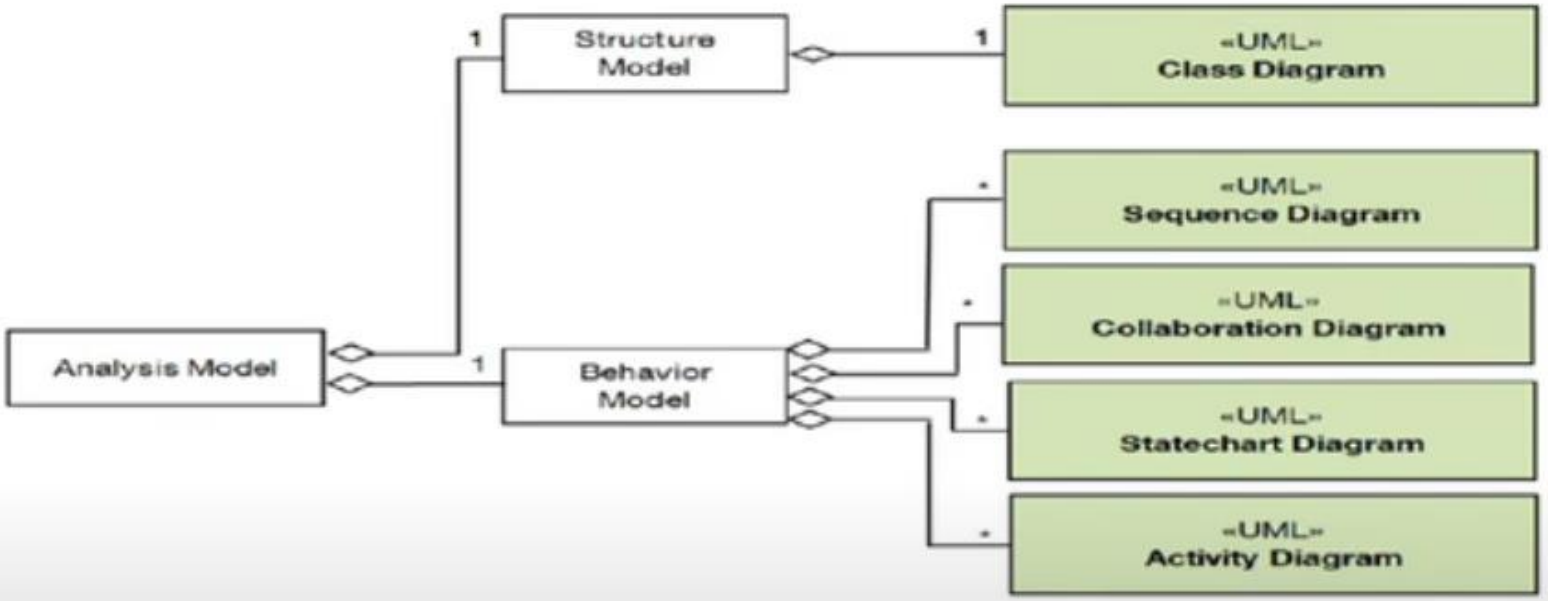
Sequence Diagrams

Overview

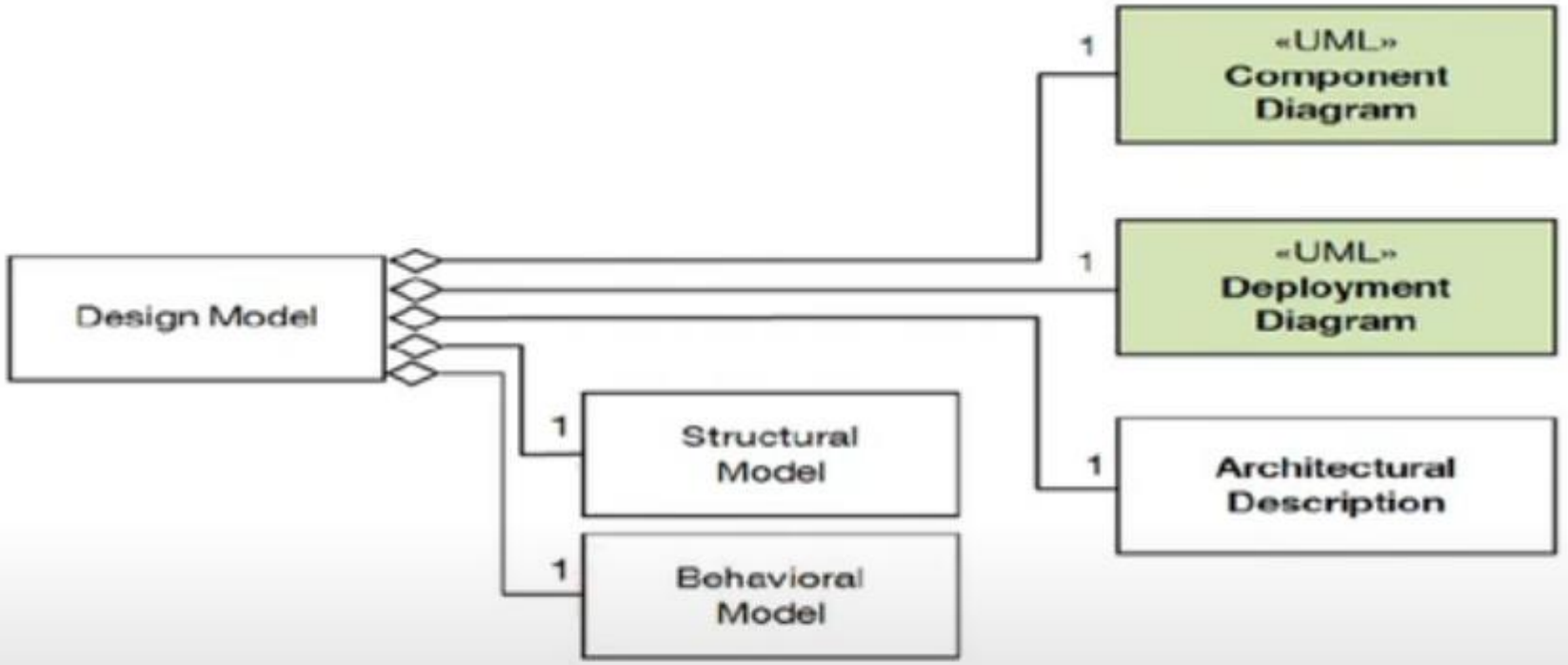
- What are Sequence Diagrams?
  - Lifeline
  - Messages
  - Interaction Fragments
  - Examples
- No object exists in isolation
- Objects are acted on and themselves act on other objects
- Leads to the **Client-Server Model** of computing where
  - Behavior is
    - Services provided by an object
  - Services are requested by
    - Sending Messages, Invoking Operations
  - In Client-Server View
    - Clients request for Services
    - Servers provide Services
    - Contract between client and server ensures correctness

When to use Sequence Diagram



- In the Analysis Phase the problem domain is analyzed and refined from the Requirements Phase
- The behavior model of the system is hence understood in this phase
- Sequence diagram is a major result of the Analysis Phase

When to use Sequence Diagram



- Sequence diagram is included in the Behavioral Model
- It is further refined in the Design Phase

What are Sequence Diagrams?

- Sequence diagram is the most common kind of Interaction diagram, which focuses on the message interchange between a number of lifelines
- Sequence diagram is a UML behavior diagram
- Sequence diagram depicts the inter-object behavior of a system, ordered by time
- The major components of a Sequence Diagram are:
  - Lifeline
  - Messages
  - Interaction Fragments

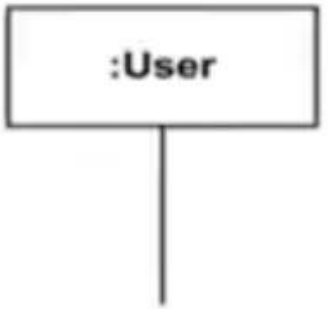
Sequence Diagrams

- **Lifeline** is an element which represents an individual participant in the interaction
- Lifelines represent only one interacting entity
- If the referenced connectable element is multi-valued (that is, has a multiplicity > 1), then the lifeline may have an expression (selector) that specifies which particular part is represented by this lifeline
- A lifeline is shown using a symbol that consists of a rectangle forming its "head" followed by a vertical line (which may be dashed) that represents the lifetime of the participant
- The information identifying a lifeline is depicted as ObjectName[selector]:ClassName

Source: *UML 2.5 Diagrams Overview*: http://www.uml-diagrams.org/uml-25-diagrams.html (18-Aug-16)

Sequence Diagrams

Lifeline "data" of class Stock



Anonymous lifeline of class User

Lifeline "x" of class X is selected with selector [k]

Source: *UML 2.5 Diagrams Overview*: http://www.uml-diagrams.org/uml-25-diagrams.html (18-Aug-16)

Named Elements in Leave Management System

- The major named elements of LMS are Employee and Leave. Few instances of them shown below.
- The major named elements of LMS are Employee and Leave. Few instances of them shown below.

- The major interaction activity of LMS is **Request Leave**, **Approve Leave** which requires interaction between the two major classes, Employee and Leave

Types of Messages

- Message is an element that defines one specific kind of communication between lifelines of an interaction
- There are 2 major types of message in Sequence Diagram
  - Messages by Action Type
  - Messages by Presence of Events

Message by Action Type: A message reflects either an operation call and start of execution or a sending and reception of a signal

Message by Presence of Events: A message depends on whether message send event and receive events are present

Source: *UML 2.5 Diagrams Overview*: http://www.uml-diagrams.org/uml-25-diagrams.html (18-Aug-16)

Messages by Action Type

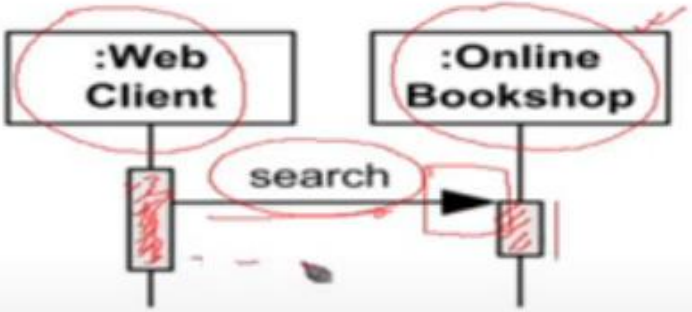
The various types of Messages by Action type are:

- synchronous call
- asynchronous call / signal
- create
- delete
- reply

Messages by Action Type

Synchronous call typically represents operation call - send message and suspend execution while waiting for response

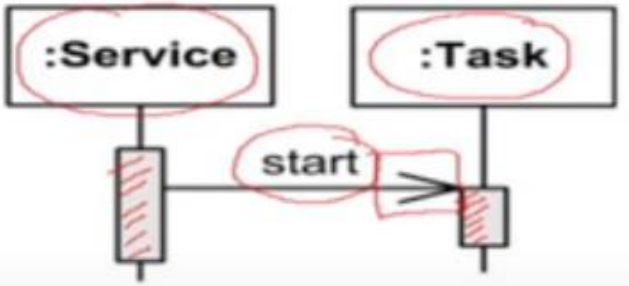
Notation: filled arrow head.



Web Client searches Online Bookshop and waits for results

Asynchronous call - send message and proceed immediately without waiting for return value

Notation: Open arrow head



Service starts Task and proceeds in parallel without waiting

Source: *UML 2.5 Diagrams Overview*: http://www.uml-diagrams.org/uml-25-diagrams.html (18-Aug-16)

Messages by Action Type

Create message is sent to a lifeline to create itself

Notation: Dashed line with open arrowhead

Online Bookshop creates Account

Delete message is sent to terminate another lifeline

Notation: lifeline usually ends with a cross (X) at the bottom

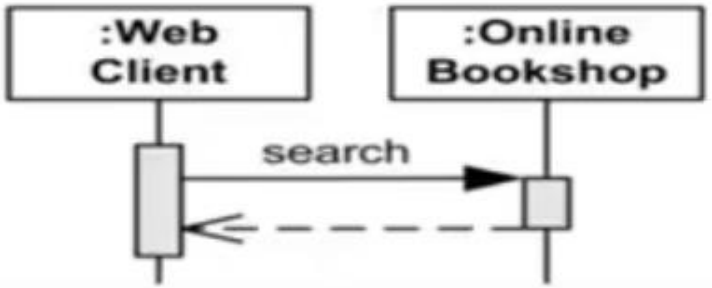
Online Bookshop terminates Account

Source: *UML 2.5 Diagrams Overview*: http://www.uml-diagrams.org/uml-25-diagrams.html (18-Aug-16)

Messages by Action Type

Reply message to an operation call

Notation: Dashed line with open arrow head



Web Client searches Online Bookshop and waits for results to be returned

Source: *UML 2.5 Diagrams Overview*: http://www.uml-diagrams.org/uml-25-diagrams.html (18-Aug-16)

Messages by Presence of Events

The various types of Messages by Presence of Events are:

- complete message
  - The semantics of a complete message is the trace <sendEvent, receiveEvent>
  - Both sendEvent and receiveEvent are present
- lost message
- found message
- unknown message (default) – both sendEvent and receiveEvent are absent (should not appear)

Messages by Presence of Events

Lost Message is a message where the sending event is known, but there is no receiving event

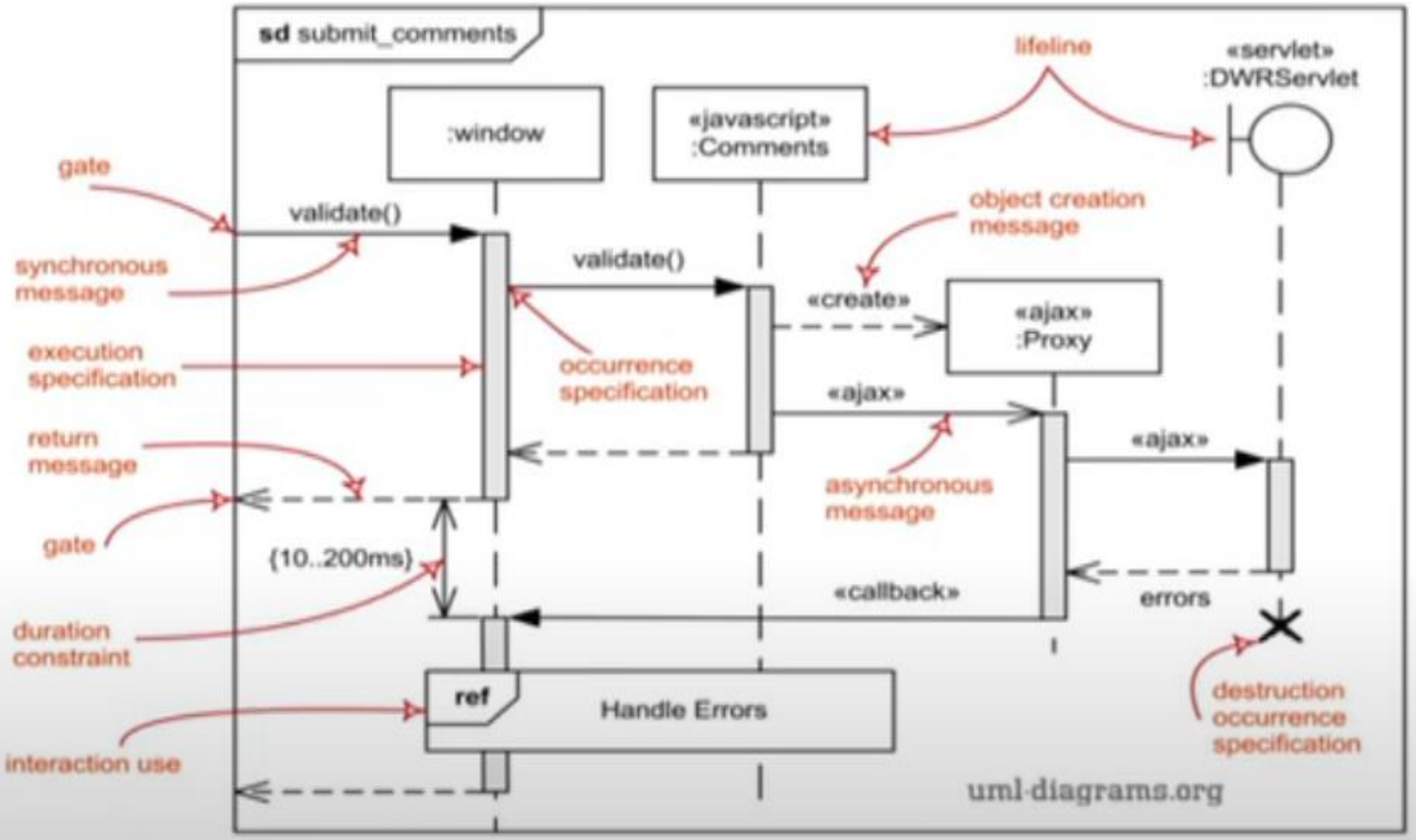
Web Client sent search message which was lost

Found Message is a message where the receiving event is known, but there is no (known) sending event

Online Bookshop gets search message of unknown origin

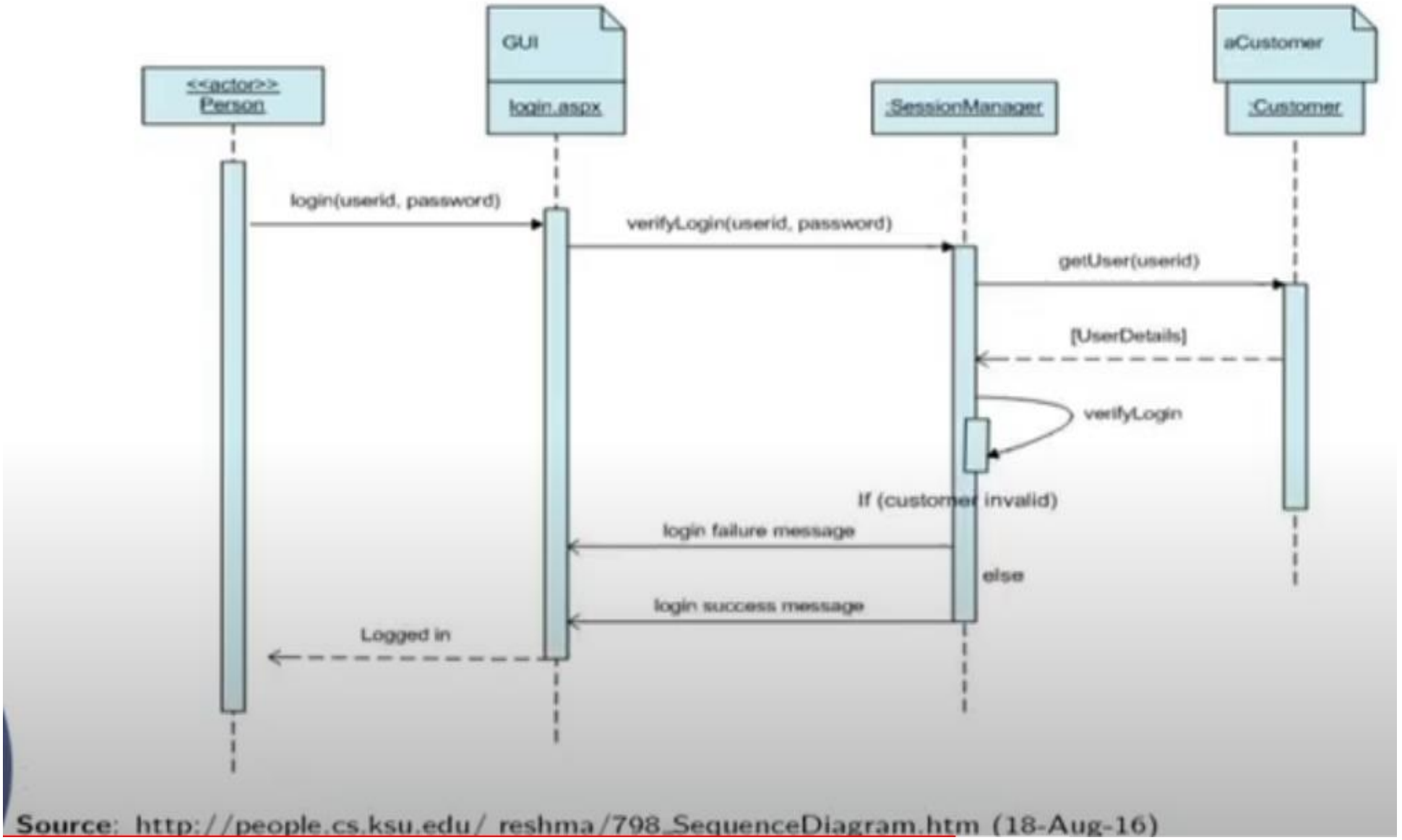
Source: *UML 2.5 Diagrams Overview*: http://www.uml-diagrams.org/uml-25-diagrams.html (18-Aug-16)

An Annotated Sequence Diagram



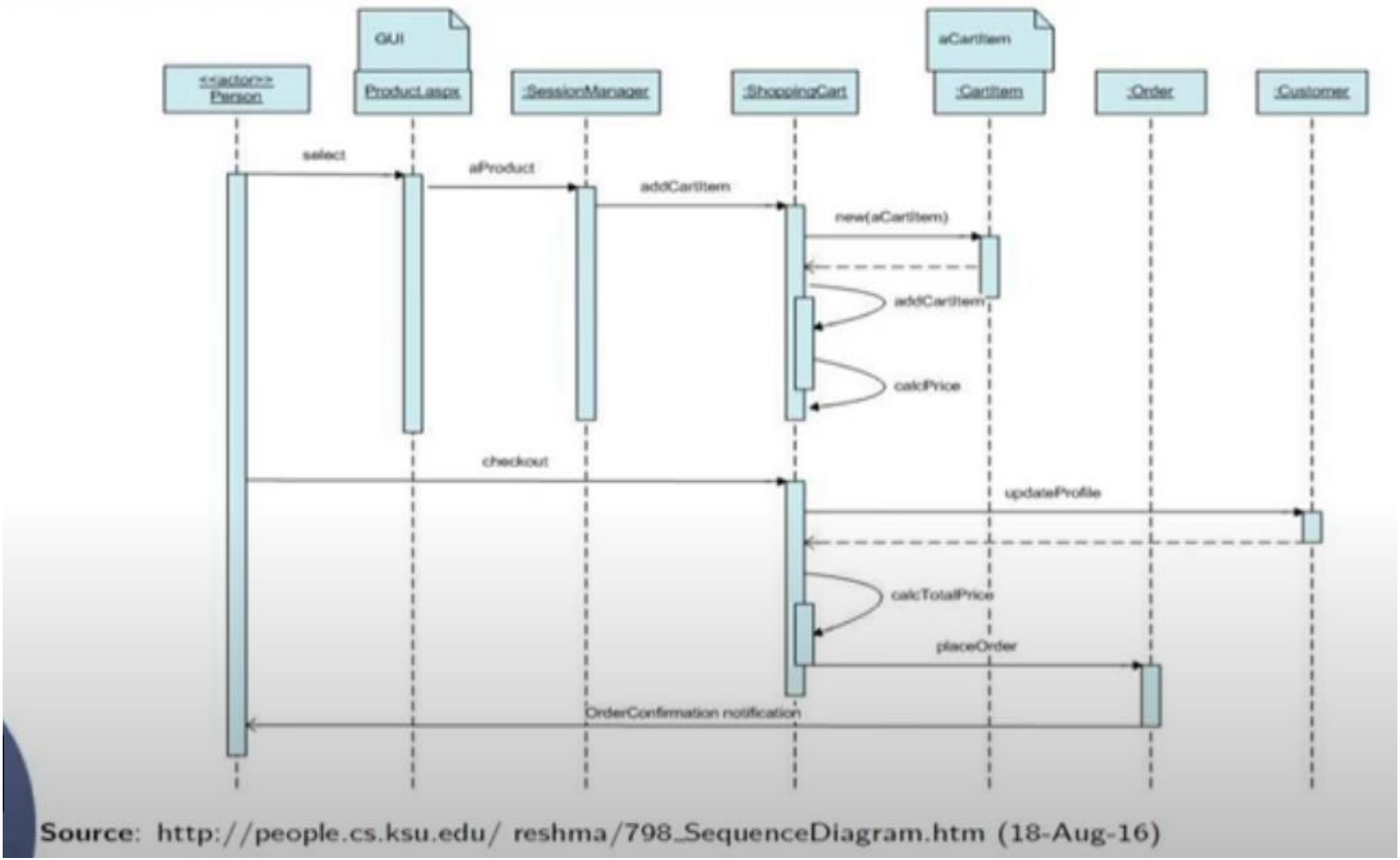
Source: *UML 2.5 Diagrams Overview*: http://www.uml-diagrams.org/uml-25-diagrams.html (18-Aug-16)

Example Sequence Diagram - Login



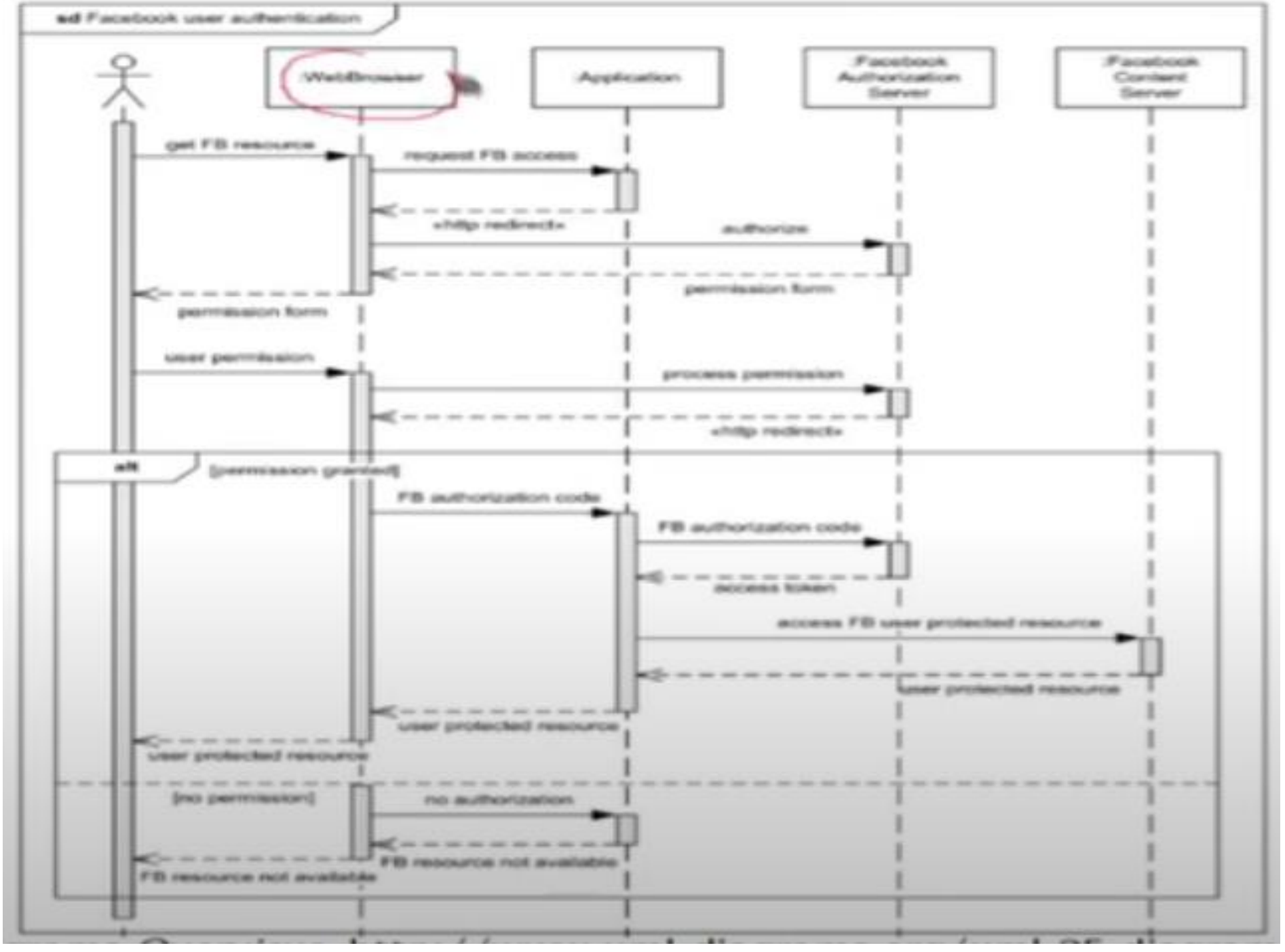
Source: http://people.cs.ksu.edu/ reshma/798_SequenceDiagram.htm (18-Aug-16)

Example Sequence Diagram – Place Order



Source: http://people.cs.ksu.edu/ reshma/798_SequenceDiagram.htm (18-Aug-16)

Example Sequence Diagram – Facebook Authentication



Messages of Leave Management System

The messages for the major activities of LMS are given below:

- **Request Leave**
  - Request Leave() from Employee
  - new() Leave
  - isValid() Leave
  - return(ifvalid == true)
- **Approve Leave**
  - Approve Leave() from Employee
  - Approver()
  - Reportingto()
  - return(Reportingto)

Summary

- Introduced sequence diagram to capture the detailed execution flows of objects, their interactions and lifeline with a temporal ordering among events
- Discussed lifeline and messages in depth with examples

Sequence Diagrams

- What are Sequence Diagrams?
  - Lifeline
  - Messages
  - Interaction Fragments
  - Examples
- Sequence Diagram for LMS

- The various objects of the system interact with each other, through exchange of messages to invoke the various operations of the object.
- Sequence diagram is a major diagram to depict the inter object behaviour of a system, ordered by time.
- Sequence diagram is the most common kind of interaction diagram, which focuses on the message interchange between a number of lifelines.
- The major components of a Sequence Diagram are
  - Lifeline
  - Messages
  - Interaction Fragments

Sequence Diagrams

- Interaction fragment is a named element representing the most general interaction unit
- Each interaction fragment is conceptually like an interaction by itself
- There is no general notation for an interaction fragment. Its sub-classes define their own notation
- Examples of Interaction Fragments include:
  - Occurrence
  - Execution
  - State invariant
  - Combined fragment
  - Interaction use

Source: *UML 2.5 Diagrams Overview*: http://www.uml-diagrams.org/uml-25-diagrams.html (18-Aug-16)

Sequence Diagrams – Interaction Fragment Occurrence

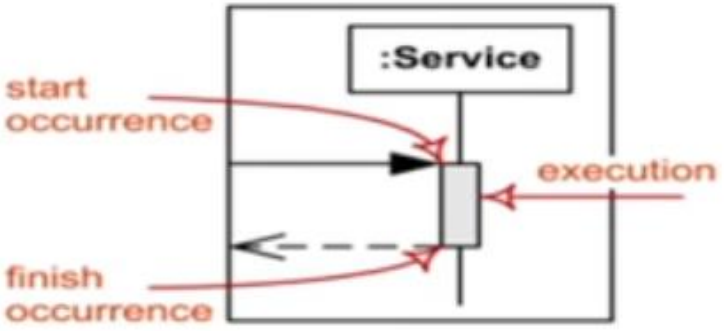
Occurrence is interaction fragment which represents a moment in time (event) at the beginning or end of a message or at the beginning or end of an execution

Message occurrence represents events as sending and receiving of signals

Destruction occurrence destruction of the instance described by the life-line

Account lifeline is terminated

Execution occurrence represents moments in time at which actions or behaviors start or finish.



Duration of an execution is represented by two execution occurrences - start and finish

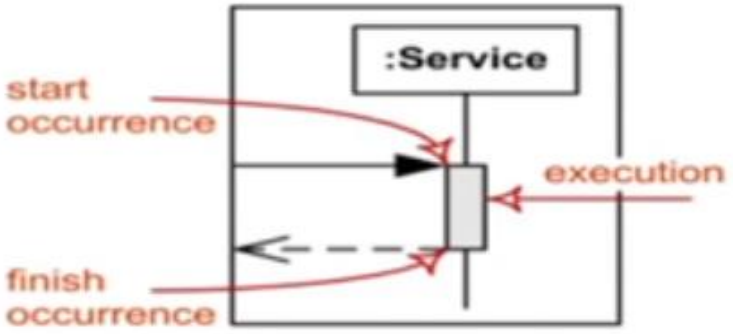
Source: *UML 2.5 Diagrams Overview*: http://www.uml-diagrams.org/uml-25-diagrams.html (18-Aug-16)

Sequence Diagrams – Interaction Fragment Execution (Activation)

Execution (Activation) is an interaction fragment which represents a period in the participant's lifetime when it is

- executing a unit of behavior or action within the lifeline, or
- sending a signal to another participant, or
- waiting for a reply message from another participant

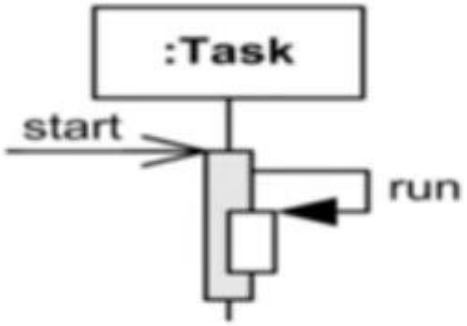
Execution is represented as a thin grey or white rectangle on the life-line	Execution can be represented by a wider labeled rectangle, where the label identifies the action



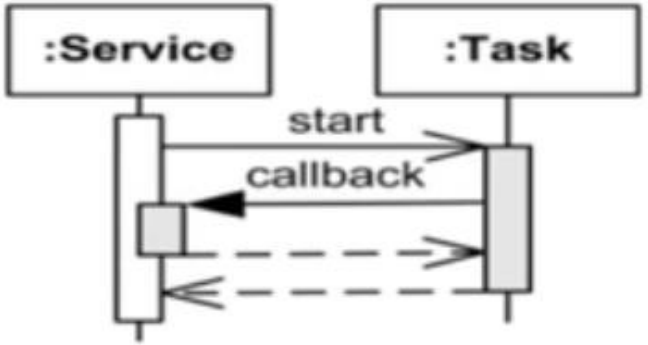
Source: *UML 2.5 Diagrams Overview*: http://www.uml-diagrams.org/uml-25-diagrams.html (18-Aug-16)

Sequence Diagrams – Interaction Fragment Overlapping Execution (Activation)

Overlapping execution specifications on the same lifeline are represented by overlapping rectangles



Overlapping execution specifications on the same lifeline - message to self

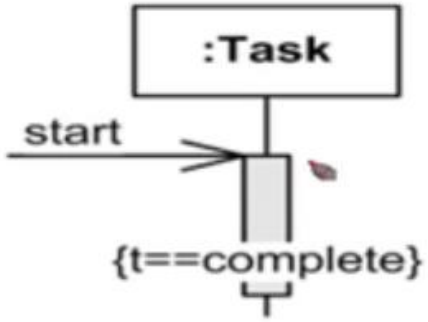
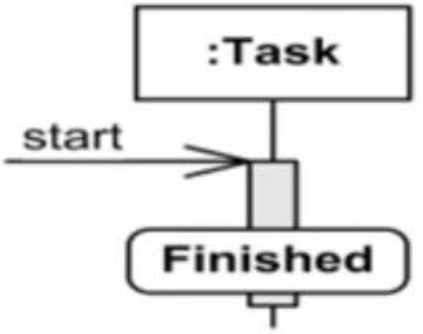


Overlapping execution specifications on the same lifeline - callback message.

Source: *UML 2.5 Diagrams Overview*: http://www.uml-diagrams.org/uml-25-diagrams.html (18-Aug-16)

Sequence Diagrams – Interaction Fragment State Invariant

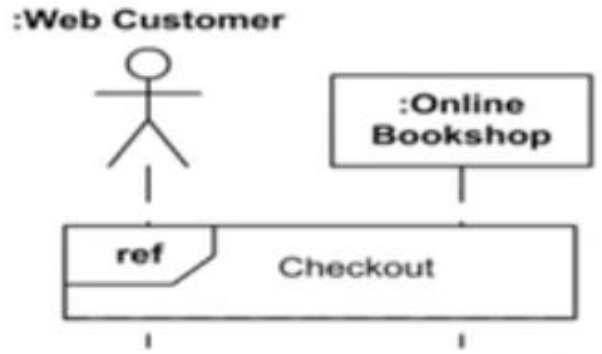
State Invariant is an interaction fragment which represents a run-time constraint on the participants of the interaction. It may be used to specify different kinds of constraints, such as values of attributes or variables, internal / external states, etc.

State invariant is usually shown as a constraint in curly braces on the lifeline	State invariant may be shown as a state symbol
	
Attribute t of Task should be equal to complete	Task should be in Finished state

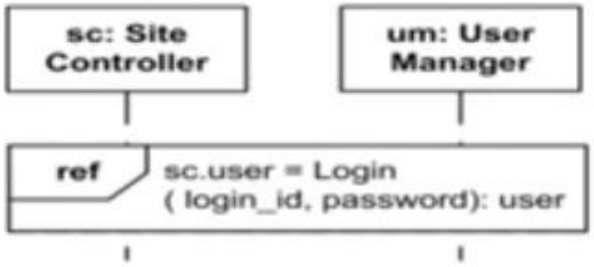
Source: *UML 2.5 Diagrams Overview*: http://www.uml-diagrams.org/uml-25-diagrams.html (18-Aug-16)

Sequence Diagrams – Interaction Fragment Interaction Use

Interaction Use is an an interaction fragment which allows to use (or call) another interaction. Large and complex sequence diagrams could be simplified with interaction uses. It is also common to reuse some interaction between several other interactions



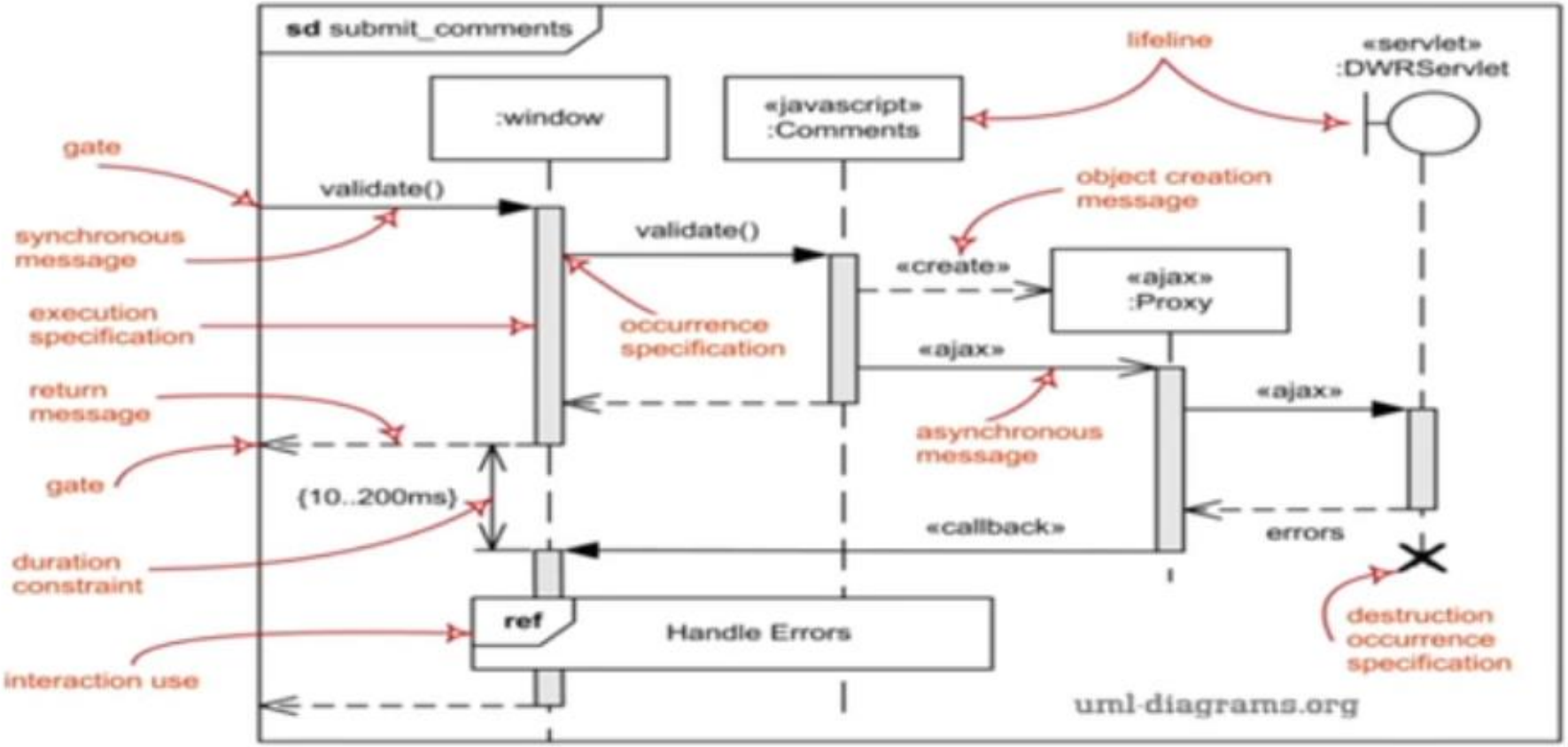
Web customer and Bookshop use (reference) interaction Checkout



Use Login interaction to authenticate user and assign result back to the user attribute of Site Controller

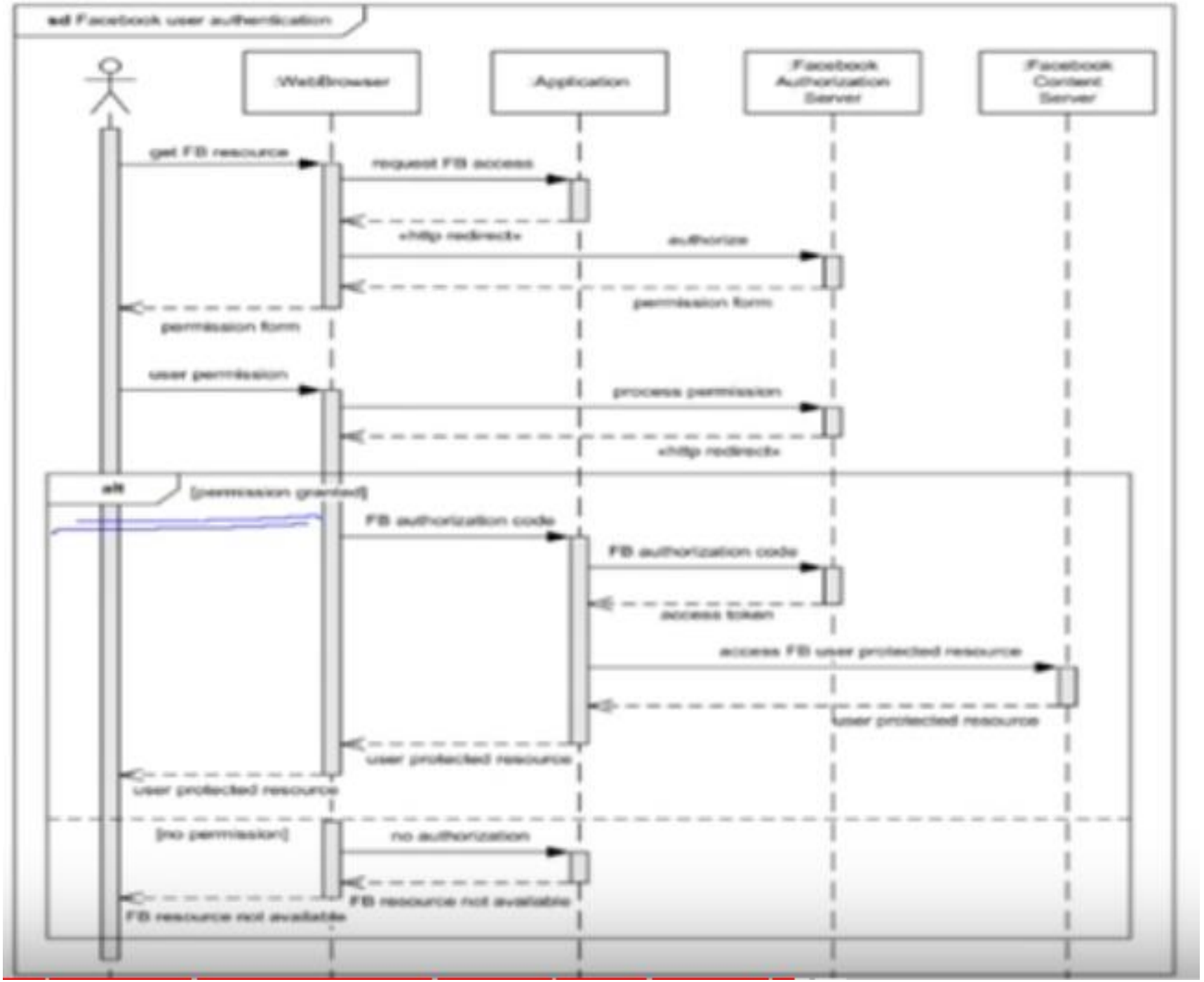
Source: *UML 2.5 Diagrams Overview*: http://www.uml-diagrams.org/uml-25-diagrams.html (18-Aug-16)

An Annotated Sequence Diagrams



Source: *UML 2.5 Diagrams Overview*: http://www.uml-diagrams.org/uml-25-diagrams.html (18-Aug-16)

Sequence Diagrams Facebook Authentication



Reference

Source: NPTEL - Object-Oriented Analysis and Design, IIT Kharagpur Prof. Partha Pratim Das Prof. Samiran Chattopadhyay Prof. Kausik Datta

https://nptel.ac.in/courses/106105153