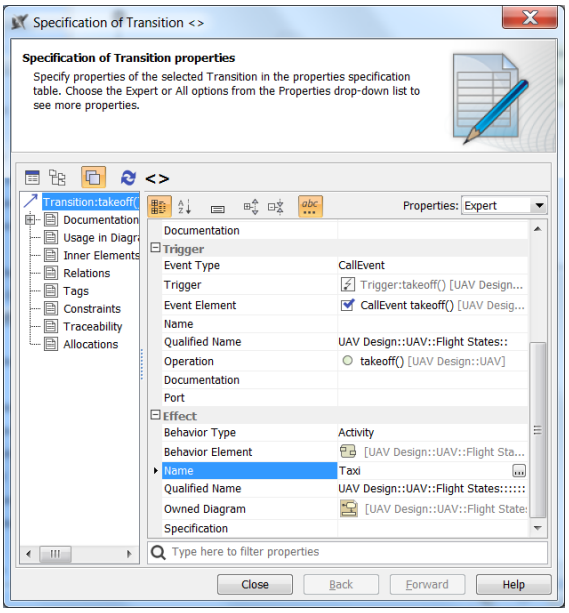

Figure 5 Specification window, first transition

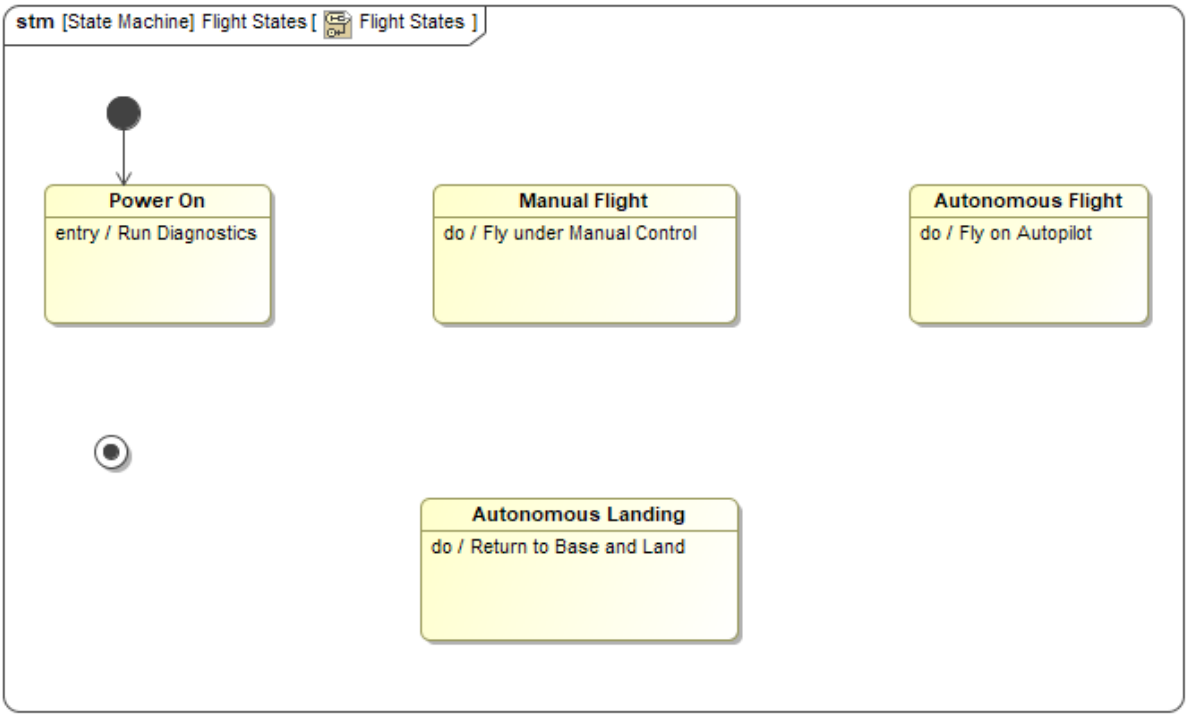

*Figure 6 **Flight States** state machine diagram, second stage*

- Under Trigger, set Event Type to CallEvent and Operation to **takeoff()**, which can be found under the **UAV** block in the tree.
- Under Effect, set Behavior Type to Activity and Name to **Taxi**. See Figure 5.
- Create the remaining transitions according to Table 1.

From	To	Trigger	Guard	Effect
Manual Flight	Autonomous Flight	EventType = CallEvent Operation = auto()	Comm == Stable	
Manual Flight	Power On	EventType = ChangeEvent Change Expression = Landed		
Autonomous Flight	Manual Flight	EventType = CallEvent Operation = abort_auto()		
Autonomous Flight	Manual Flight	EventType = ChangeEvent Change Expression = Course Complete		
Autonomous Flight	Autonomous Landing	EventType = ChangeEvent Change Expression = Comm==Failed		
Manual Flight	Autonomous Landing	EventType = ChangeEvent Change Expression = Comm==Failed		
Autonomous Landing	Manual Flight	EventType = ChangeEvent Change Expression = Comm Restored		Behavior Type = Activity, Name = Issue Alert
Autonomous Landing	Power On	EventType = ChangeEvent Change Expression = Landed		
Power On	Final State	EventType = CallEvent Operation = power_down()		

*Table 1 Transitions in **Flight States***

- The final diagram should look similar to Figure 7.

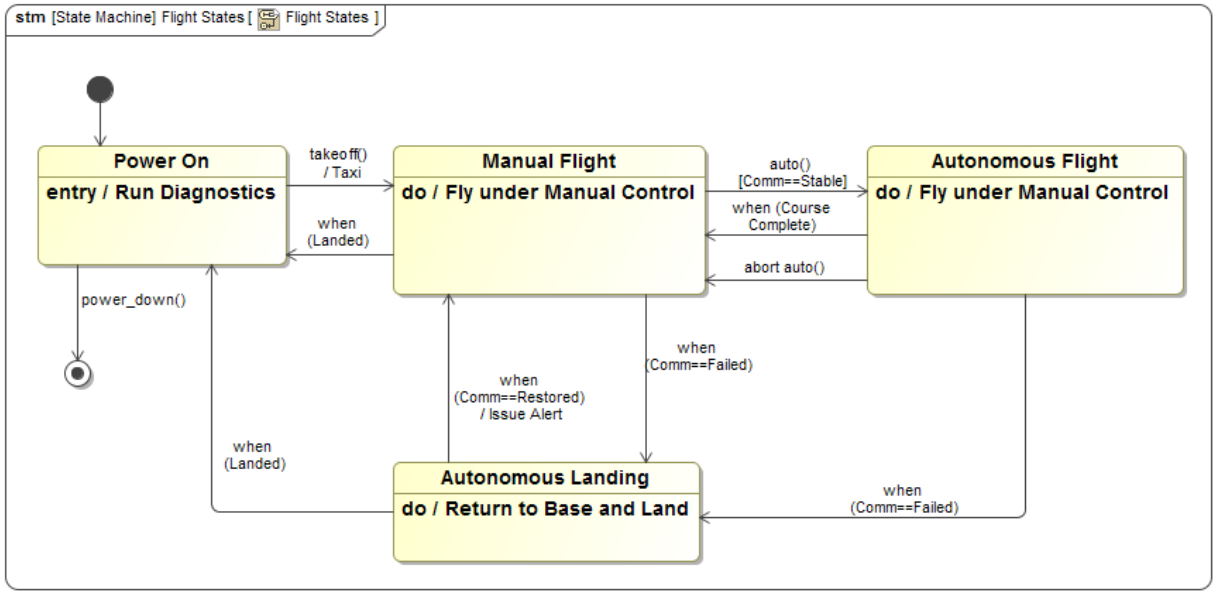

*Figure 7 **Flight States** state machine diagram, final stage*