SysML Hands-On Exercises

Exercise 8.2
SysML Sequence Diagrams

MagicDraw

August 2018

OBJECTIVES

The objective of this exercise is to create an Interaction and Sequence Diagram to describe the full flight cycle of the UAV in the context of the UAV Domain. This interaction references several lower-level interactions and uses several combined fragments to represent three different scenarios of operation.

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 8.2 Starter UAV.mdzip and Exercise 8.2 Final UAV.mdzip.
3. The student should view the video Introduction to SysML Part 8 Exercise 8.2 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

8.2.1 Start Cameo System Modeler

8.2.2 Open Exercise 8.2 Starter UAV.mdzip

8.2.3 Create an Interaction and Sequence Diagram

- Right-click the **UAV Domain:: UAV Domain Behaviors** package in the browser and select Create Element → Interaction. Name the interaction **UAV Flight Complete Cycle**.
- In the browser, right-click **UAV Flight Complete Cycle** and select Create Diagram → SysML Sequence Diagram.
- Drag three part properties of the **Unmanned Aerial System Domain** block from the browser into the diagram: **uAV:UAV**, **p_stn:Pilot Station**, and **p:Pilot** to create lifelines.

8.2.4 Create State Invariants and Interaction Uses

- Select State Invariant in the Diagram Toolbar (under the Sequence Diagram heading), then click on the tail of the **uAV:UAV** lifeline.
- In the Select State window (see Figure 1), check **Manual Flight** and click OK.
- Select Interaction Use in the Diagram Toolbar (under the Sequence Diagram heading), then click inside the diagram to the left of the first lifeline, dragging the symbol across all three lifelines.
- Open the Specification window for the interaction use and select **UAV Takeoff** in the Refers To pull-down list (see Figure 2).
- Repeat the previous two steps to create another Interaction Use referencing **UAV Manual Flight Control**.
- The diagram should appear similar to Figure 3

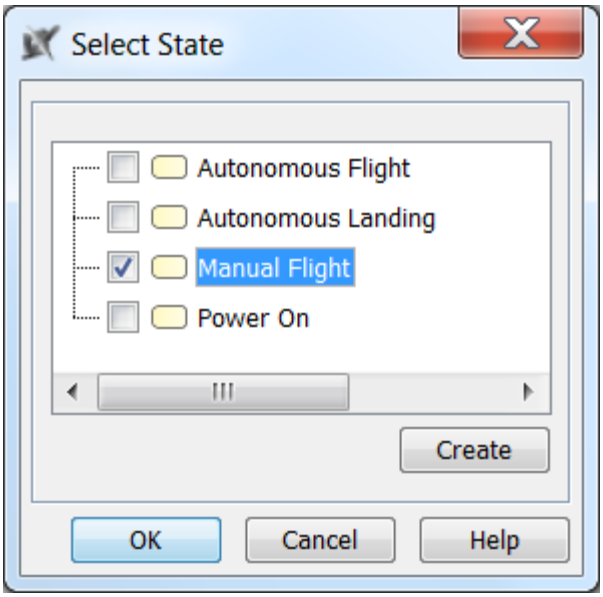

*Figure 1 Select State window, **uav**:**UAV** lifeline*

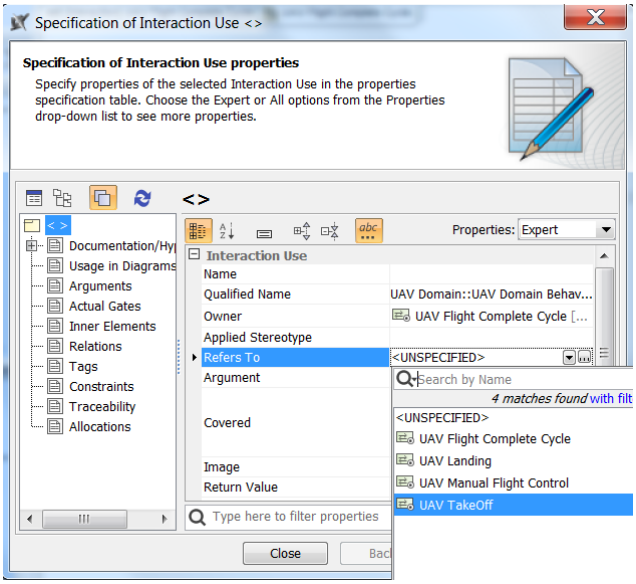

Figure 2 Specification window, interaction use

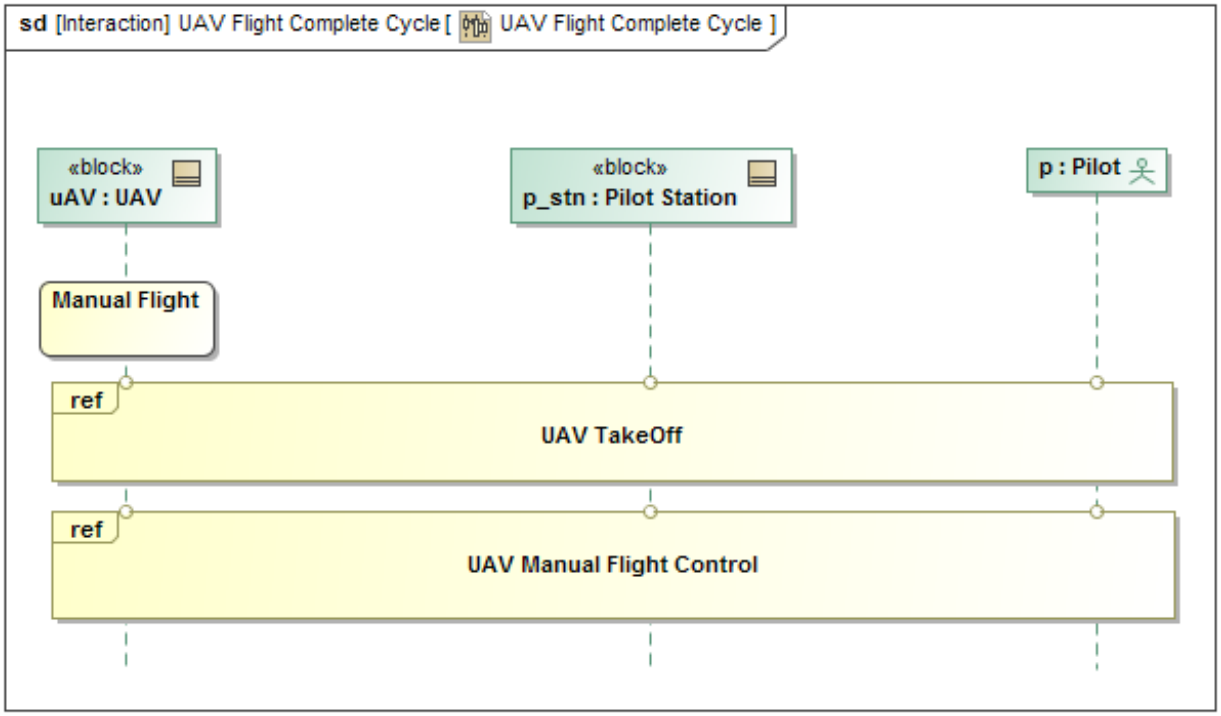

*Figure 3 **UAV Complete Flight Cycle** sequence diagram, first stage*

8.2.5 Create OPT and ALT Combined Fragments

- Select Option in the Diagram Toolbar (under the Sequence Diagram heading), then then click inside the diagram to the left of the first lifeline, dragging the symbol across all three lifelines. Click on the guard condition brackets and enter “Autonomous Mode”.
- Inside the opt combined fragment, draw an asynch call message, Operation = **inititiate_auto_mode(),** from **p:pilot** to **p_stn:Pilot Station**.
- Inside the opt combined fragment, draw an asynch call message, Operation = **auto(),** from **p_stn:Pilot Station** to **uAV:UAV**.
- Select State Invariant in the Diagram Toolbar (under the Sequence Diagram heading), then click on the tail of the **uAV:UAV** lifeline.
- In the Select State window check **Autonomous Flight** and click OK.
- Select Alternatives in the Diagram Toolbar (under the Sequence Diagram heading), then then click inside the opt combined fragment to the left of the first lifeline, dragging the symbol across all three lifelines.
- Double-click to edit the guard brackets in the first compartment and enter “early termination”
- Inside the top compartment, draw an asynch call message, Operation = **inititiate_auto_abort(),** from **p:pilot** to **p_stn:Pilot Station**.
- Inside the top compartment, draw an asynch call message, Operation = **abort auto(),** from **p_stn:Pilot Station** to **uAV:UAV**.
- Inside the lower compartment, draw an asynch signal message, Signal = **auto termination** from **uAV:UAV** to **p_stn:Pilot Station**.
- Inside the lower compartment, draw an asynch call message, Name = **Preprogrammed course complete** from **p_stn:Pilot Station** to **p:pilot**.
- Select State Invariant in the Diagram Toolbar (under the Sequence Diagram heading), then click on the tail of the **uAV:UAV** lifeline.
- In the Select State window, check **Manual Flight** and click OK.
- Create another Interaction Use referencing **UAV Manual Flight Control**.
- Below the bottom of the opt frame, create another Interaction Use referencing **UAV Landing.**
- The final diagram should look similar to Figure 4.
- Save and close the project.

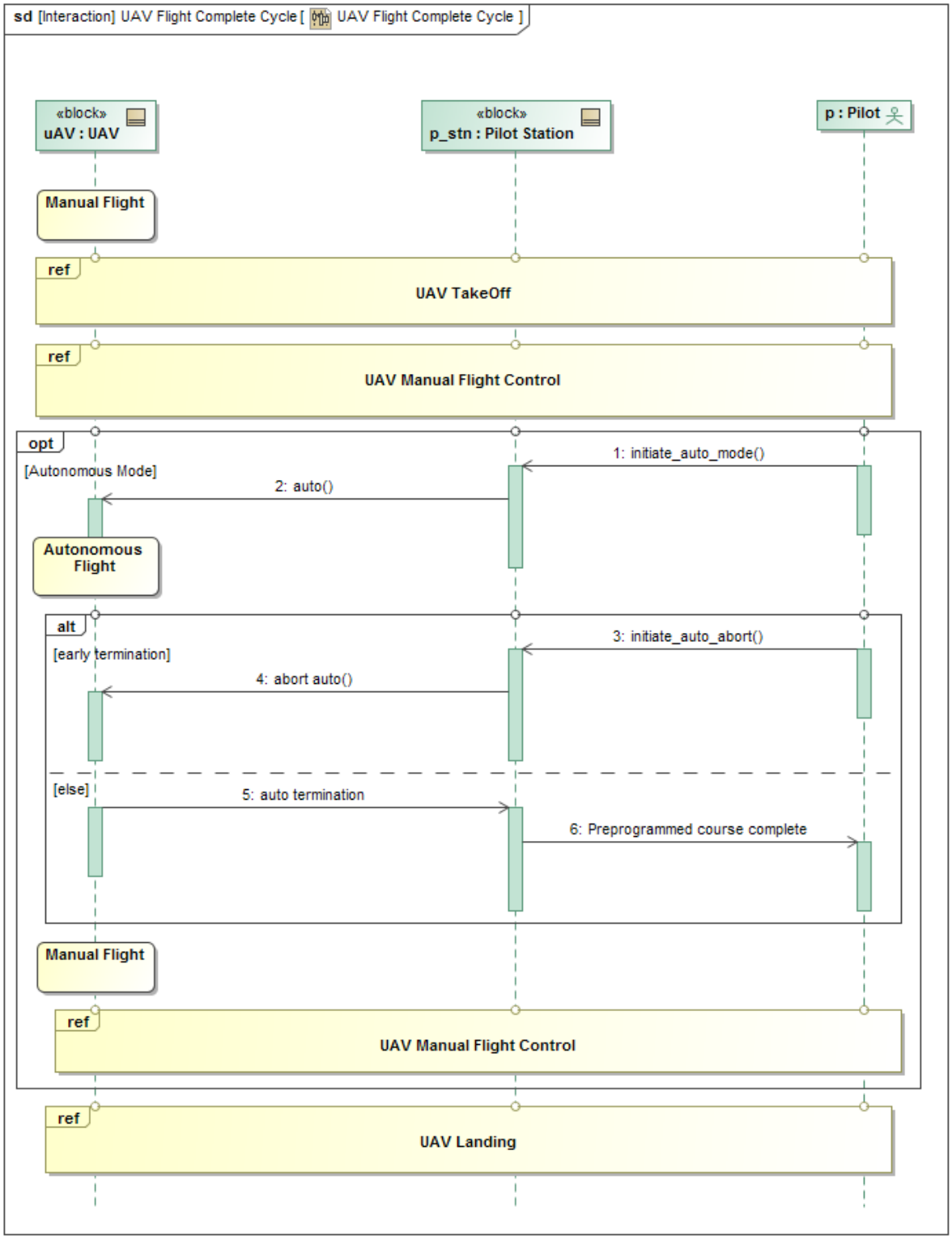

Figure 4 ***UAV Complete Flight Cycle*** *sequence diagram, final stage*