Modeling Message-Based Behavior with Interactions (Part 1 – SysML Concepts)

Section Objectives

- In this Section, you will learn:
  - How to model Sequence Diagrams in SysML

Overview

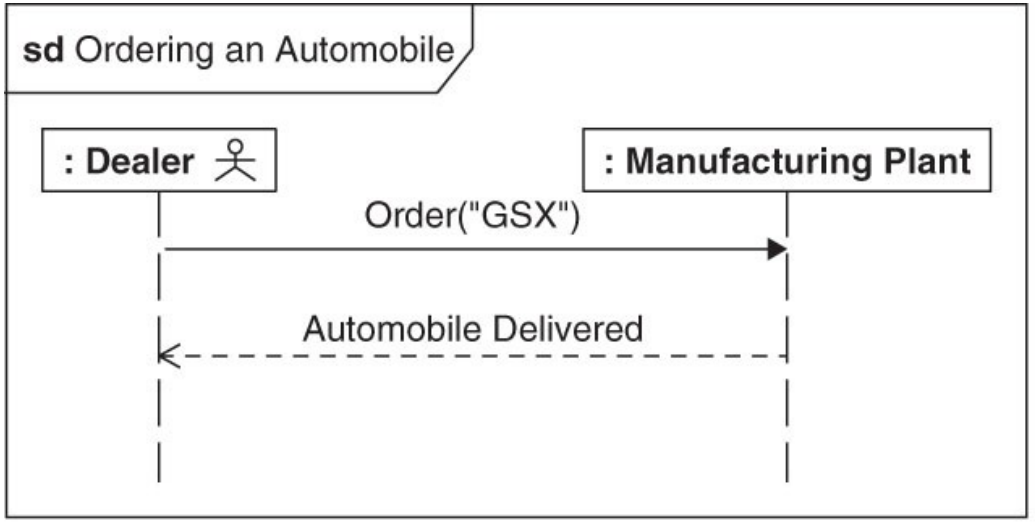
- This section will discuss:
  - Sequence Diagram Concepts
    - Why model Sequence Diagrams?
    - Sequence Diagram Components
    - Depicting Complex Interactions
    - Referencing Interactions
  - Sequence Diagrams for In-Class Project

Why Model Sequence Diagrams?

- Depicts the interactions between structural elements of a block
- Used to elaborate Use Cases
  - Depicts all of the paths (or scenarios) that a use case may take
  - Graphical depiction of the sequence of message exchanges
- Particularly useful in modeling Service-oriented processes
  - Where parts request services from other parts of the system
- Clarification, Elaboration, Communication

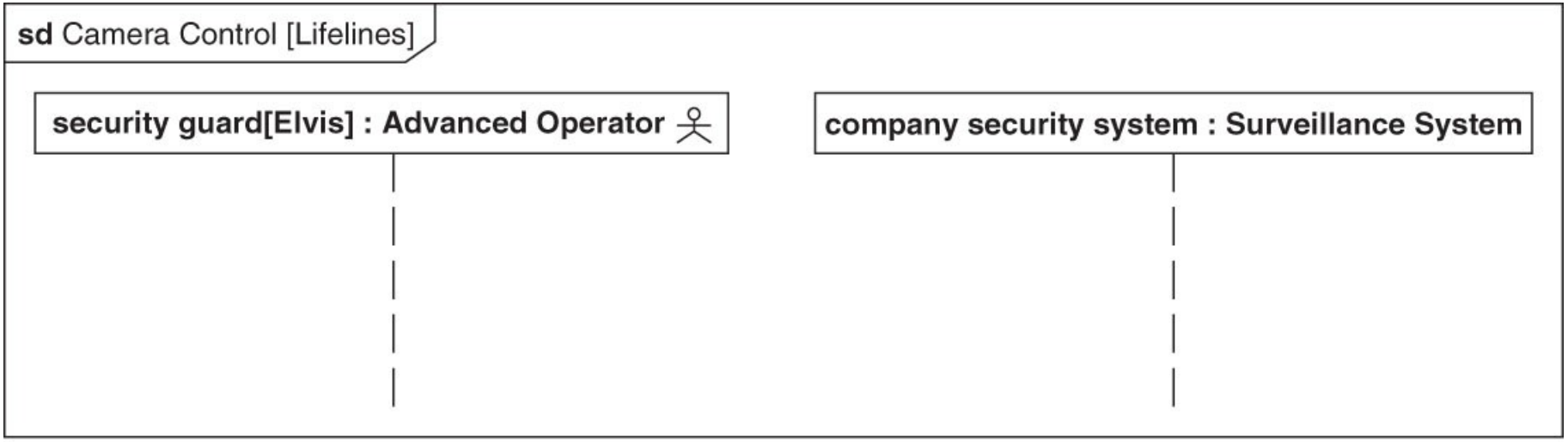
Sequence Diagram Components

- Sequence diagrams can be comprised of the following:
  - Lifelines
    - Represents a Structural Element of a system
    - Depicts 'Time'
  - Messages
    - Asynchronous
    - Synchronous
    - Reply



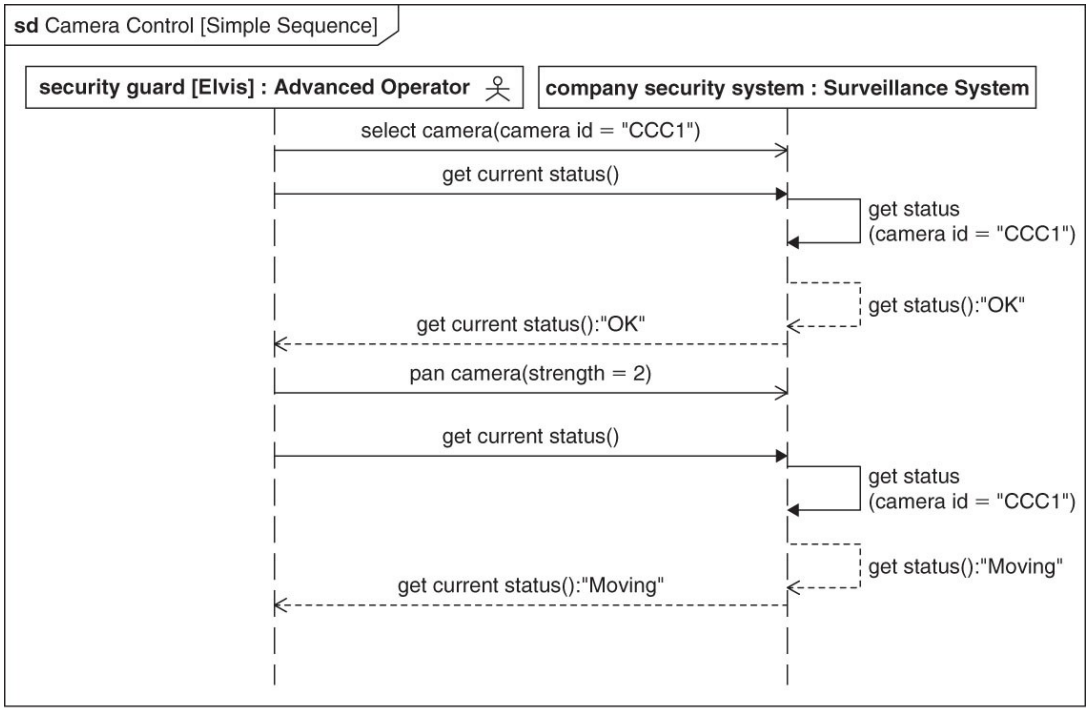
Lifelines

- Represents a Structural Element of a system
  - Depicted by a rectangle, with the name and type of the element
- Depicts 'Time'
  - Depicted by a dashed line descending from the base of the rectangle
  - Time 'advances' from top to bottom



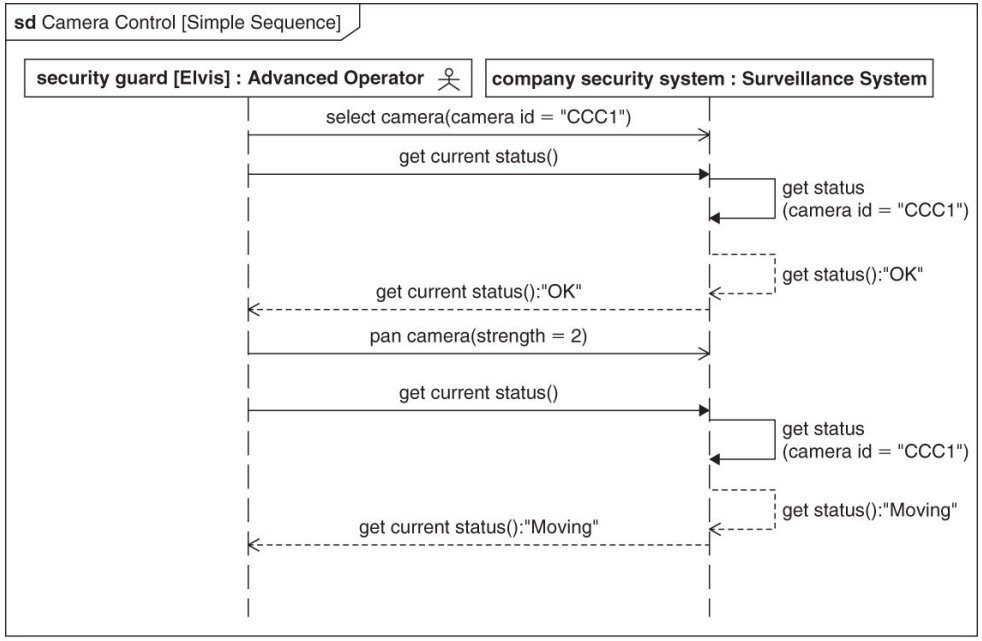
Asynchronous Messages

- Asynchronous - sender continues to execute after sending message
  - Used when sending a signal or for an asynchronous call for an operation
  - Represented by a solid line with an open arrowhead



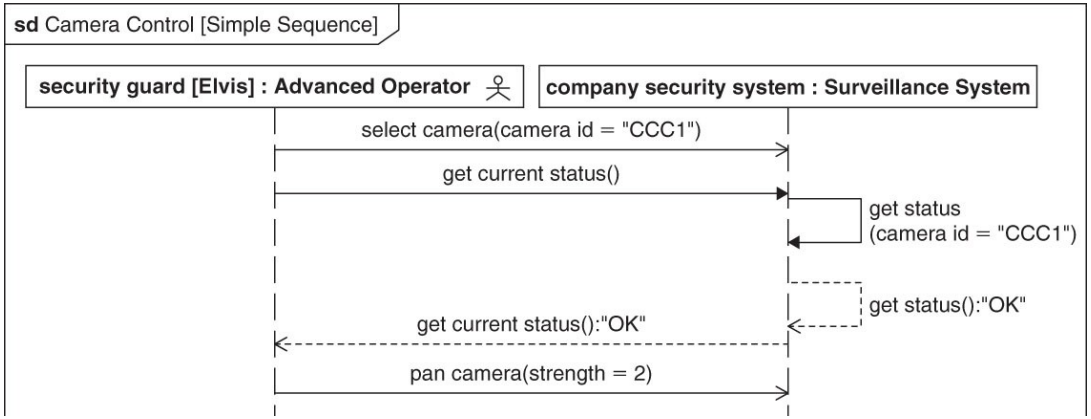
Synchronous and Reply Messages

- Synchronous – sender waits for a response
  - Used for a synchronous call for an operation
  - Represented by a solid line with a closed (solid) arrowhead
- Reply – used to depict a response to a synchronous message
  - Represented by a dashed line with an open arrowhead (optional)



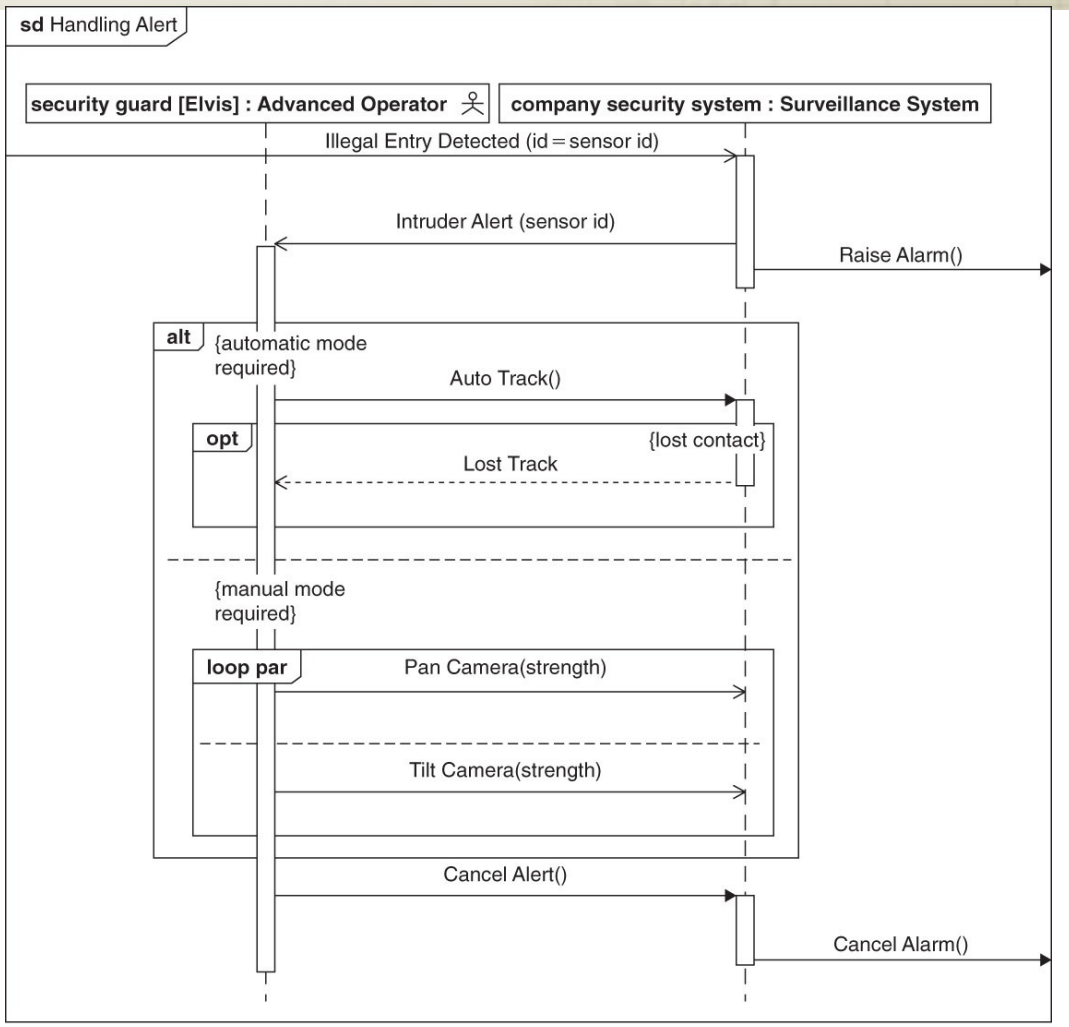
More on Messages

- Messages usually represent a ‘call’ for an operation
  - Depicts a request from the sender for the receiver to carry out one of the receiver's operations
- Messages are ‘labeled’ with the requested operation name
  - May include parameters – contained within parentheses ( )
  - May include return values – listed after parameter list
  - May include conditions – contained within brackets [ ], and listed before the operation name



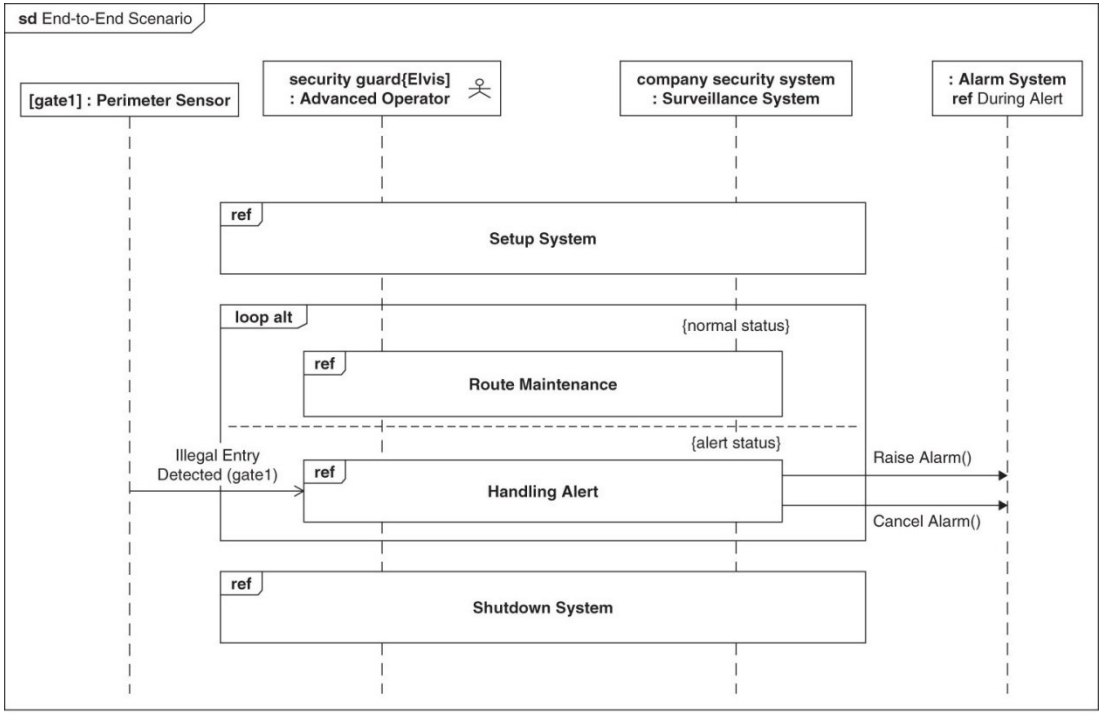
Depicting Complex Interactions

- Complex interactions can be depicted in Sequence Diagrams through use of Combined Fragments
- Combined Fragments allow logic to be depicted in Sequence Diagrams in order to show alternative (alt) paths, optional (opt) paths, parallel paths (par), or loops (loop)
- Fragments are depicted as frames in a SD and contain an Interaction Operator to define the type of logic to apply (e.g. alt, opt, par, loop)



Referencing Interactions

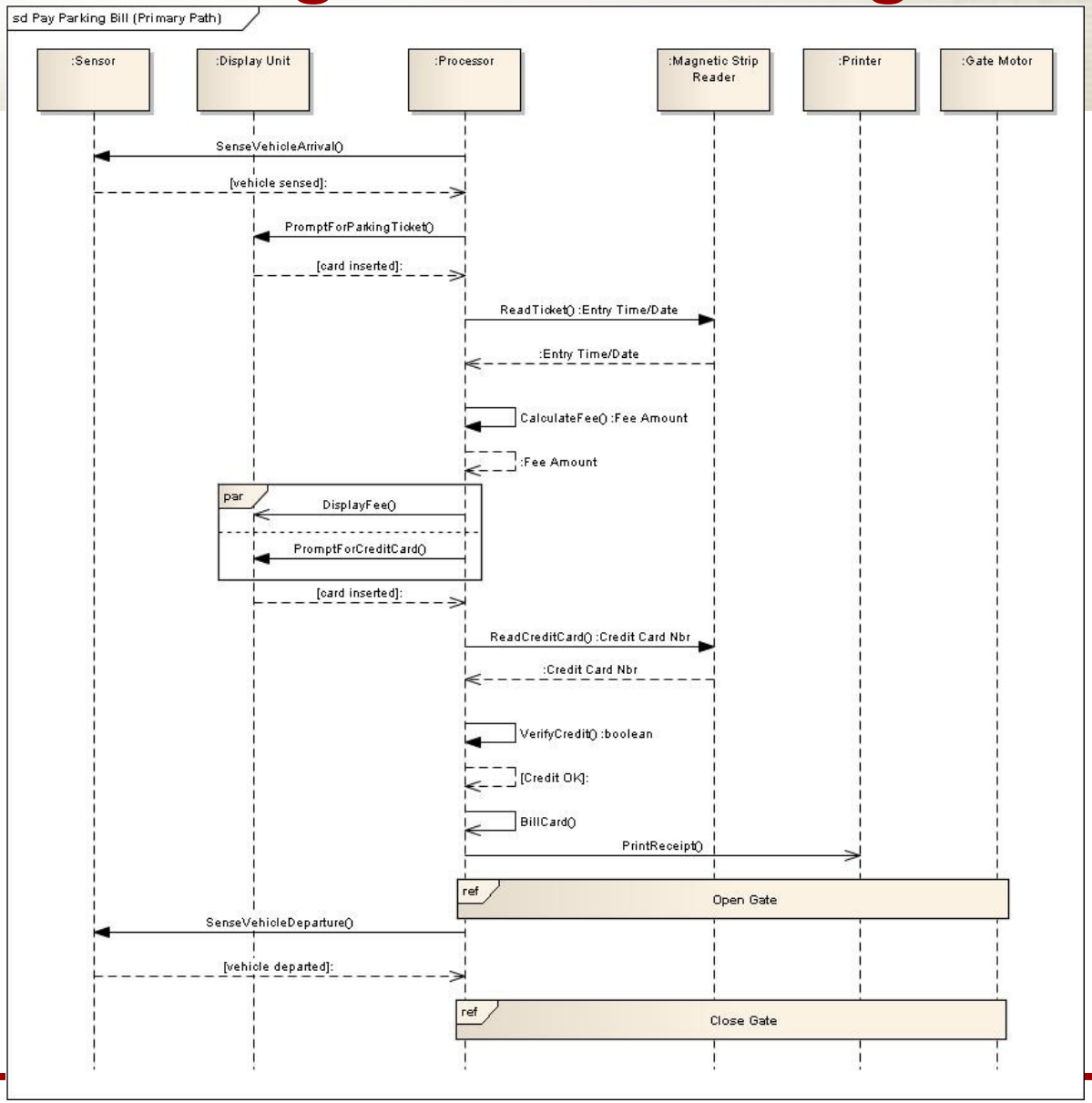
- Interactions can be referenced on Sequence Diagrams
  - Depicted as a frame with the keyword 'ref'
  - Body of the frame contains the name of the referenced interaction
  - Supports re-use of common interactions



Sequence Diagram for In-Class Project

- Build a Sequence Diagram for Parking Garage Gate Project using EA
- Define
  - Lifelines
  - Messages

Sequence Diagram for Parking Garage Gate



Summary

- Sequence Diagrams are used to depict the interactions between structural elements of a Block
- Sequence Diagrams are comprised of:
  - Lifelines
  - Messages
- Lifelines represent the structural element and depicts Time
- Messages can be either:
  - Asynchronous
  - Synchronous
  - Reply
- Messages represent a call for an operation
- Combined Fragments are used to depict complex interactions and include: alternate paths, parallel paths, optional paths or loops
- Reference Interactions depict re-use of common interactions