Introduction to SysML

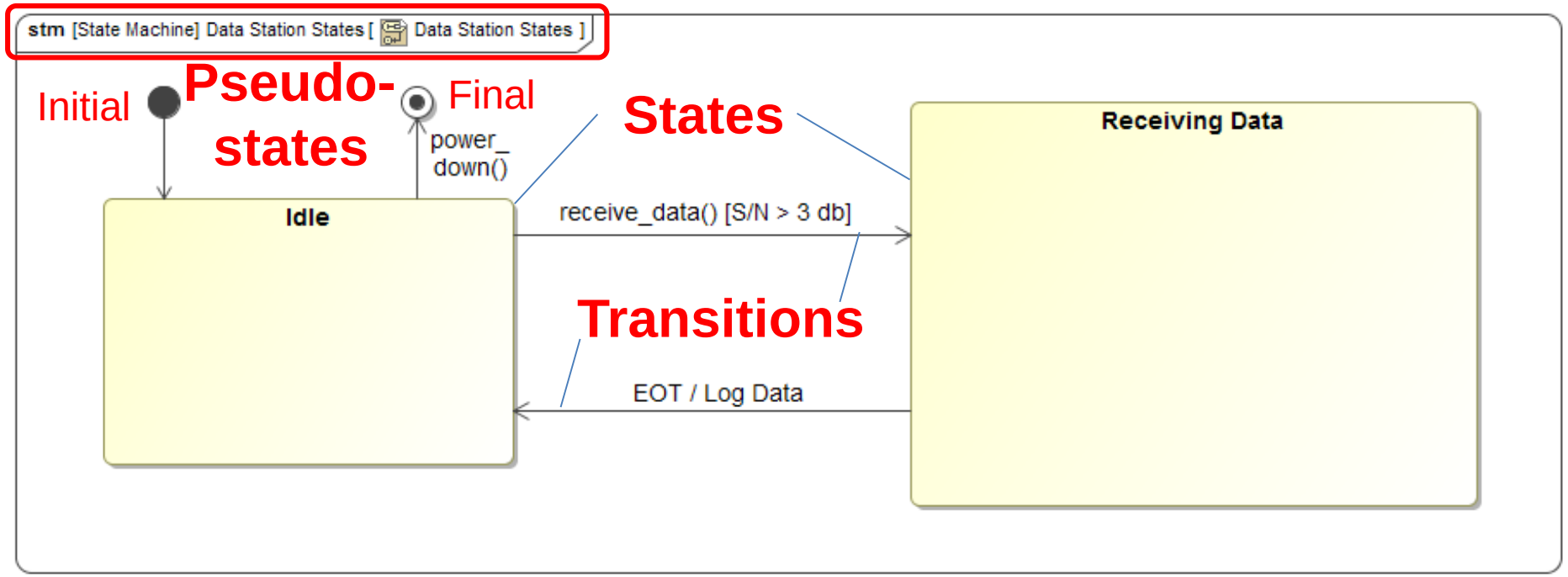
Part 8.0: State Machine and Sequence Diagrams

With tutorial exercises using Magicdraw

State Machines

- State Machine is a kind of Behavior
- State Machine Diagram is a view of a State Machine
- A State is a collection of Behaviors that the system exhibits at some stage of its lifecycle or under certain conditions
- States are mutually exclusive within a State Machine
- A State Machine can be in one and only one state at a time
- A Transition is an allowable path from one State to another
- State Machines are often the top-level Classifier Behaviors of Blocks
- State Machines are a compact way to describe systems with distinct operating modes and subject to many internal and external factors

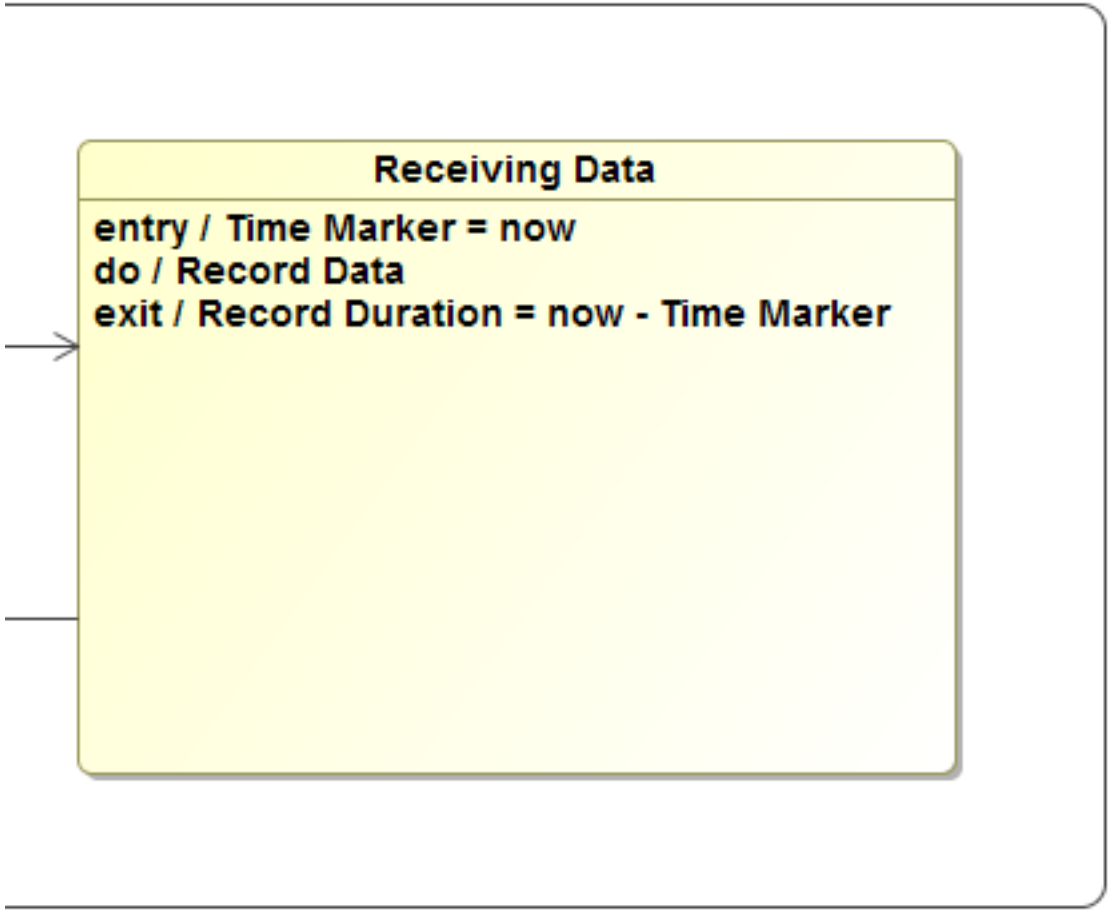
State Machines



- Initial Pseudostate identifies first state after state machine becomes active.
- Final State identifies state system must be in to terminate the state machine.
- If no Final State symbol, state machine may terminate from any state.

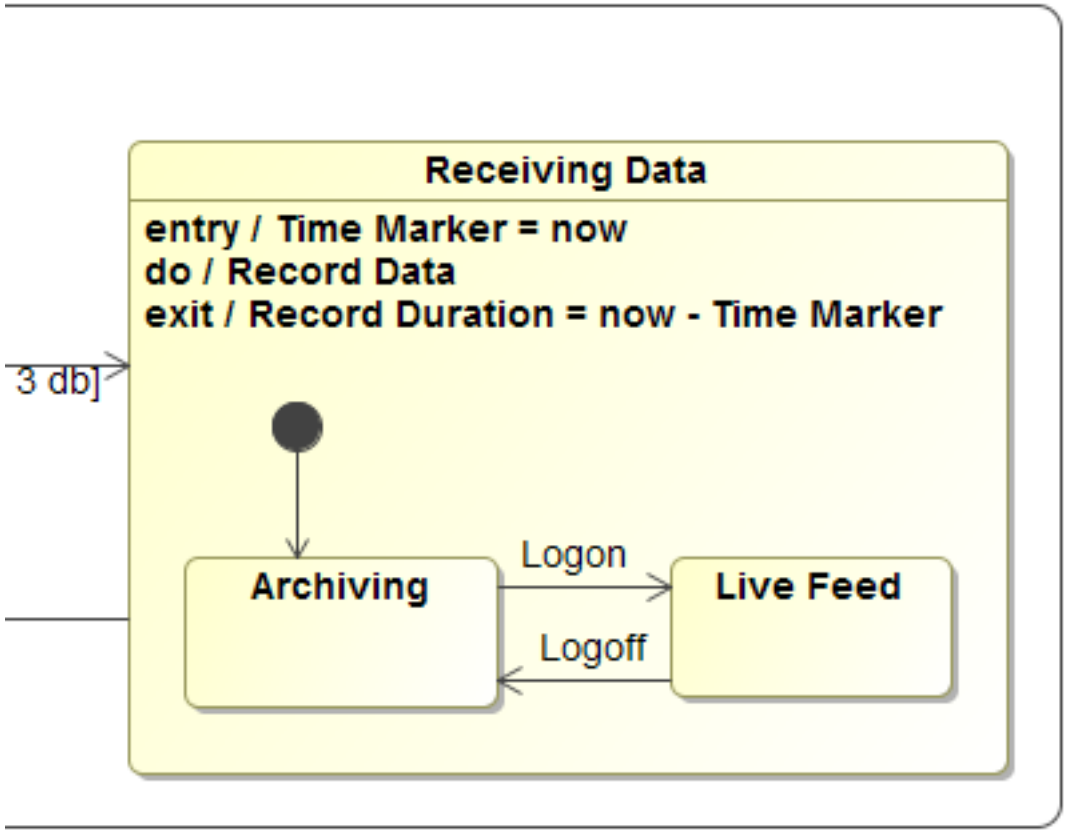
States

- A State is a collection of Behaviors.
- The text inside a State lists them
  - Entry – behaviors carried out as soon as the state is entered
  - Do – behaviors carried out while the system is in that state (or until natural completion)
  - Exit – behaviors carried out just before the system transitions out of a state



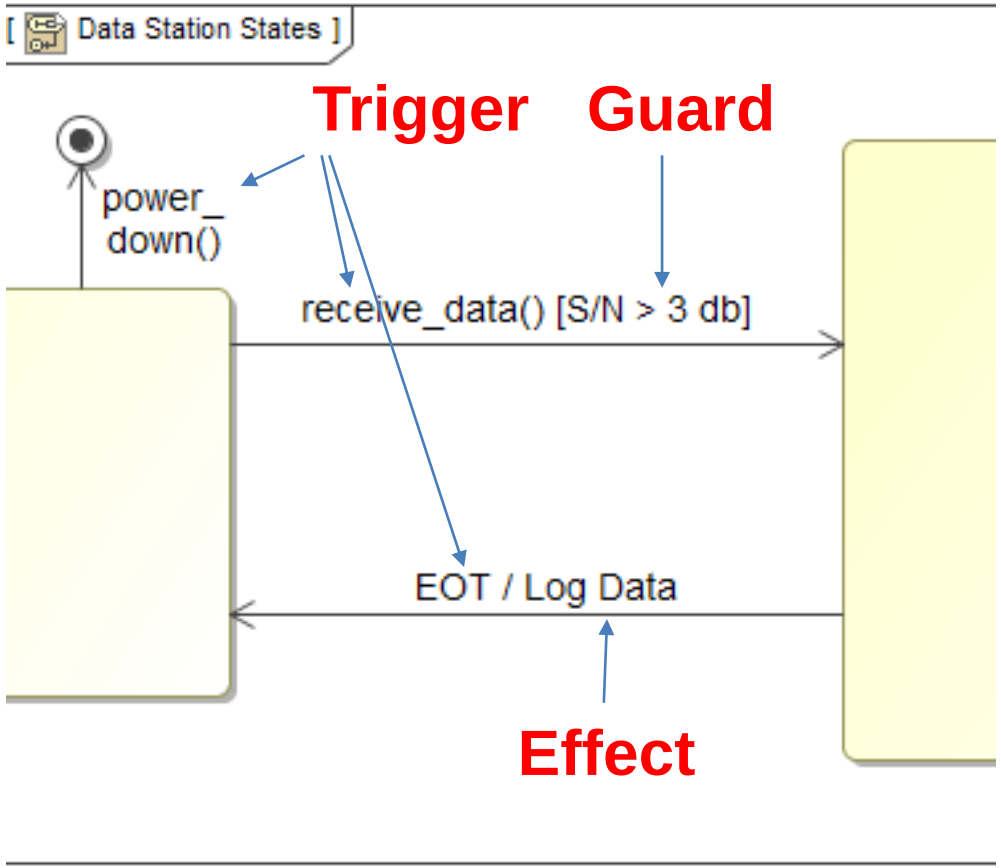
States

- States can contain State Machines as do behaviors.
- Such a State is called a Composite State.
- A Composite State can contain more than one State Machine.
- These parallel State Machines are called Concurrent Behaviors.
- SysML uses an organizing element called a Region to separate States in different State Machines.

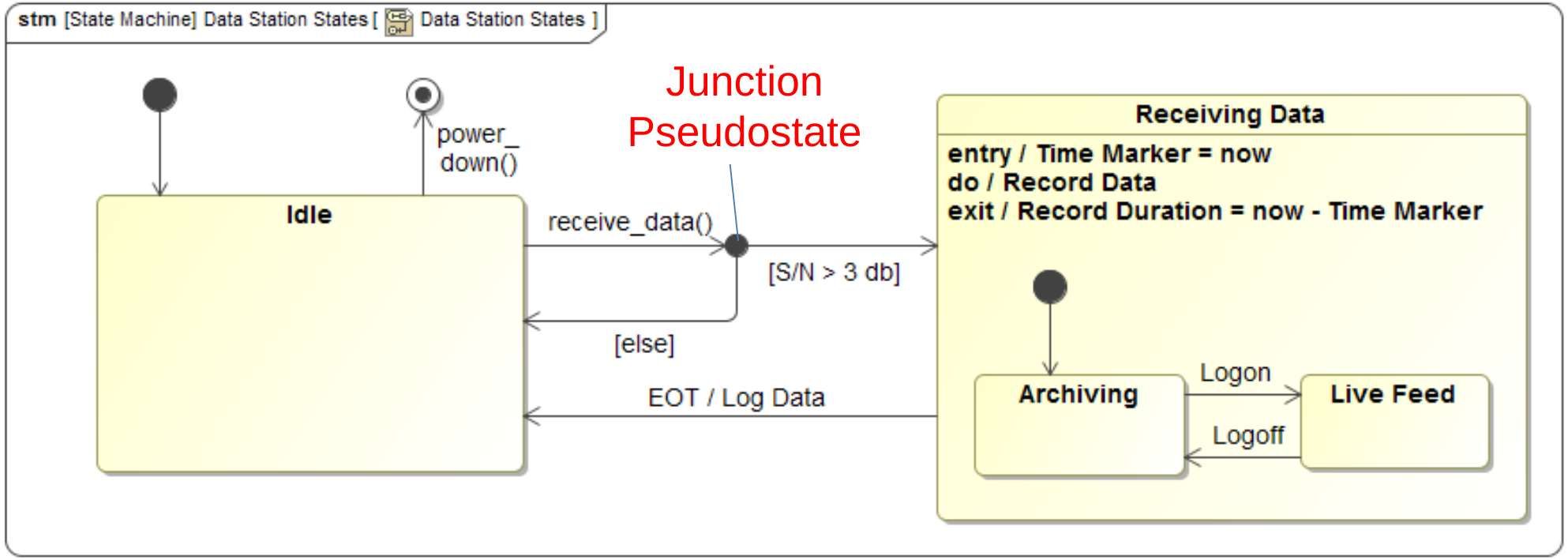


Transitions

- **Triggers [Guard] / Effect**
- A Trigger is an Event that causes a transition to occur
- These may be Call, Signal, Time, Change or Completion events
- Every Transition must have at least one Trigger, though it may not be shown
- A Guard is a condition that must be true for a Transition to occur
- An Effect is a Behavior that occurs during the Transition.

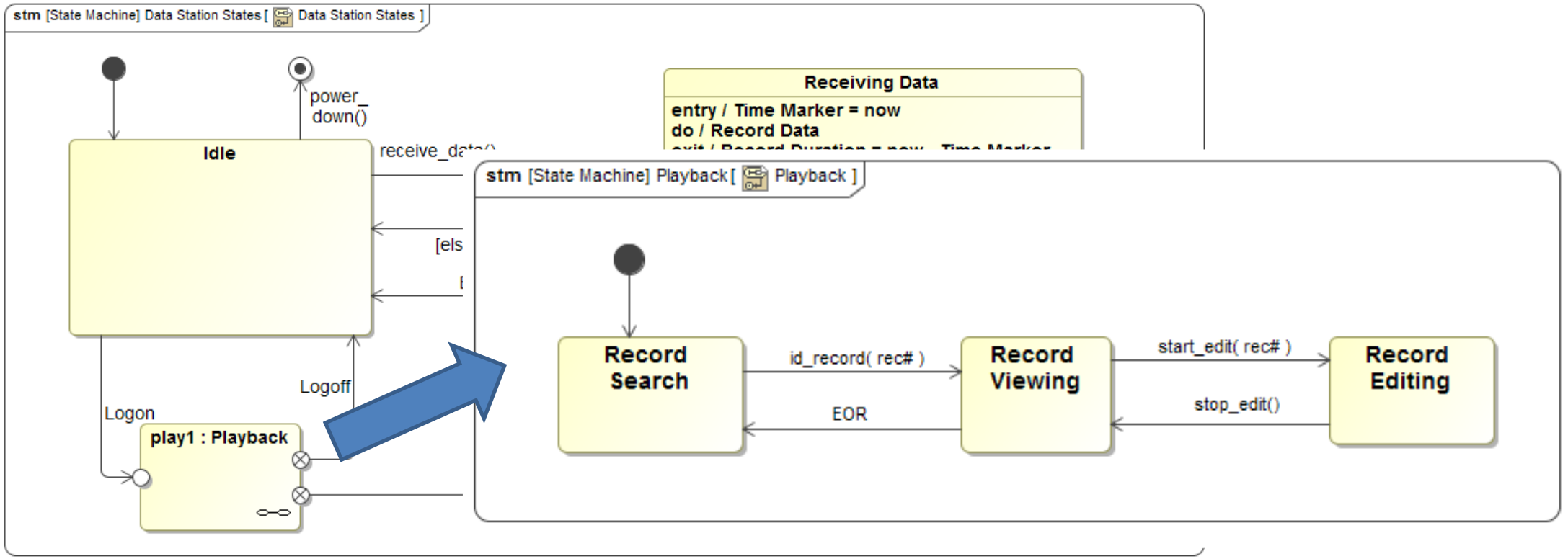


Transitions



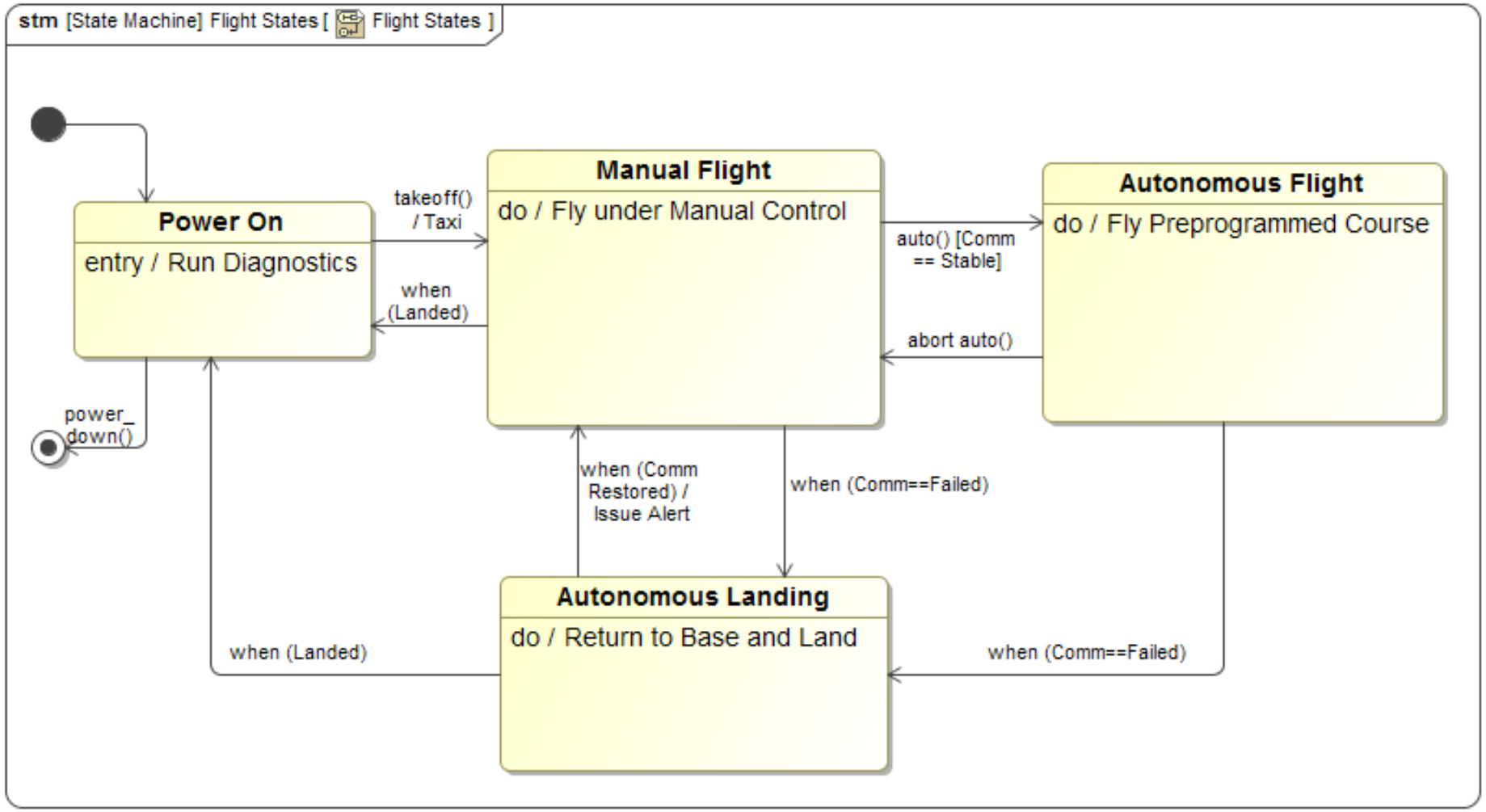
- A Junction Pseudostate allows a Transition to branch, controlled by Guard conditions.

Submachine State

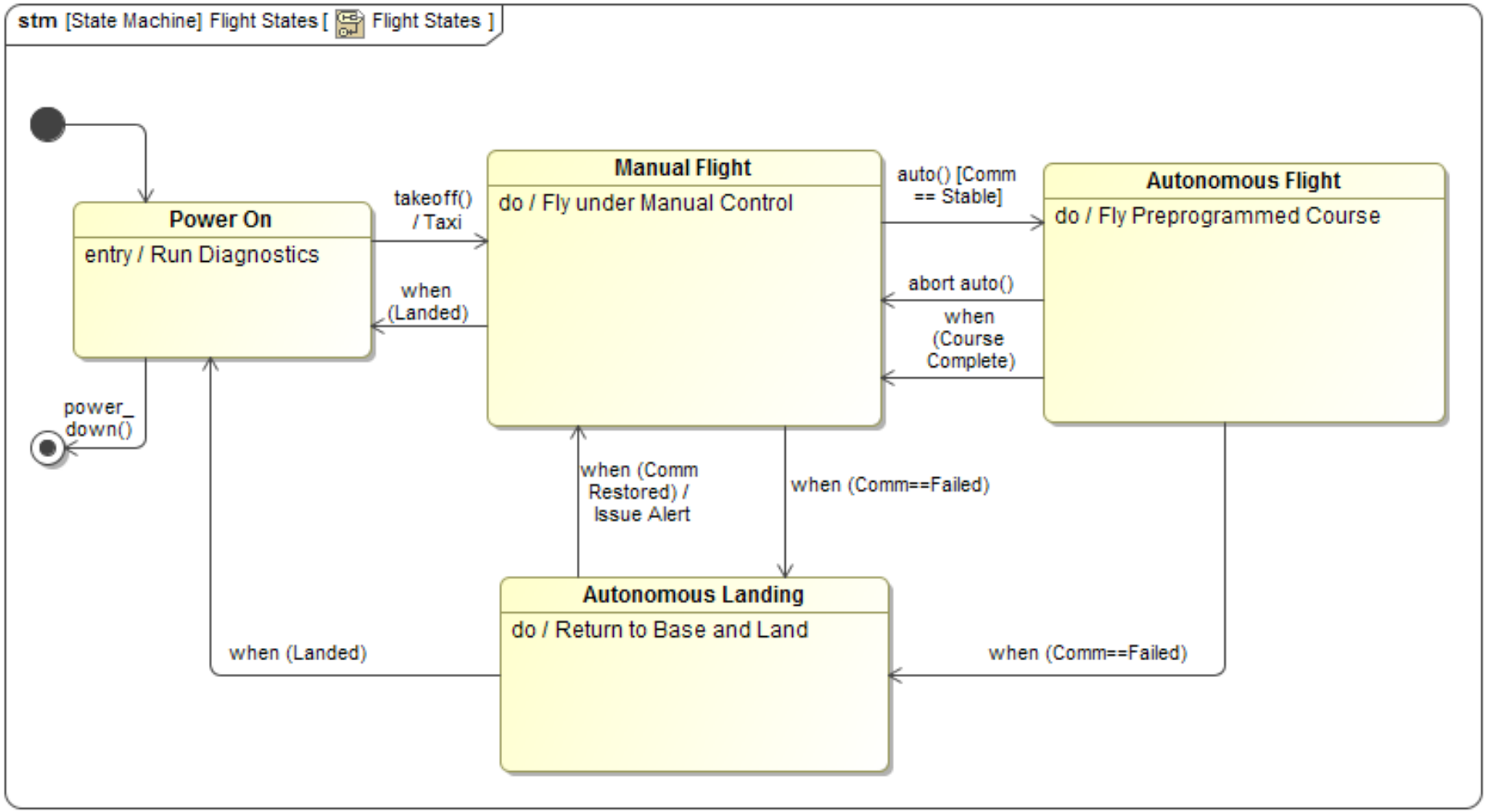


- A Submachine State is a usage of a State Machine in a higher-level State Machine Diagram. The small state machine icon in the lower right corner and the entry and exit points identify it.

UAV Flight State Machine

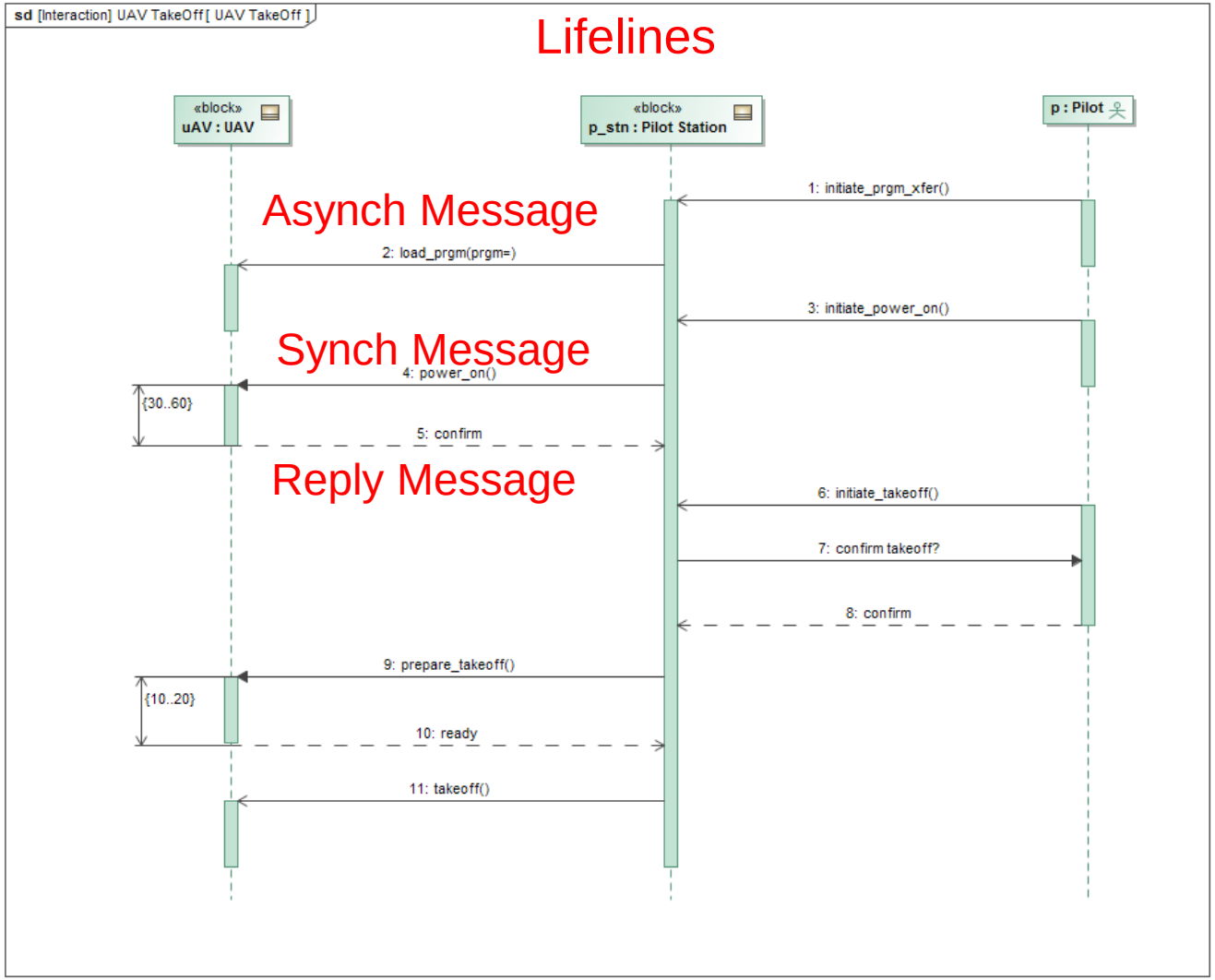


First Exercise



Review - Interactions and Sequence Diagr

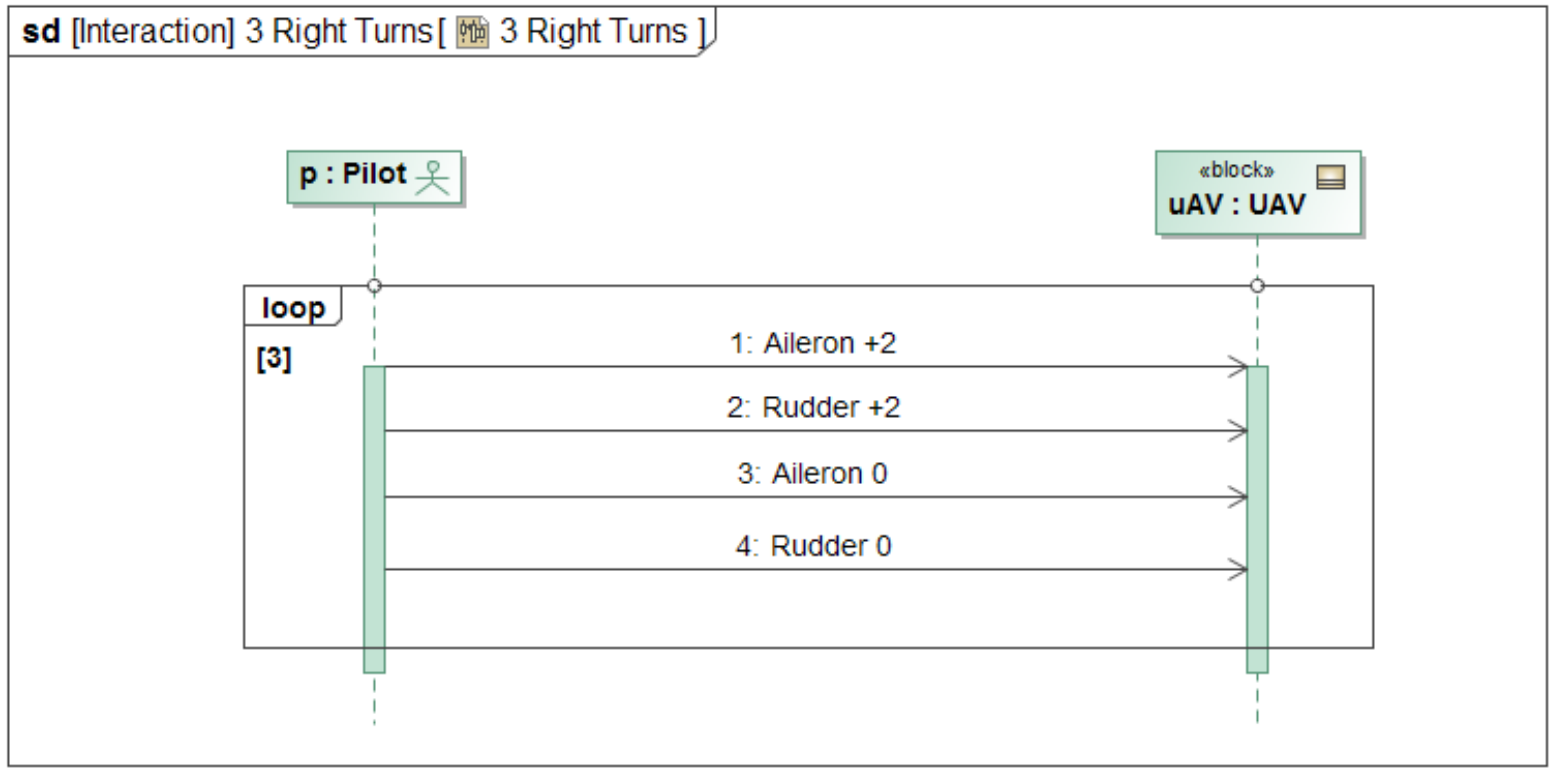
- Interactions model message-based behaviors
- A Sequence Diagram shows the internal elements of an Interaction
- Lifelines represent part properties, parts of the system or domain
- Messages can be synchronous, asynchronous or reply, and are read from the top down.



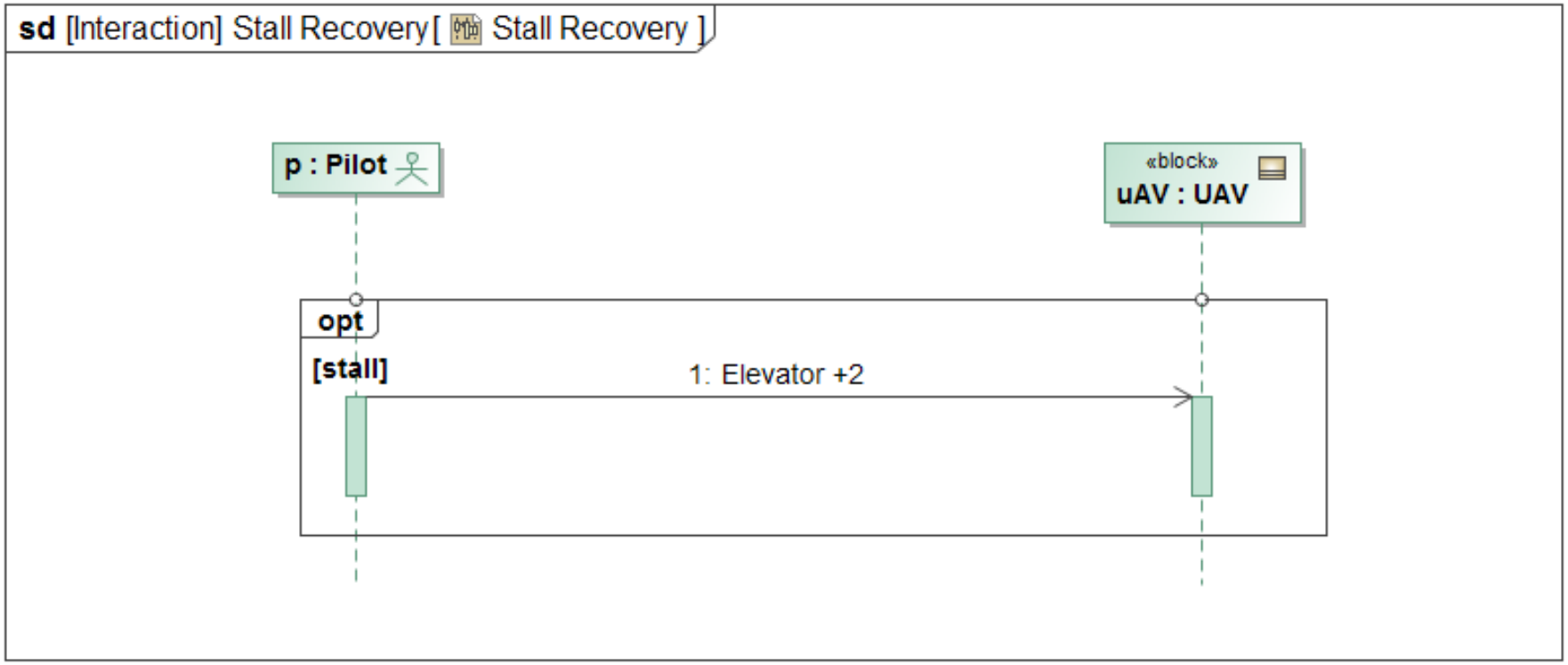
Combined Fragments

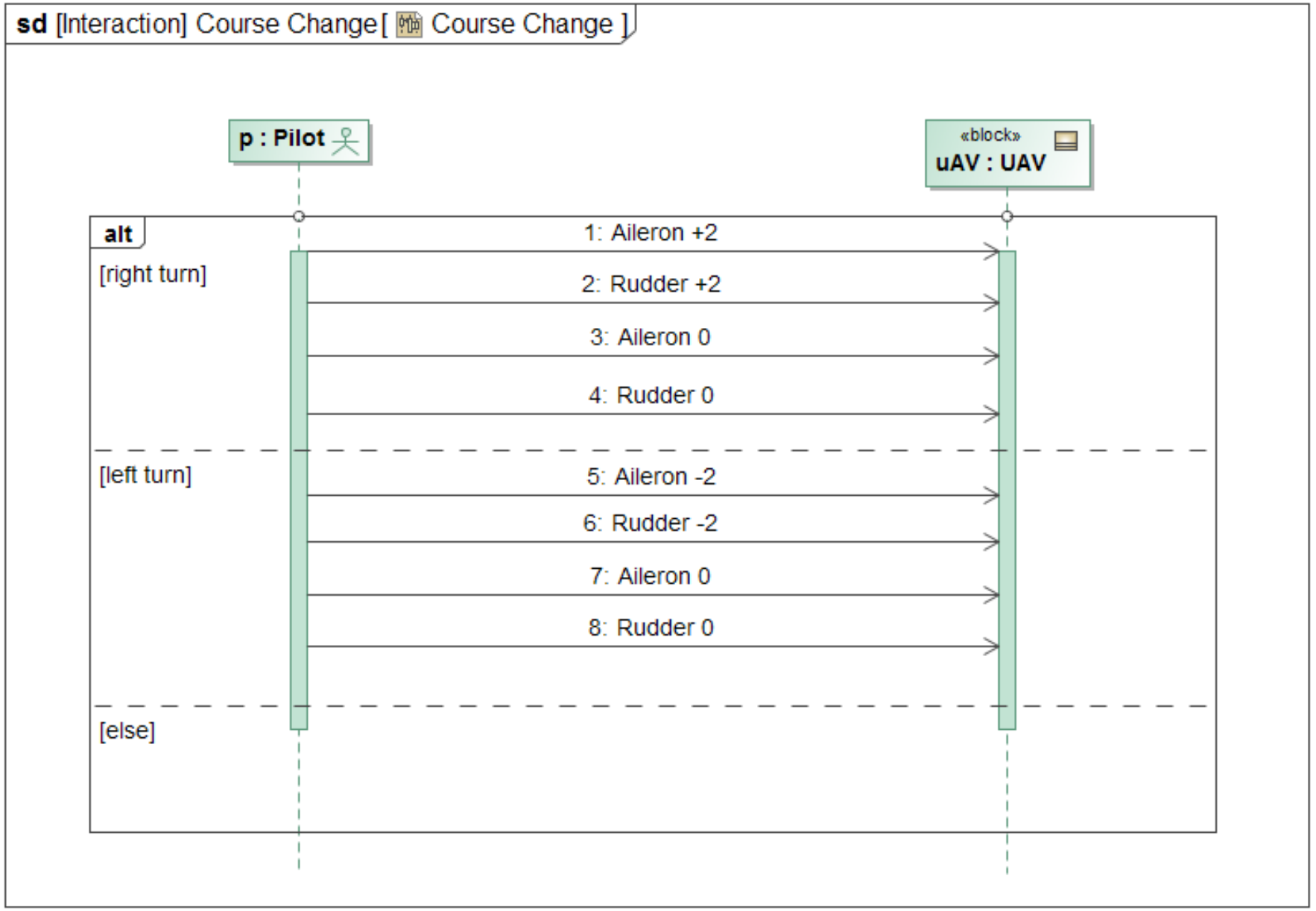
- A Combined Fragment consists of an interaction operator and a frame containing one or more messages
- Four important ones are Loop, Opt, Alt, and Par
- Using a combination of fragments, including nesting, very complex multibranch scenarios can be modeled

Combined Fragments

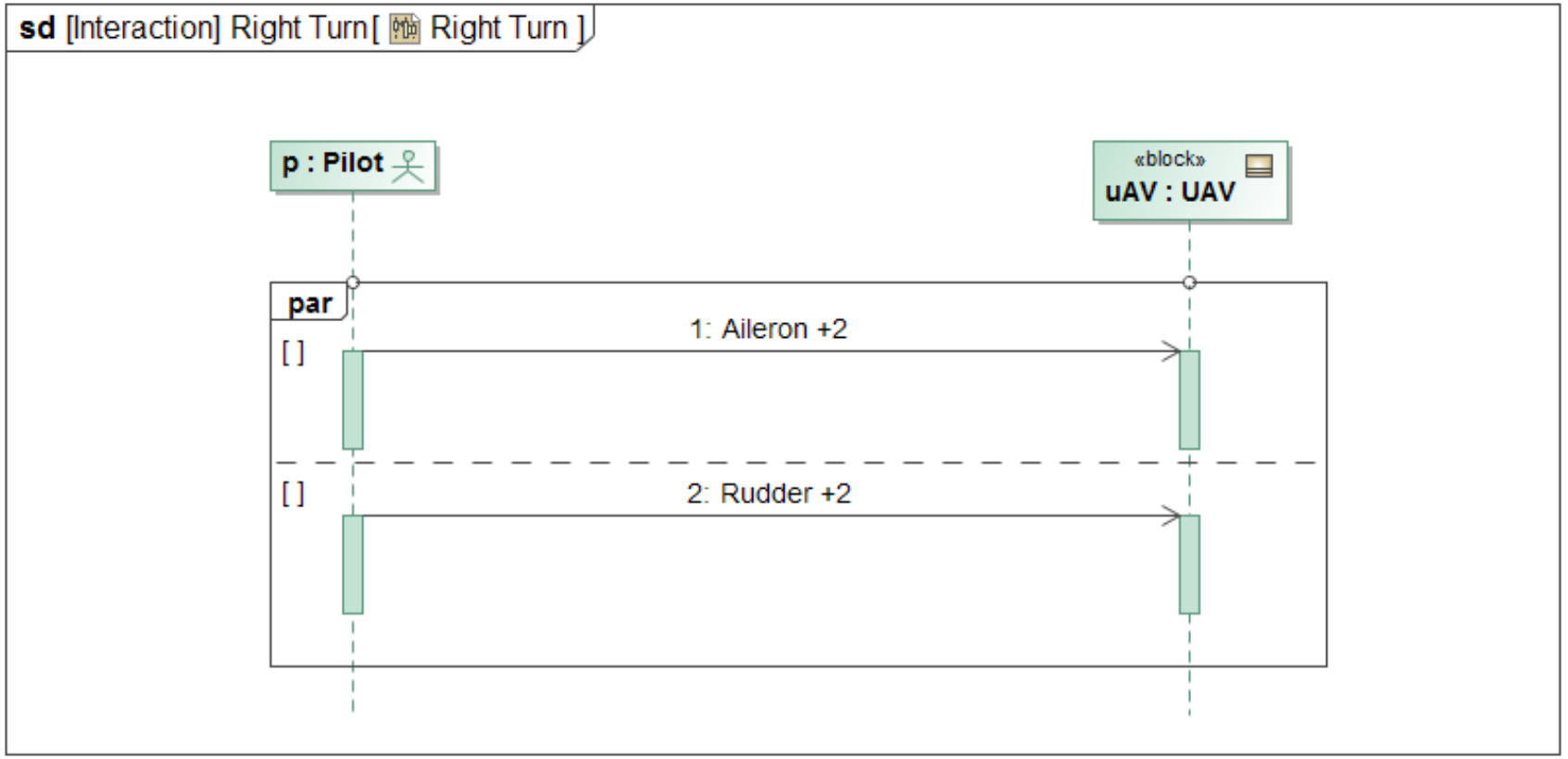


Combined Fragments



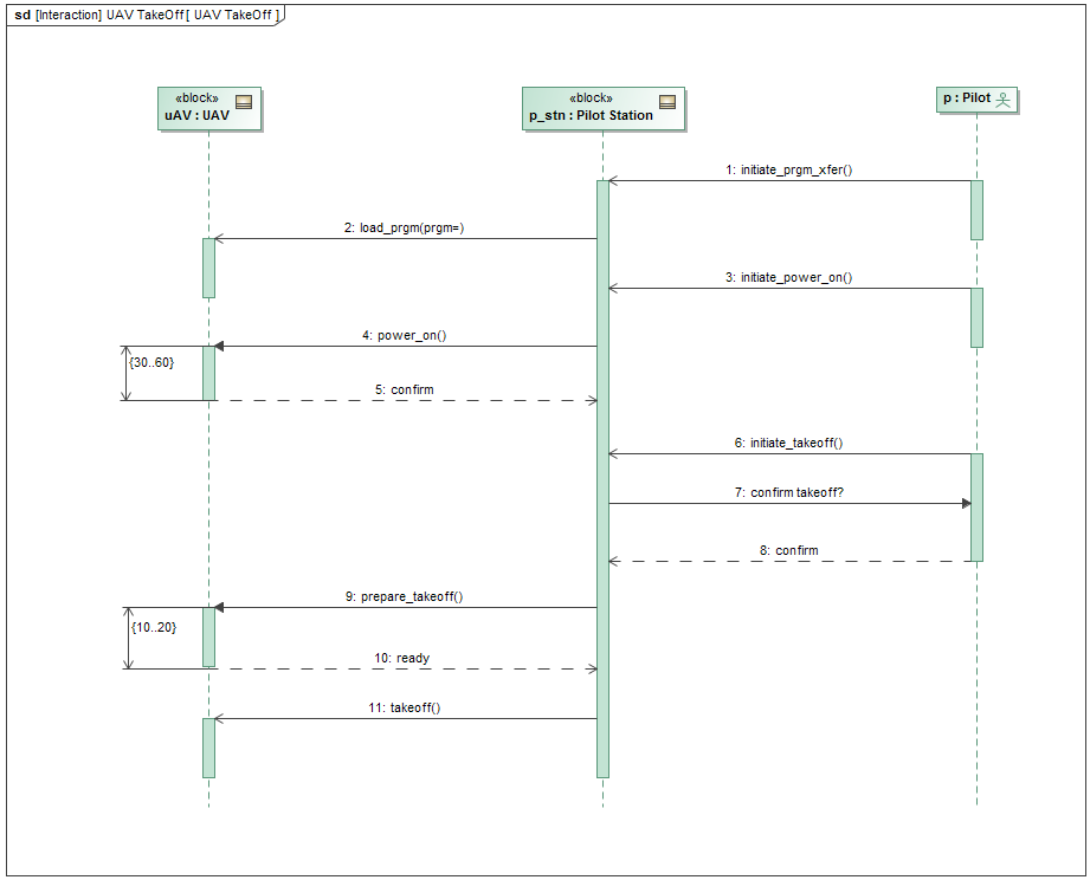


Combined Fragments



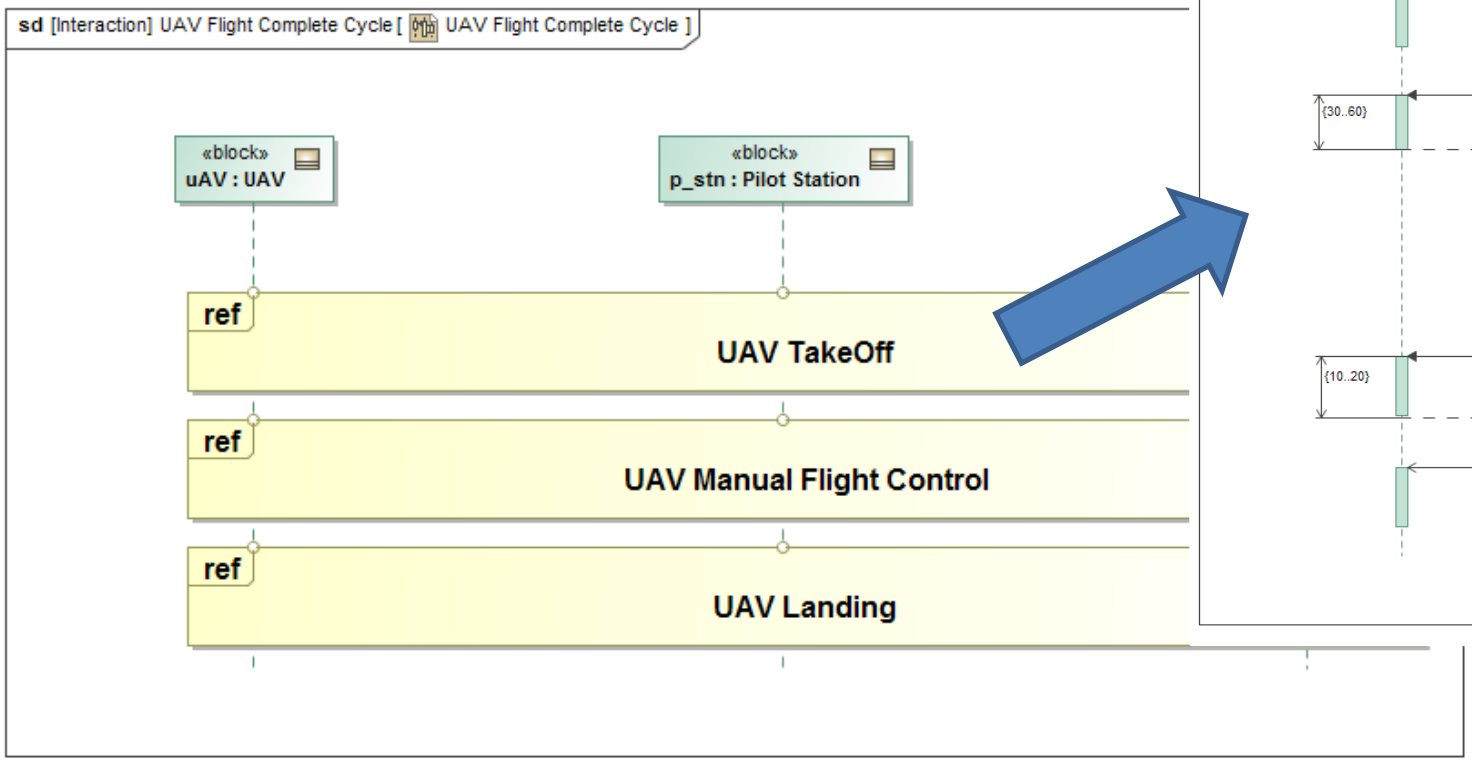
Interaction Uses

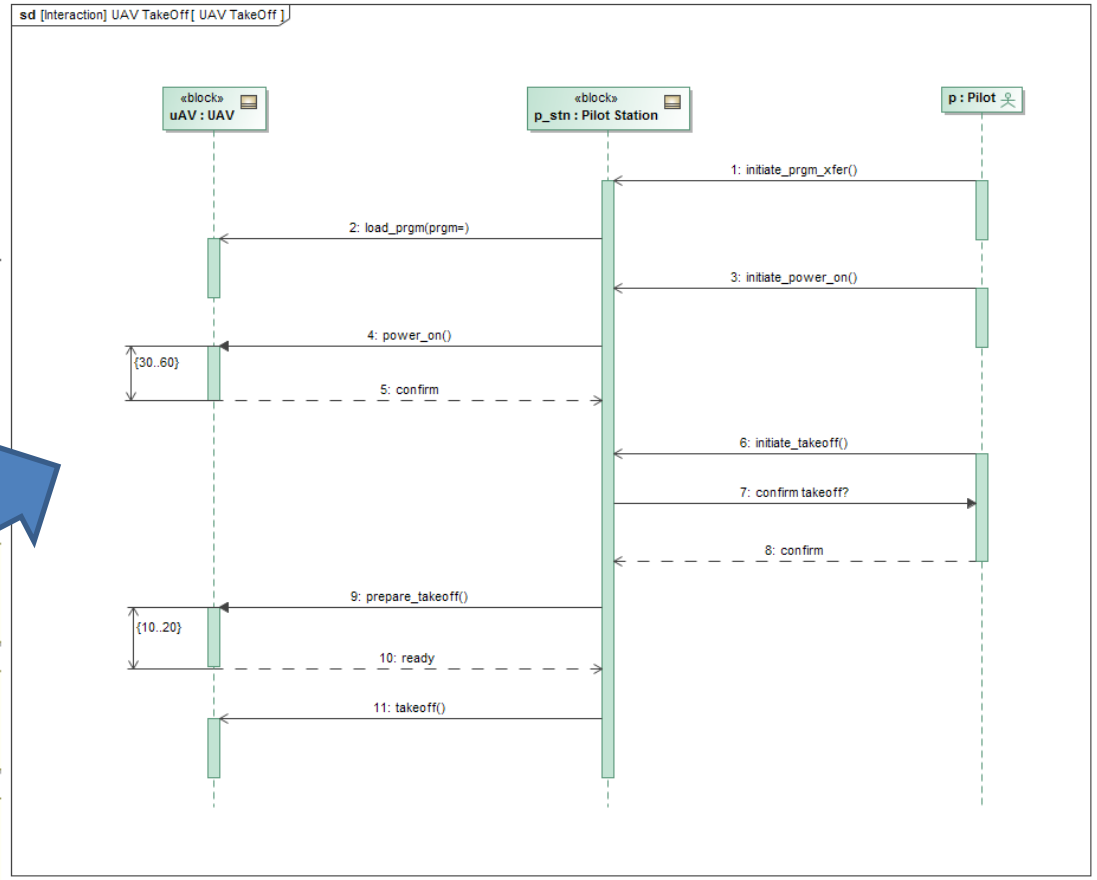
- Interaction Uses reference a lower-level interaction in a higher-level sequence diagram



Interaction Uses

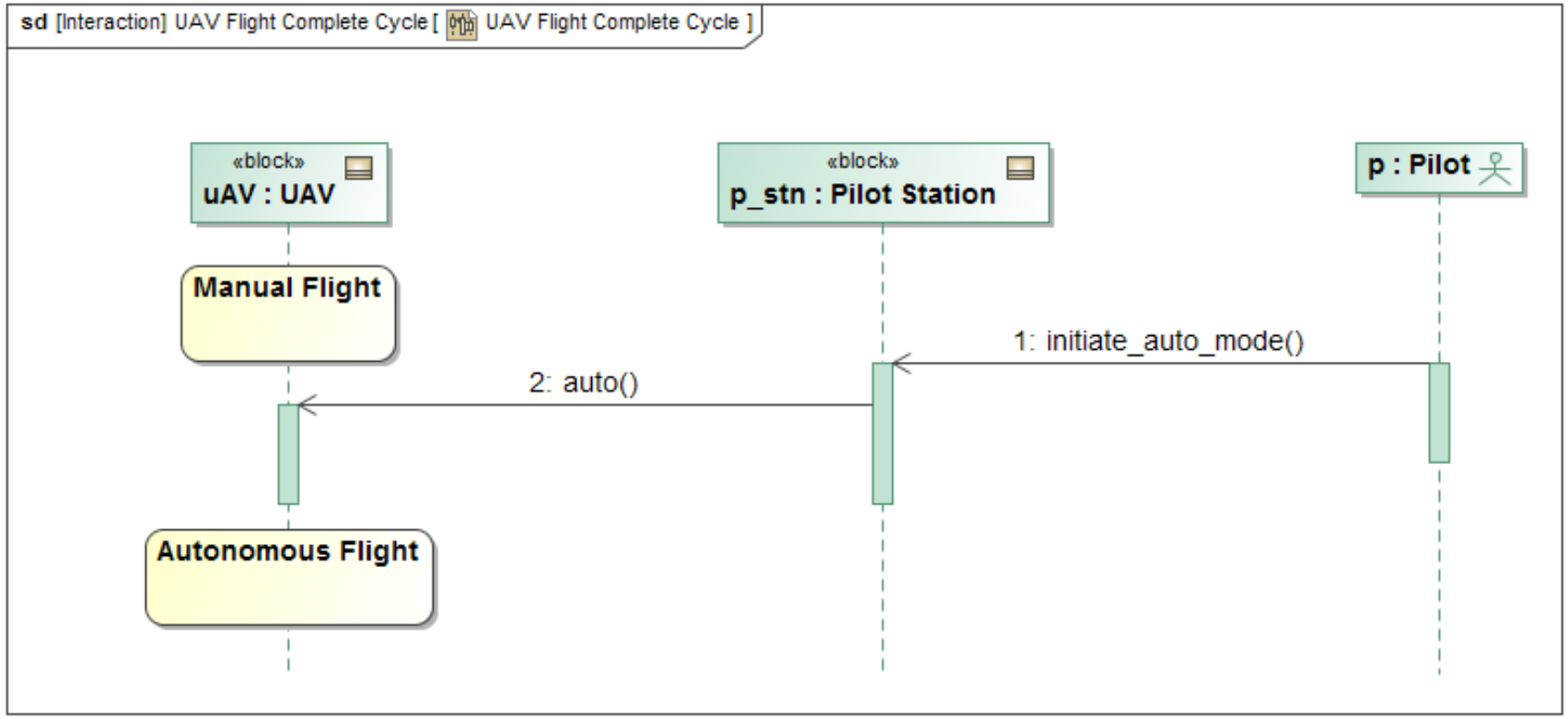
- Interaction Uses reference a lower-level interaction in a h





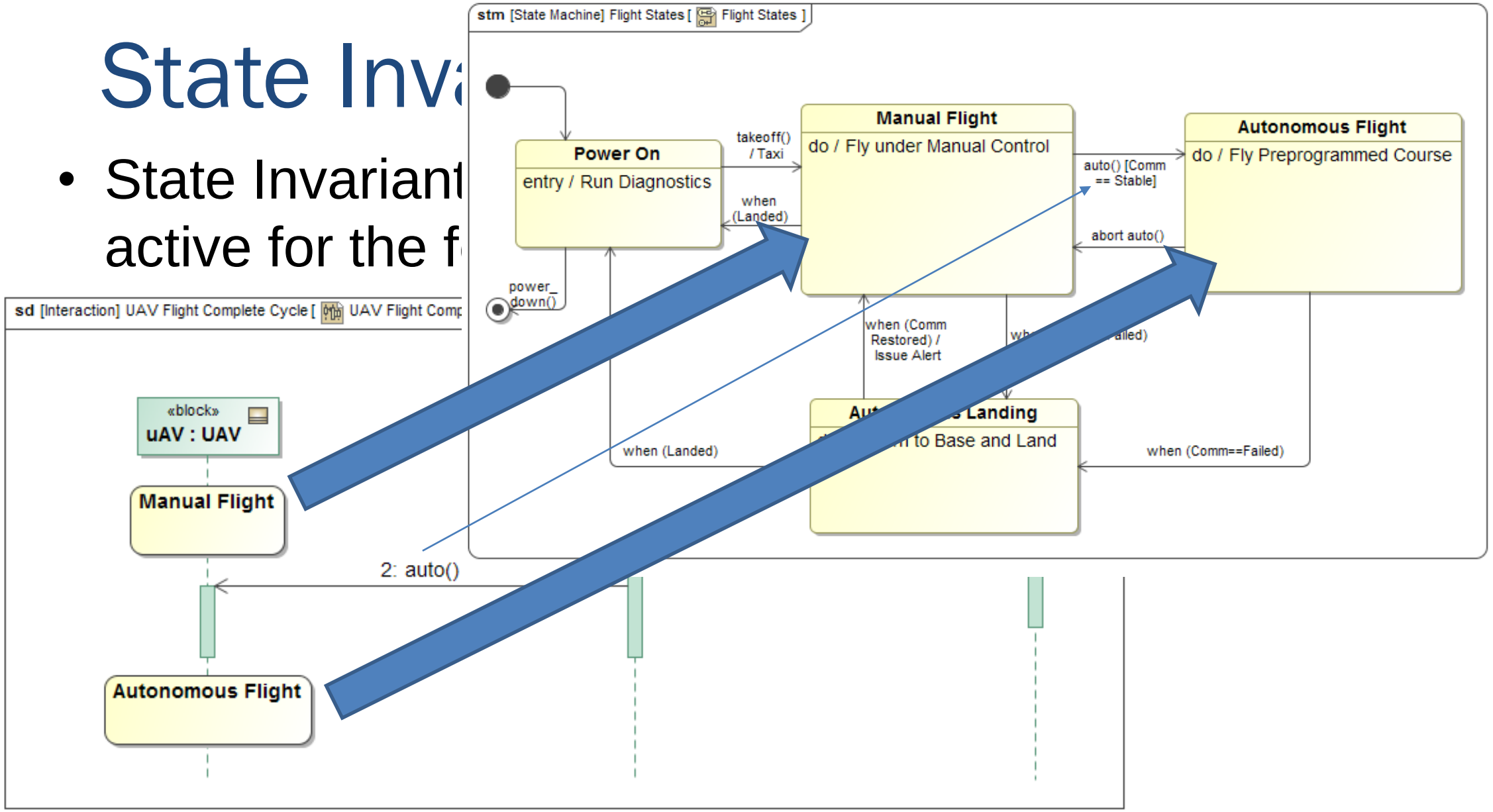
State Invariants

- State Invariants show the State that must be active for the following messages to be valid

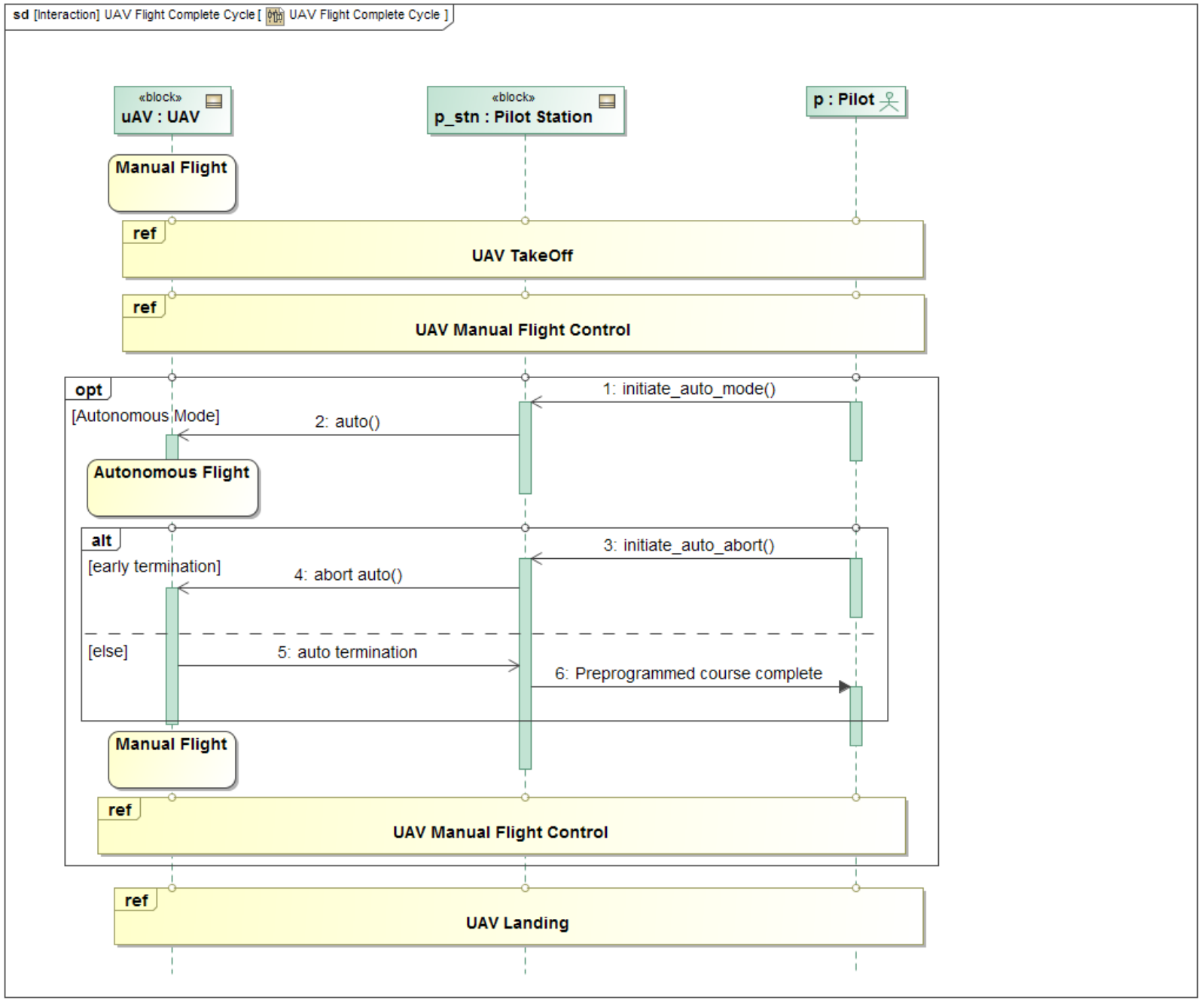


State Inva

- State Invariant
  active for the f



Seco



Recap

- At the end of the hands-on exercises, you should be able to
  - Explain the following terms: state, transition, pseudostate, combined fragment, interaction use, submachine state, trigger, guard, effect
  - Create a state machine diagram
  - Create combined fragments and interaction uses in a sequence diagram
  - Identify the principle purpose(s) of state machine diagrams

Questions?

www.intercax.com

email info@intercax.com

blog www.intercax.com/blog

Twitter @intercax | LinkedIn intercax-llc

Disclosure

- This course material is developed by InterCAX LLC. All copyrights held by InterCAX LLC.
- Usage of this material is governed by the licensing agreement terms between user and InterCAX. Usage or distribution for educational or other purposes outside these terms is not permitted. No part of this material may be reproduced or repurposed without the written consent of InterCAX.
- The intent is to present vendor-independent concepts and examples in an objective educational way that the course participants will find helpful. References are made to commercial products by InterCAX for the purpose of making these concepts concrete. Course participants are responsible to evaluate these products and tools for themselves and to investigate similar products and tools by other organizations where applicable.