SysML Behavioural Diagrams

Sequence Diagrams

Introduction to Systems Engineering
I2ISE

SysML Stucture vs. behaviour

- We have learned a lot about how to model *structure* in SysML
  - Block Definition Diagrams
  - Internal Block Diagrams

- Now, we will look at how we can model *behaviour* in SysML
  - Sequence diagrams
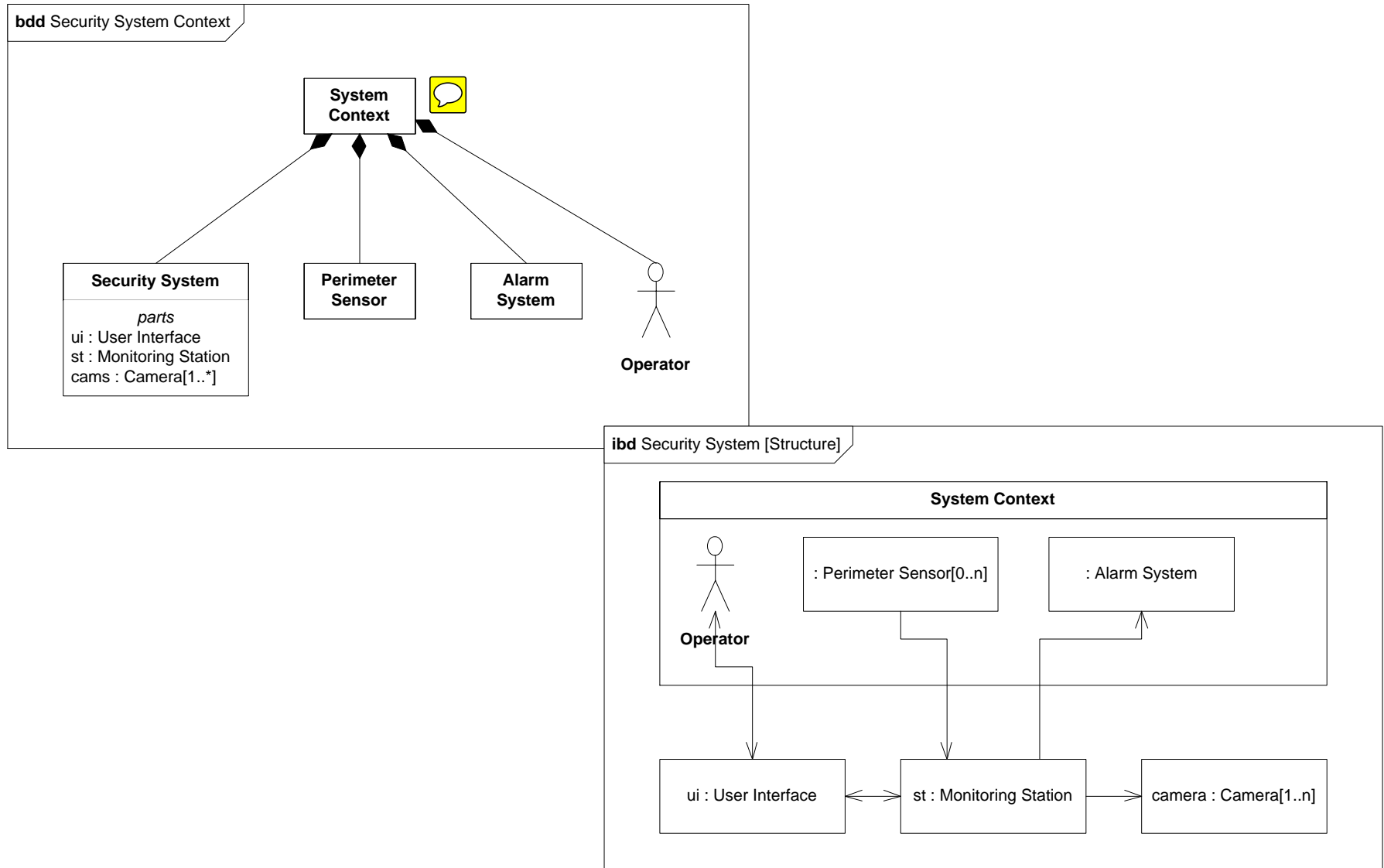 - State Machines

Sequence diagrams

- Sequence diagrams (diagram type *sd*) model interactions between *parts* of a block
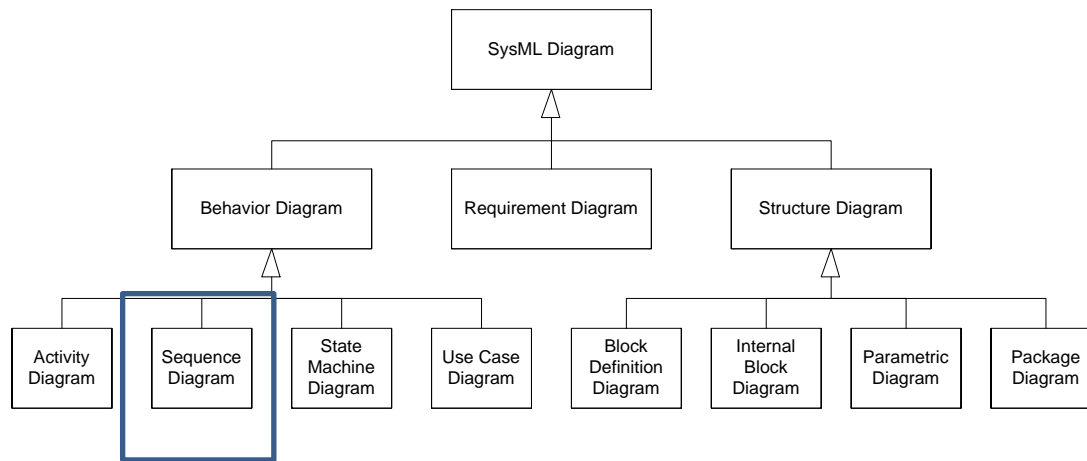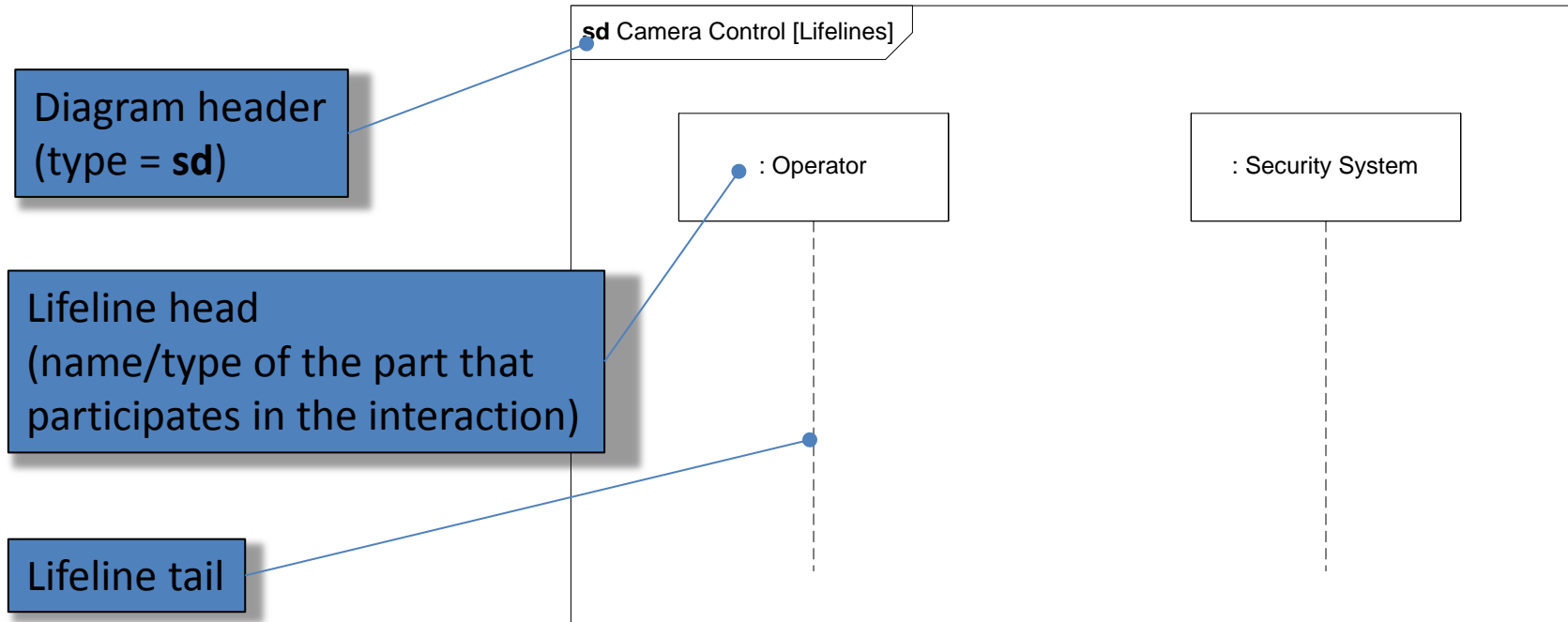
Sequence diagrams

- Sequence diagrams are used to model *message*-based behaviour

- The interactions take place within a block between its elements of internal structure (parts)

- The basic diagram consists of *lifelines* with *messages* between them.
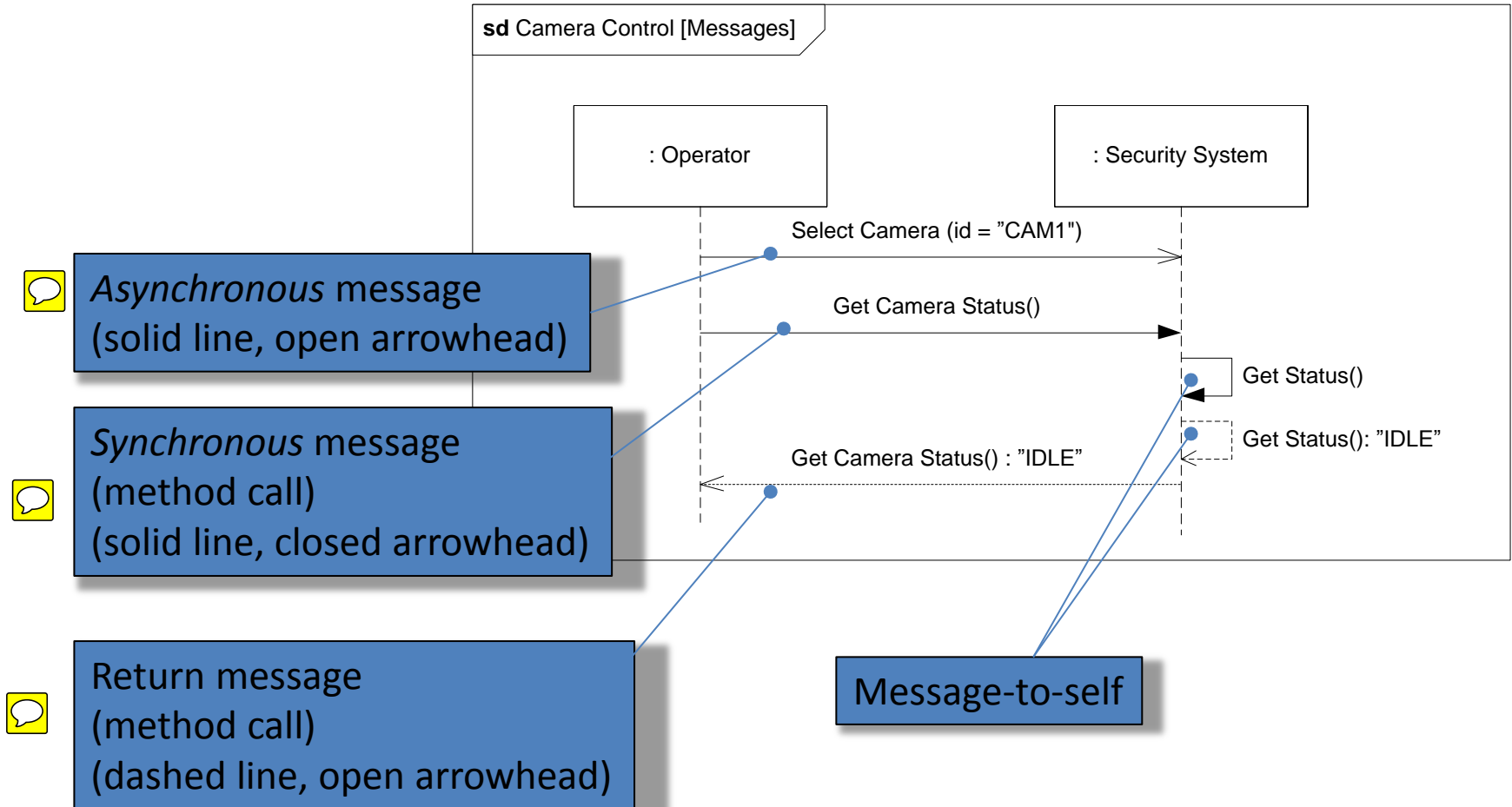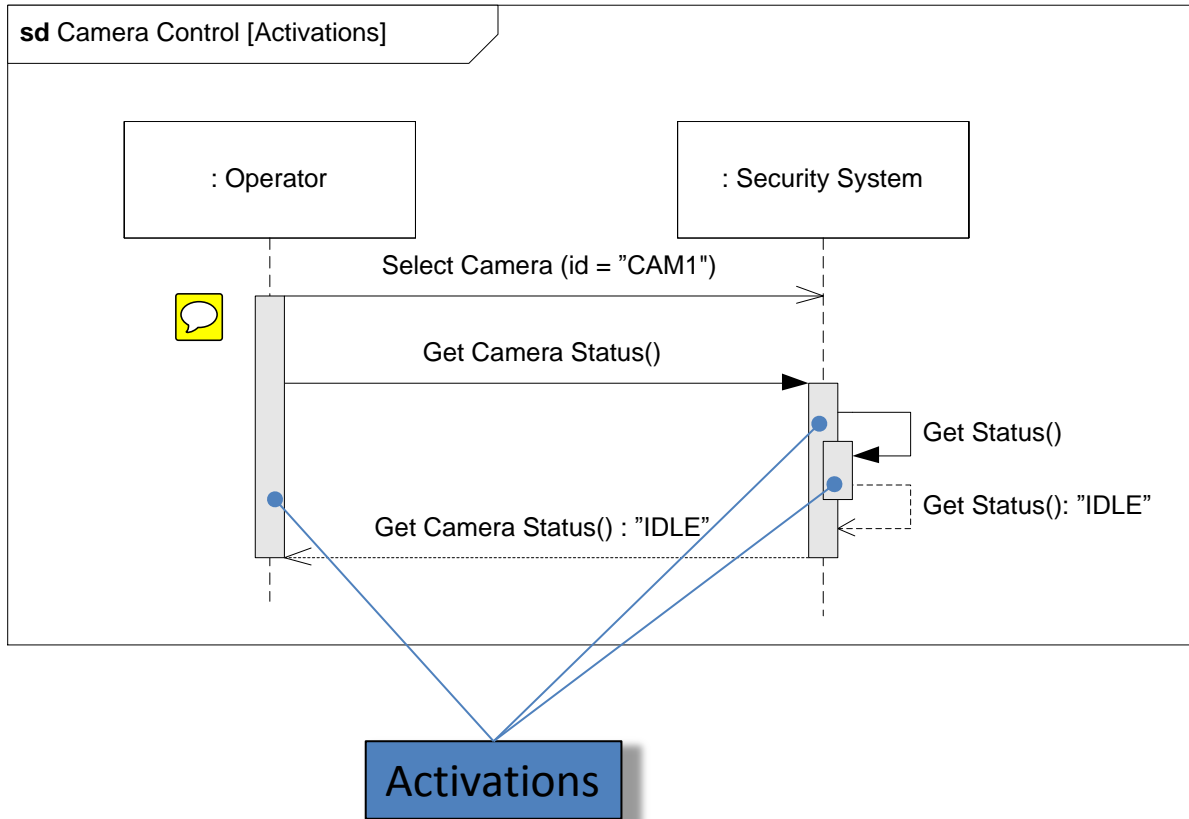
SD's – example system (structure)

SD's – lifelines

Diagram header
(type = **sd**)

Lifeline head
(name/type of the part that
participates in the interaction)

Lifeline tail

sd Camera Control [Lifelines]

: Operator

: Security System

SD's – messages

sd Camera Control [Messages]

: Operator

: Security System

Select Camera (id = "CAM1")

Asynchronous message
(solid line, open arrowhead)

Get Camera Status()

Get Status()

Synchronous message
(method call)
(solid line, closed arrowhead)

Get Status(): "IDLE"

Get Camera Status() : "IDLE"

Return message
(method call)
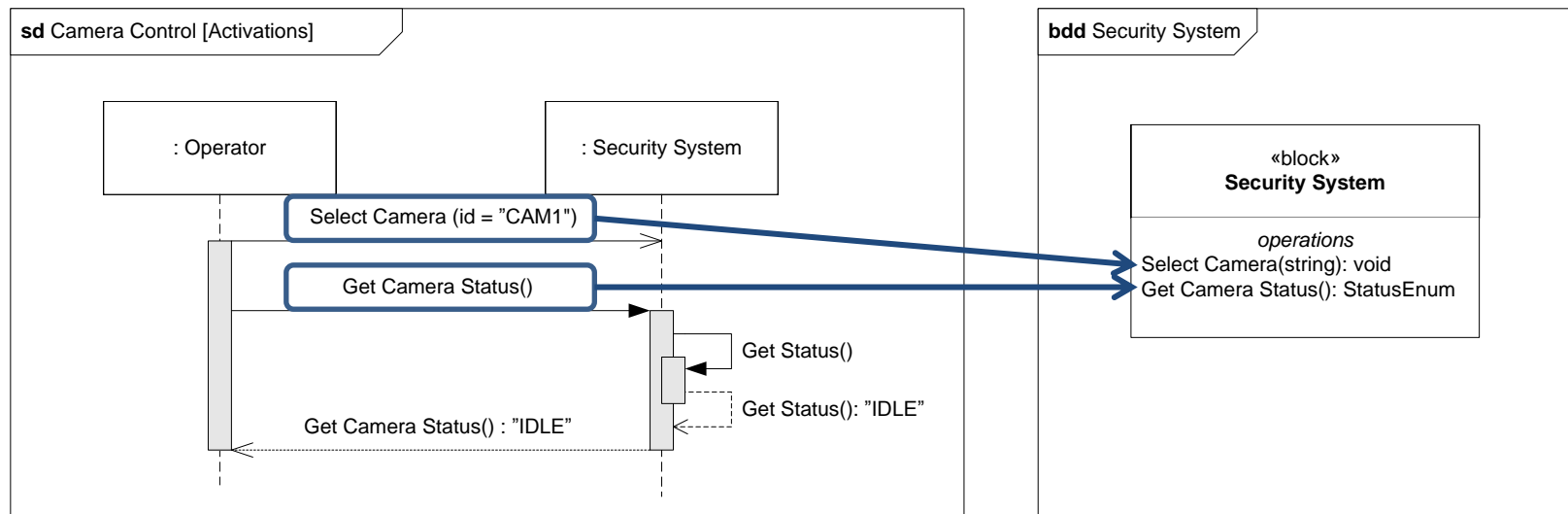(dashed line, open arrowhead)

Message-to-self

SD's – activations
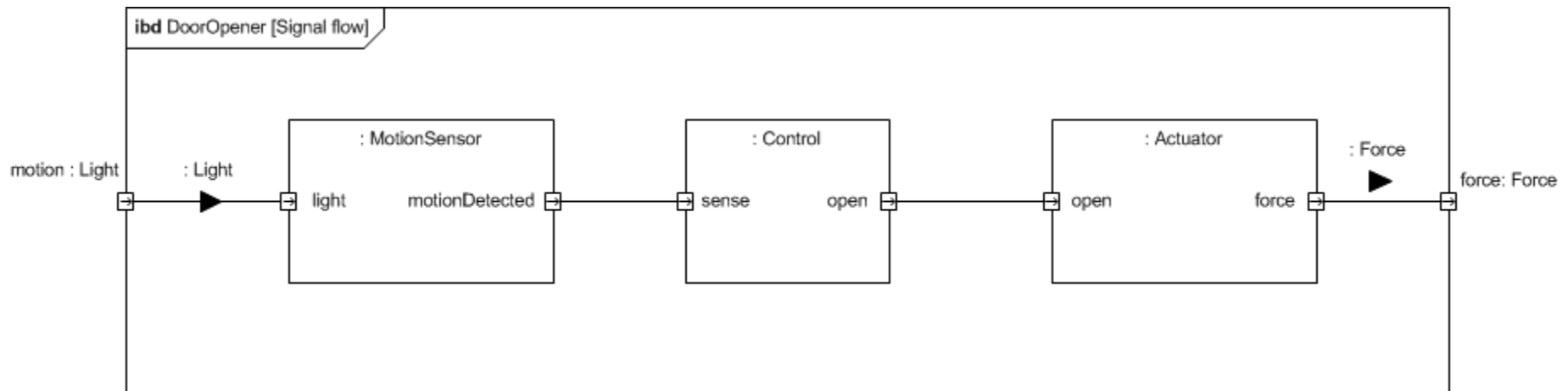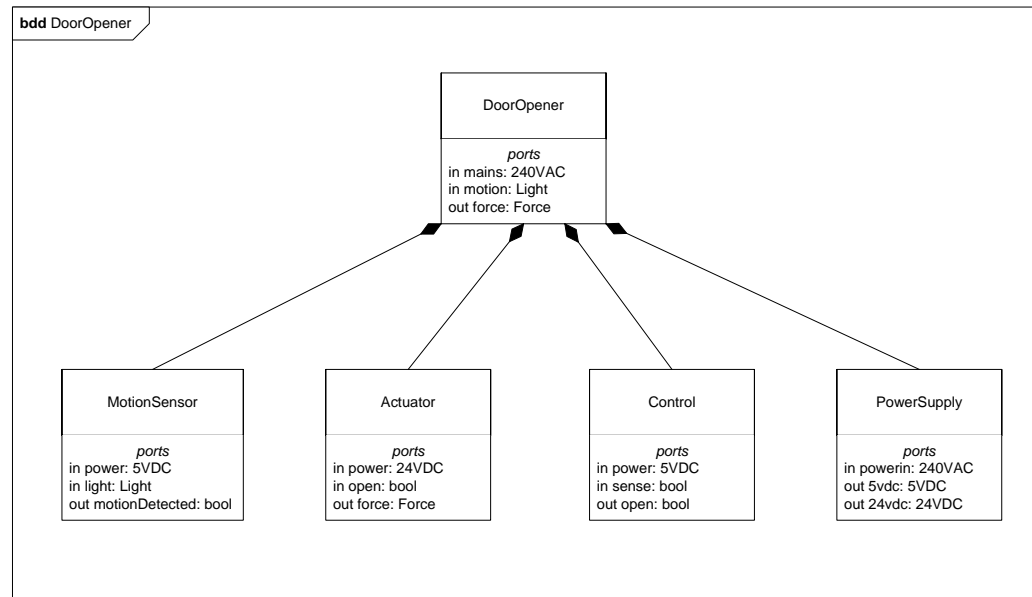
Structure and behaviour – a link

- The behaviour (operations) used in an SD are found in the operations compartment of the enclosing block
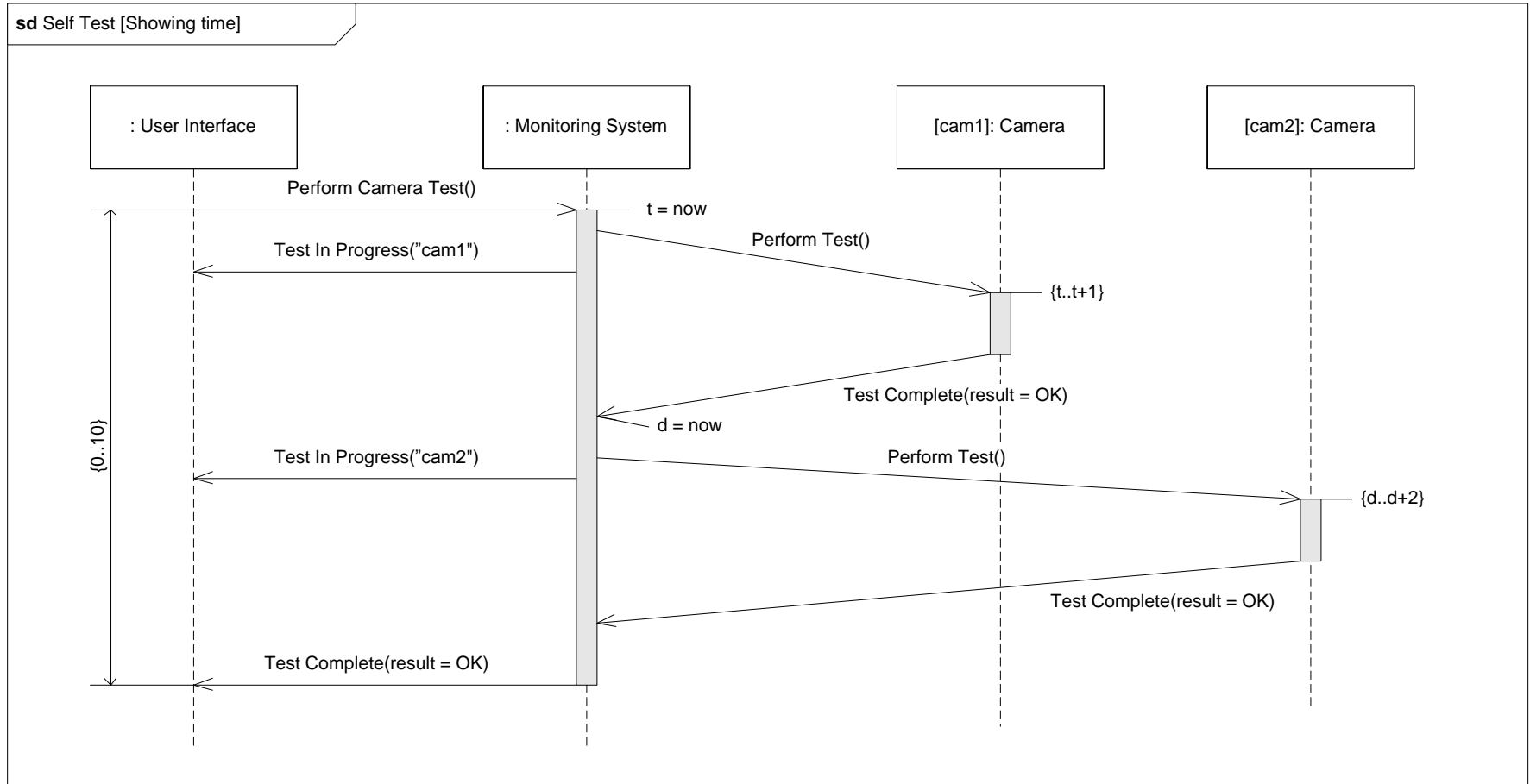
SD's – your turn!

- Draw an SD for the DoorOpener system
- Add operations to blocks

SD's – representing time

sd Self Test [Showing time]

: User Interface   : Monitoring System   [cam1]: Camera   [cam2]: Camera

Perform Camera Test()

t = now

Perform Test()

Test In Progress("cam1")

{t..t+1}

Test Complete(result = OK)

d = now

Test In Progress("cam2")

Perform Test()

{d..d+2}

Test Complete(result = OK)

Test Complete(result = OK)

{0..10}

SD's – fragments
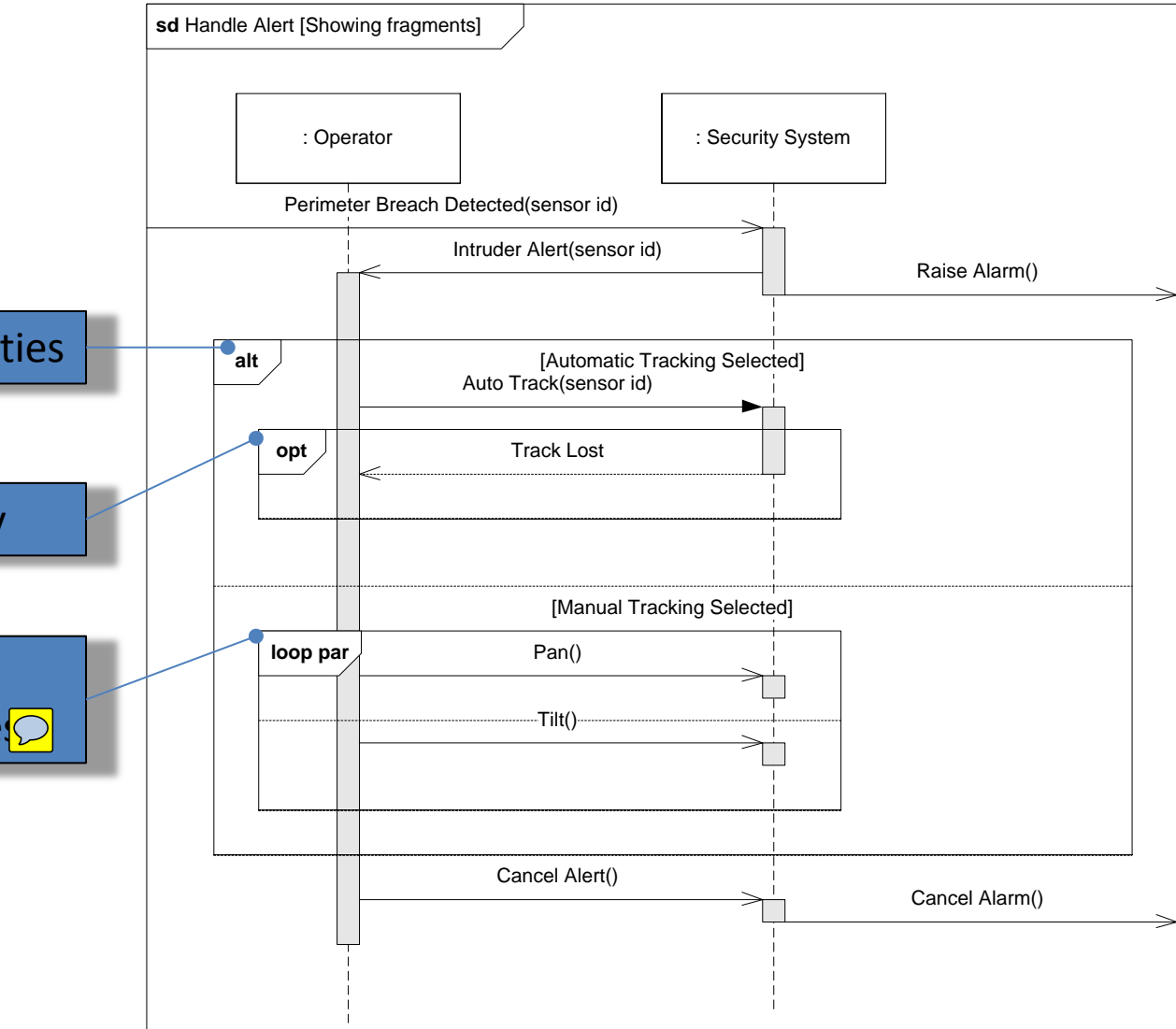
SD's – reference blocks
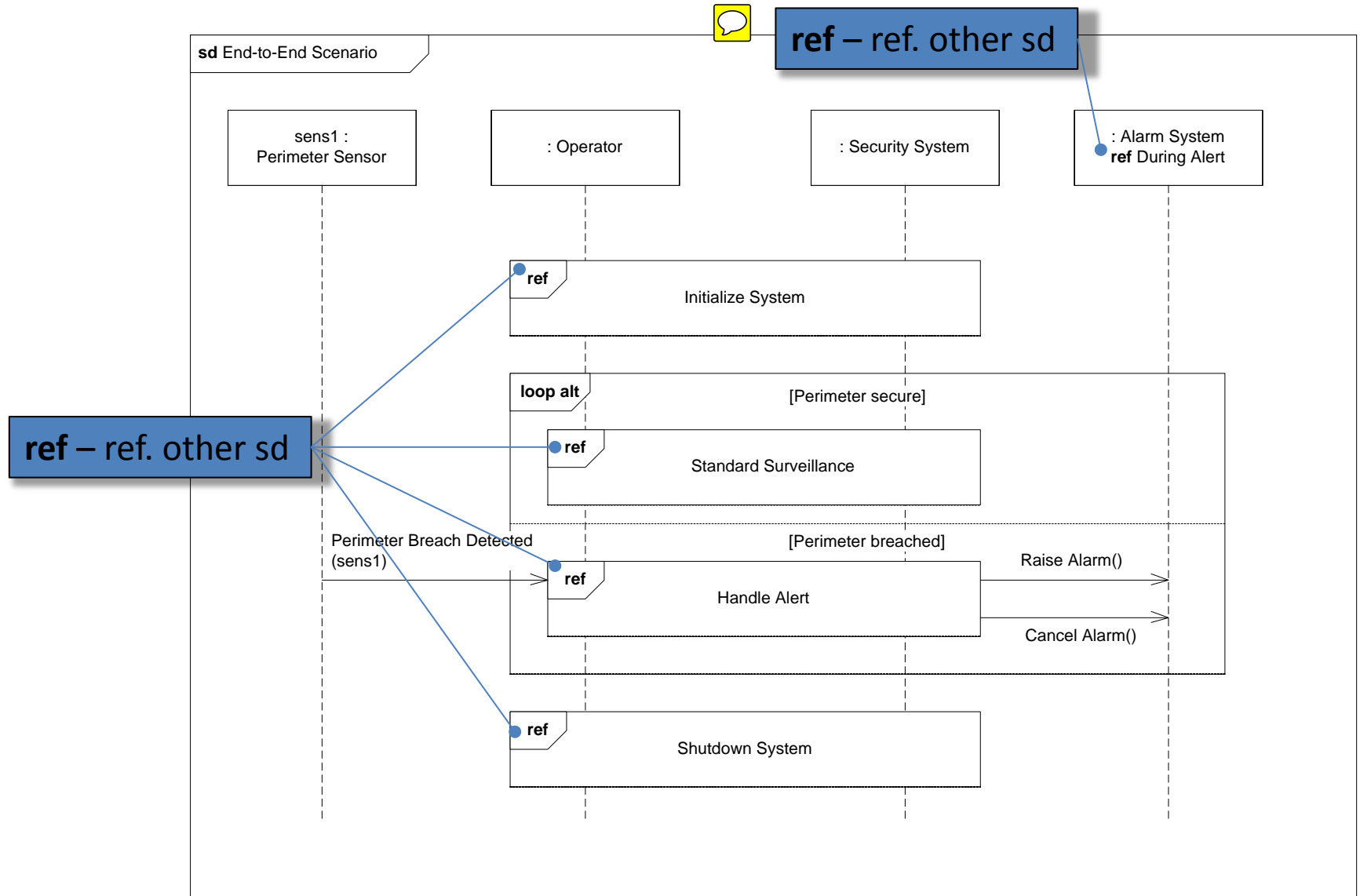
SD's – your turn!

- Create a sequence diagram for the RVM scenario *Recycle Containers* below
  - Participants: *User* and *RVM*
- Add operations to the RVM on a BDD

Main Scenario for Use Case *Recycle containers*
1. User arrives at RVM and is informed to insert containers.
2. User places container in the in-feed.
3. RVM scans container and *either*
   a) accepts the container, collects the container from the in-feed, adds the return deposit to the collected amount, and displays the type and value of the accepted container and the total collected amount; *or*
   b) does not accept the container, rejects the container to User, and displays that the container is not accepted and the total collected amount.

Step 2 through 3 is repeated until User is done feeding containers.
1. User request the return deposit receipt.
2. RVM prints out the return deposit receipt, and resets the collected amount.

SD's – your turn!