University of Buea
Faculty of Engineering and Technology
Department of Computer Engineering

CEF 331

OBJECT ORIENTED MODELING AND UNIFIED MODELING LANGUAGE (UML)

By I.R. DJOUELA KAMGANG Epse NOUABO; Dr. Eng.

dorlystevia08@gmail.com
ines.djouela@supcom.tn

Course outline

Course Objectives

Chapter 5: Sequence diagram

At the end of this chapter, you should be able to:

- Identify the main components of a sequence diagram
- Extract a sequence diagram from a use case scenario

Course outline

Chapter 5: sequence diagram

- Reminder: interaction diagrams
- Sequence diagram
- Sequence diagram different components
  - ✓ Participants
  - ✓ lifeline
  - ✓ messages
  - ✓ The axis
  - ✓ The fragments
- How to produce sequence diagrams

Interaction diagrams

- UML Specifies a number of interaction diagrams to model dynamic aspects of the system
- Interaction diagrams show the communication behavior between parts of the system
- Dynamic aspects of the system
  - ✓ Messages moving among objects/classes
  - ✓ Flow of control among objects
  - ✓ Sequences of events

Interaction diagrams

- Four types of diagrams:
- Sequence diagram: emphasis on the sequence of communications between parts
- Communication diagram: emphasis on structure and the communication paths between parts
- Timing diagram: emphasis on change in state over time
- Interaction overview diagram: emphasis on flow of control between interactions

Interaction diagrams

Sequences of events can be shown in two types of diagrams:

- Sequence diagram which focuses on the time sequences
- Communication diagram which focuses on the relationship among the object that exchange the messages

Sequence diagram

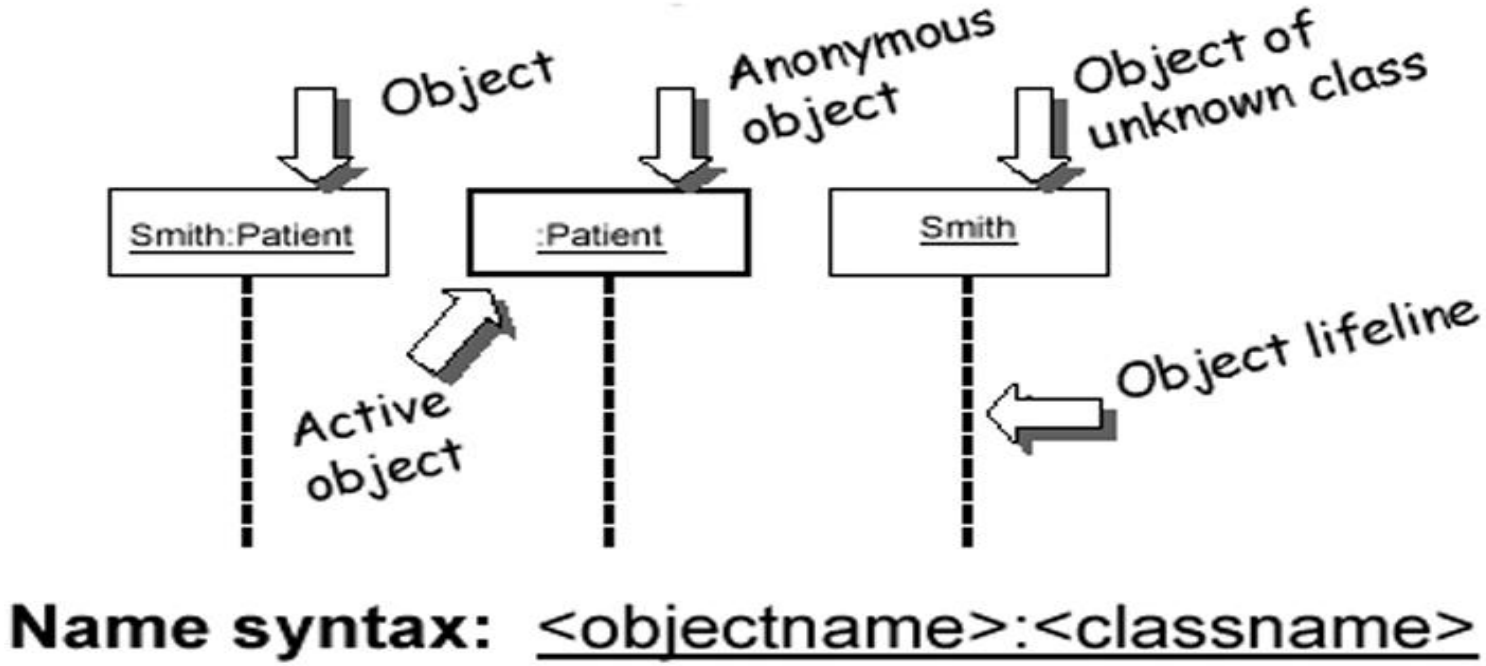
- Display object interactions arranged in time sequence
- Depict the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the system *over time*.
- A sequence diagram shows – for one particular scenario of a use case
  - ✓ the events that external actors generate,
  - ✓ their order, and inter-system events
- A sequence diagram is shown in a rectangular frame with the string sd name in a small pentagon in the upper left corner.

Sequence diagram components

- Consist of:
  - ✓ Participants
  - ✓ lifelines
  - ✓ Messages
  - ✓ The axis
  - ✓ The fragments

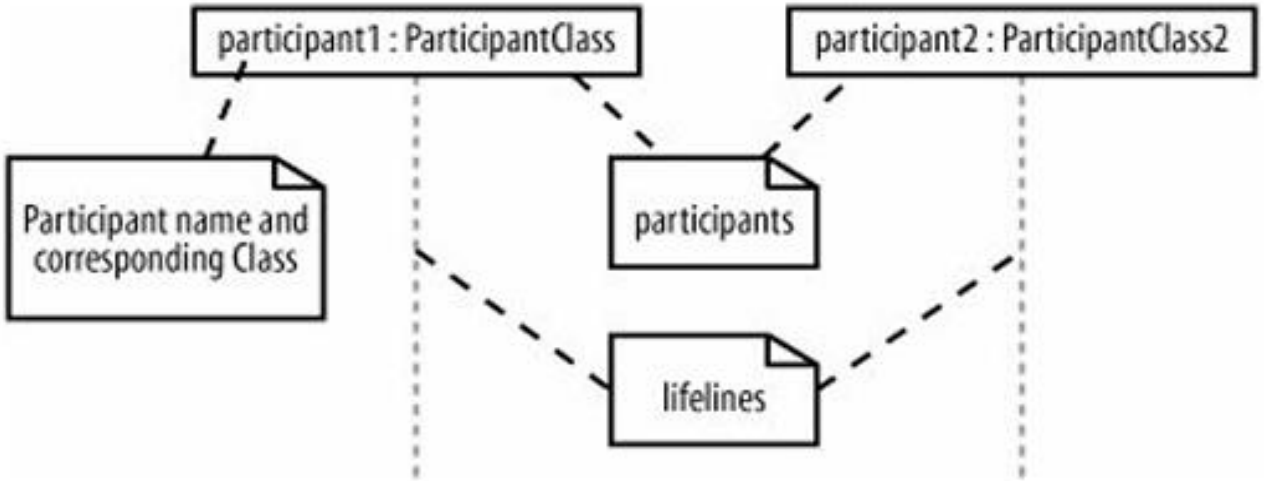
Participants

- Object or entity that acts in the diagram
- Participants are always arranged horizontally with no two participants overlapping each other
-



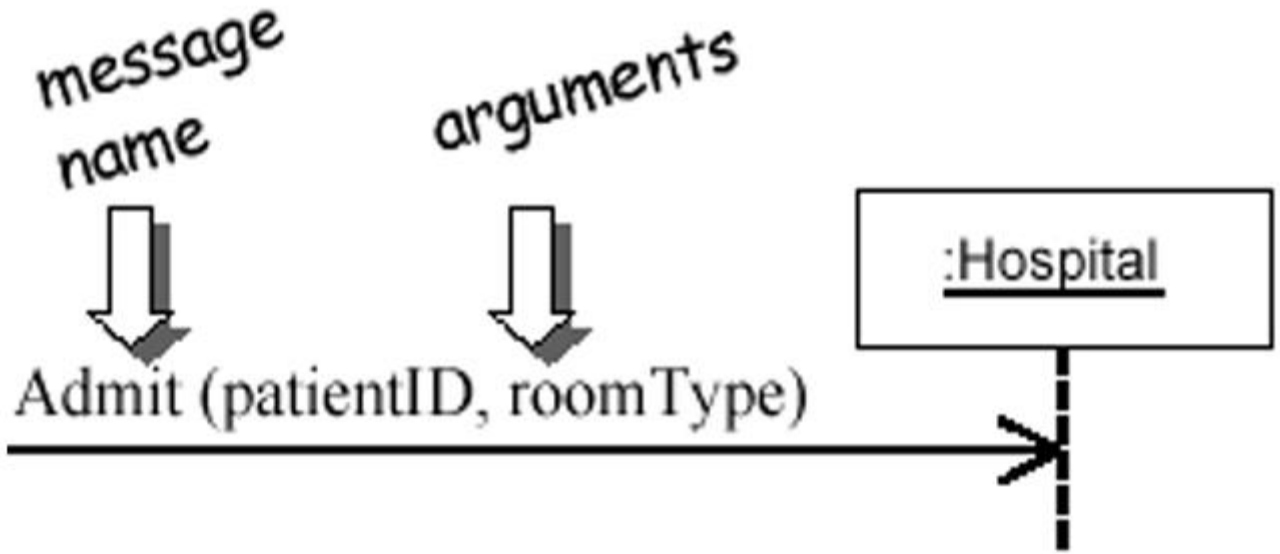
Lifeline

- Each participant has a corresponding lifeline running down the page.
- A participant's lifeline simply states that the part exists at that point in the sequence.
- The lifeline is only really interesting when a part is created and/or deleted during a sequence



Messages

- Communication between participant objects—method calls
- Indicated by an arrow from the lifeline of one object to that of another object.
- In sequence diagram, the message name and arguments are written above the arrow



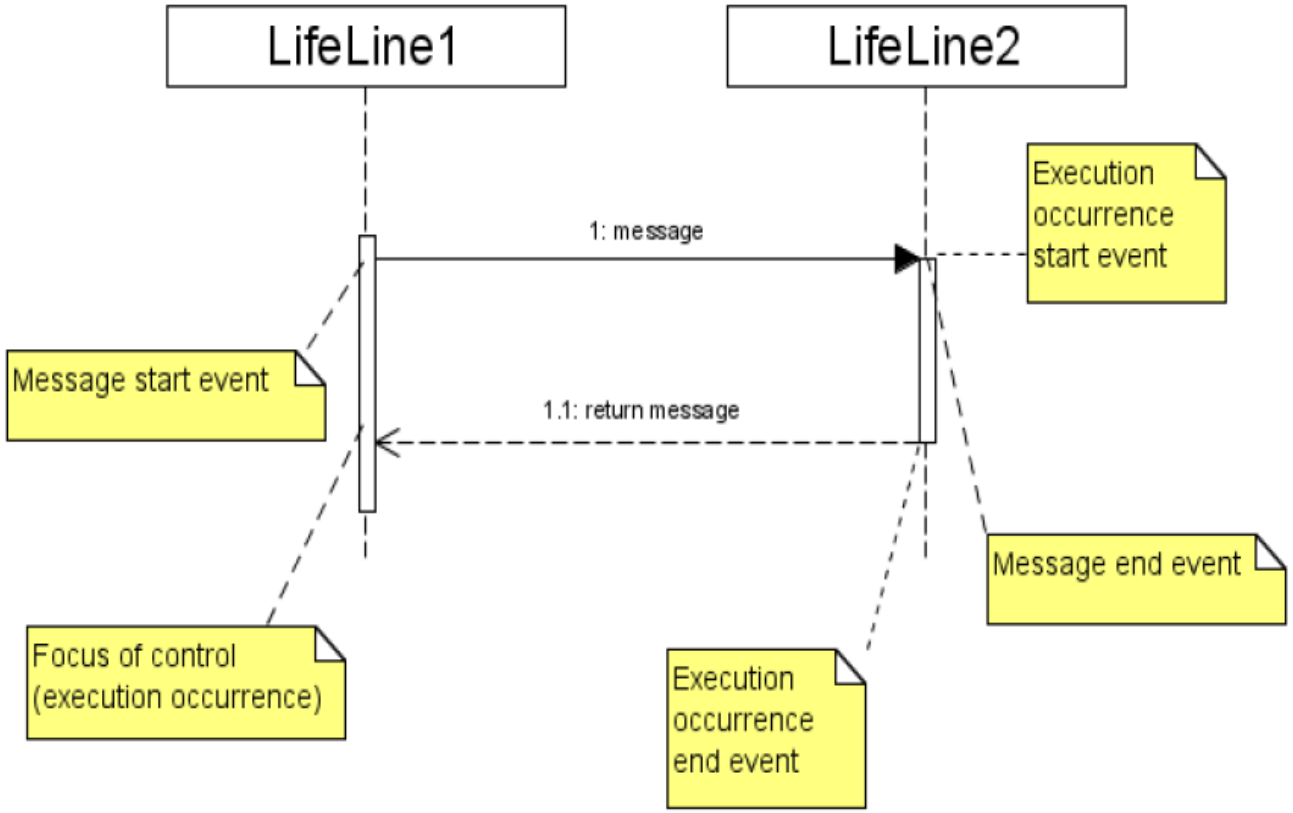
Messages types notation

Message	Description
——▶	synchronous message: the sender waits for the message to be handled before it continues
——>	Asynchronous message: the sender does not wait for the message before it continues
<- - - - - - - -	Return message: return message from another message
- - - - - - - - ->	Creation: This message results in the creation of a new object.

Messages types notation

Message	Description
	Lost: A lost message occurs whet the sender of the message is known but there is no reception of the message.
	Found: A found message indicates that although the receiver of the message is known in the current interaction fragment, the sender of the message is unknown.
	Element destruction: When an element is destroyed during an interaction, the communication that destroys the element is shown with its arrowhead to the elements lifeline where the destruction is marked with a large X symbol

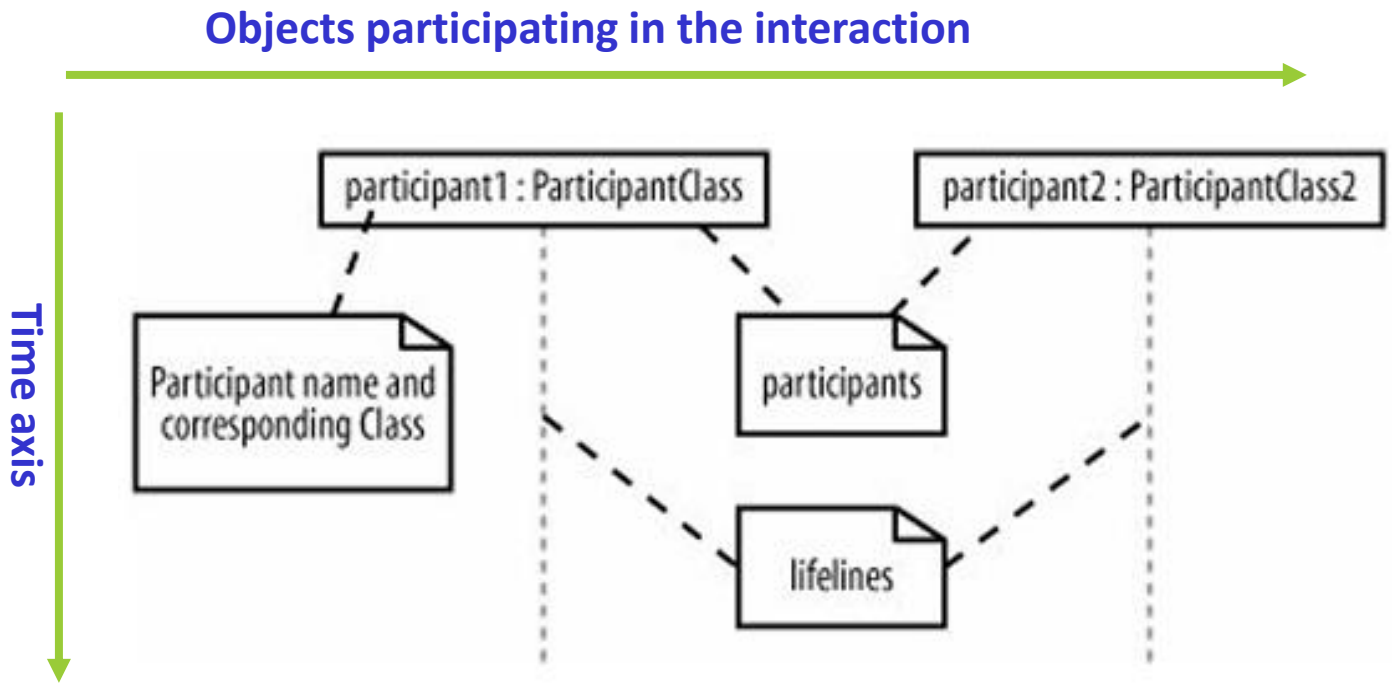
Messages and focus of control representation



- Focus of control (an execution occurrence) represents the period during which an element is performing an operation.

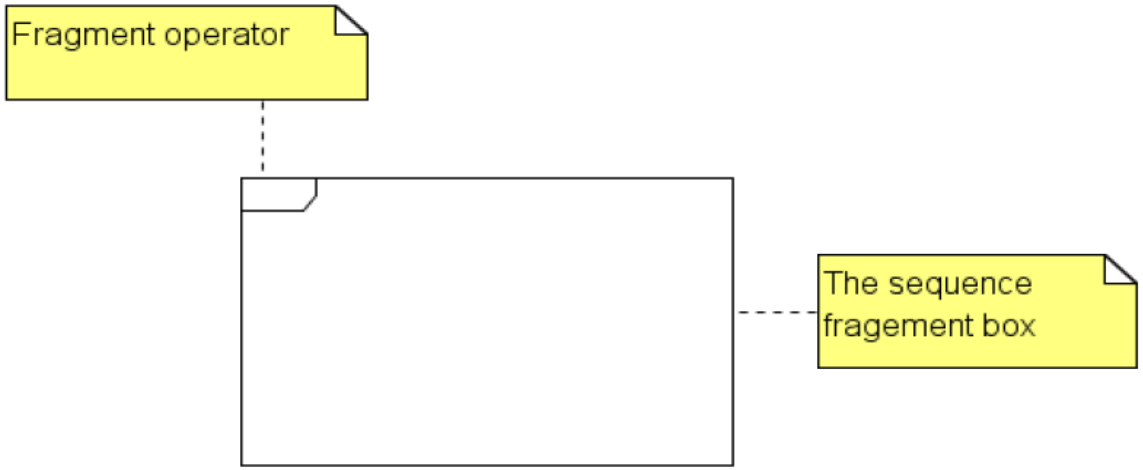
The axis

- horizontal: which object/participant is acting
- Vertical: time (down -> forward in time)



Sequence Fragments

- They make it easier to create and maintain accurate sequence diagrams
- represented as a box, called a combined fragment, which encloses a portion of the interactions within a sequence diagram
- The fragment operator (in the top left cornet) indicates the type of fragment (ref, assert, loop, break, alt, opto u neg)



Combined fragment types representation

- frame: box around part of a sequence diagram to indicate selection or loop
- if -> (opt) [condition]
- if/else -> (alt) [condition], separated by horizontal dashed line
- loop -> (loop) [condition or items to loop over]

Combined fragment types

- Alternatives (alt)
  - ✓ choice of behaviors –at most one will execute
  - ✓ depends on the value of the guard (“else” guard supported)
- Option (opt)
  - ✓ Special case of alternative
- Loop (loop)
  - ✓ Optional guard: [<min>, <max>, <Boolean-expression>]
  - ✓ No guard means no specified limit
- Break (break)
  - ✓ Represents an alternative that is executed instead of the remainder of the fragment (like a break in a loop)

Common operators for interaction frames

Operator	Meaning
alt	**Alternative multiple fragments:** only the one whose condition is true will execute.
opt	**Optional:** the fragment executes only if the supplied condition is true. Equivalent to an alt only with one trace.
par	**Parallel:** each fragment is run in parallel.
loop	**Loop:** the fragment may execute multiple times, and the guard indicates the basis of iteration.
region	**Critical region:** the fragment can have only one thread executing it at once.
neg	**Negative:** the fragment shows an invalid interaction.
ref	**Reference:** refers to an interaction defined on another diagram. The frame is drawn to cover the lifelines involved in the interaction. You can define parameters and a return value.
sd	**Sequence diagram:** used to surround an entire sequence diagram.

How to produce sequence diagrams

- 1.Decide on Context: Identify behavior (or use case) to be specified
- 2. Identify structural elements:

  (a) Model objects (classes)

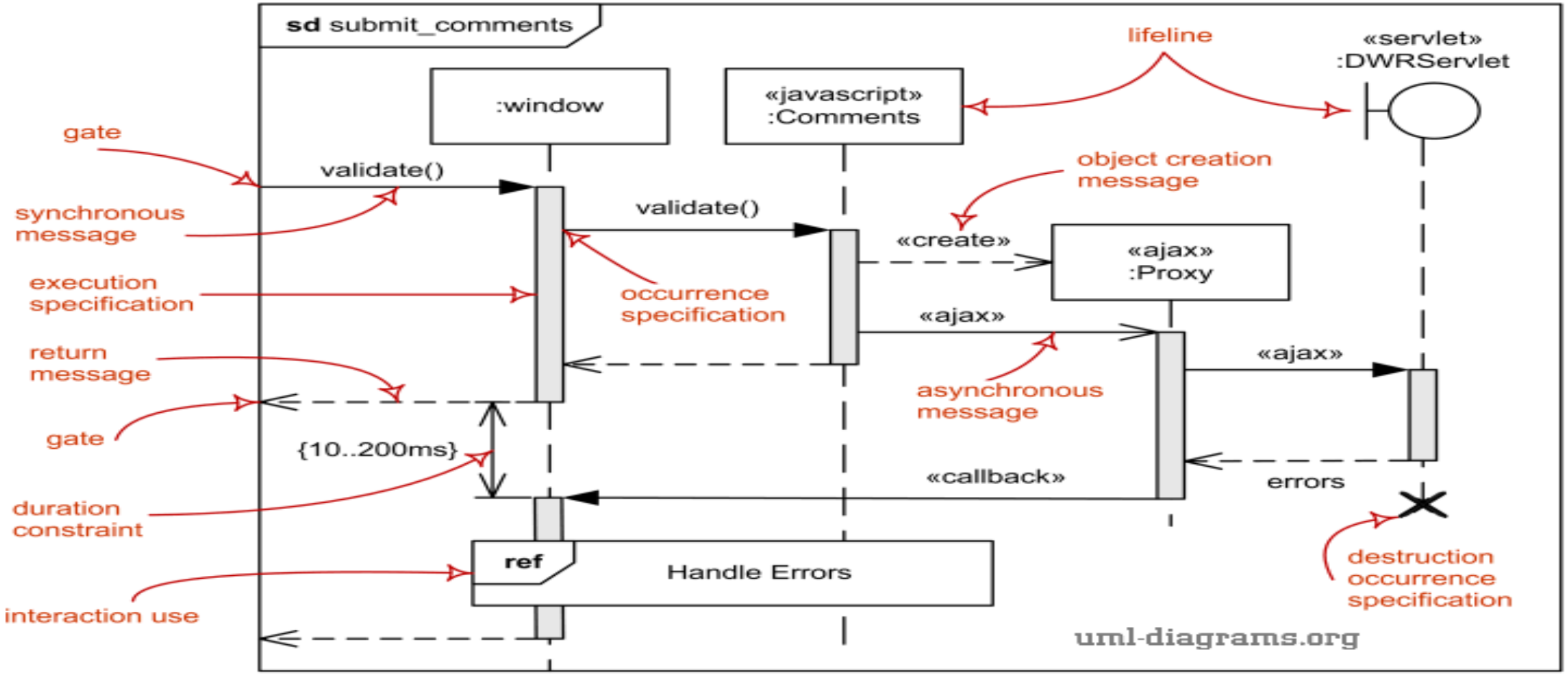
  (b) Model lifelines

  (c) Model activations

  (d) Model messages

  (e) Model Timing constraints
- 3. Refine and elaborate as required

Summary: Sequence diagram graphical notations



Tutorial 1

- Present a sequence diagram for the use cases of the banking system established during the previous class

Tutorial 2: Use Case: Process Sale Scenario - Main Success Story

- 1. Cashier starts new sale
- 2. Cashier enters item identifier
- 3. System records sale line item and presents item description, price and running total

  Steps 2 and 3 are repeated until all items are processed.
- 4. System presents total with taxes calculated
- 5. Cashier tells Customer the total and asks for payment
- 6. Customer pays and System handles payment

Task: Propose a sequence diagram for the above use case