

# NATIONAL UNIVERSITY OF COMPUTER AND EMERGING SCIENCES

## CS3005– Software Design & Architecture Lab

LAB Instructors: Sobia Iftikhar “Sobia.iftikhar@nu.edu.pk”

### Lab 09

#### Outline

- To Understand Interaction Diagrams
- Sequence Diagram
- System Sequence Diagram
- Collaboration Diagram
- Exercise

## Interaction Diagrams

The purpose of interaction diagrams is to visualize the interactive behaviour of the system. Visualizing the interaction is a difficult task. Hence, the solution is to use different types of models to capture the different aspects of the interaction.

Sequence and collaboration diagrams are used to capture the dynamic nature but from a different angle.

*The purpose of interaction diagram is –*

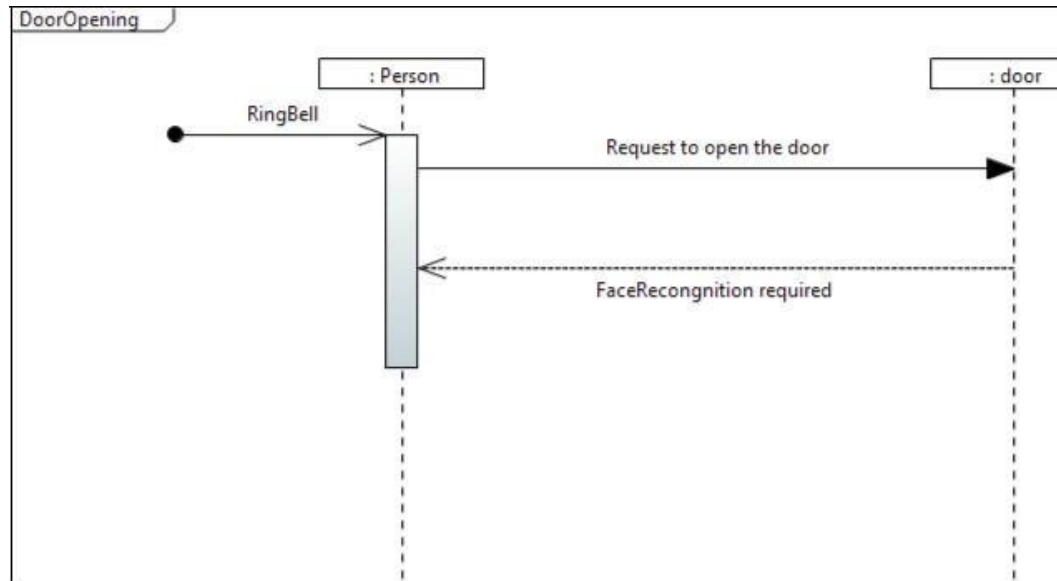
- To capture the dynamic behaviour of a system.
- To describe the message flow in the system.
- To describe the structural organization of the objects.
- To describe the interaction among objects.

### *Interaction Diagram Example - Access Control System*

- The Interaction Establish Access occurs first with argument “Illegal PIN” followed by an interaction with the message Card Out which is shown in an inline Interaction.
- Then there is an alternative as we find a decision node with an Interaction Constraint on one of the branches.
- Along that control flow we find another inline Interaction and an Interaction Use in the sequence.

#### Example 01

The example shows a person who found ring bell for calling inside the room, being a person he request to open the door. Door is digital they required face recondition



## Sequence Diagram


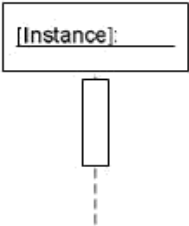
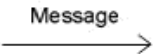
- Sequence diagrams model the dynamic aspects of a software system.
- The emphasis is on the “sequence” of messages rather than relationship between objects.
- A sequence diagram maps the flow of logic or flow of control within a usage scenario into a visual diagram enabling the software architect to both document and validate the logic during the analysis and design stages.
- Sequence diagrams provide more detail and show the message exchanged among a set of objects over time.
- Sequence diagrams are good for showing the behavior sequences of a diagram, shows only the sequence of messages not their exact timing. It shows when and how long object is active/idle.

## Purpose

- The main purpose of this diagram is to represent how different business objects interact.
- A sequence diagram shows object interactions arranged in time sequence.
- It depicts 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 scenario.

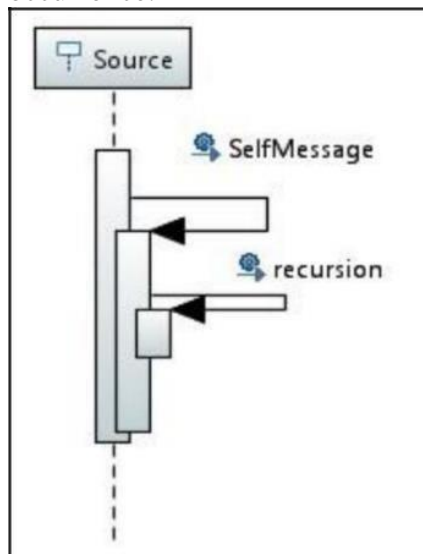
## When to use: Sequence Diagram

- Sequence diagram can be a helpful modeling tool when the dynamic behavior of objects needs to be observed in a particular use case or when there is a need for visualizing the “big picture of message flow”.
- A company’s technical staff could utilize sequence diagrams in order to document the behavior of a future system.
- It is during the design period that developers and architects utilize the diagram to showcase the system’s object interactions, thereby putting out a more fleshed out overall system design.

Life line		Lifeline represents the duration during which an object is alive and interacting with other objects in the system. It is represented by dashed lines.
Scope		It shows the time period during which an object or actor is performing an action.
Message transition		To send message from one object to another.

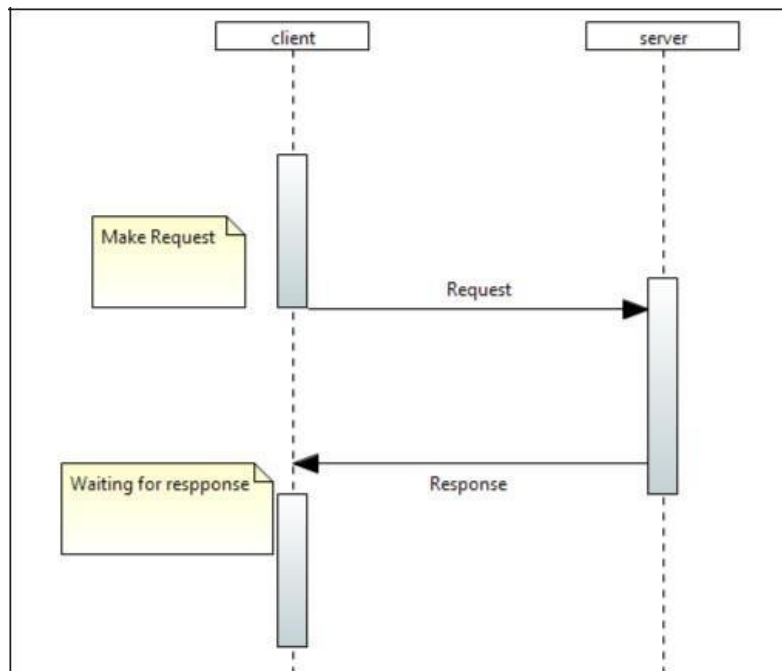
## Self-Message

A self-messages can represent a recursive call of an operation, or one method calling another method belonging to the same object. It is shown as creating a nested focus of control in the lifeline's execution occurrence.



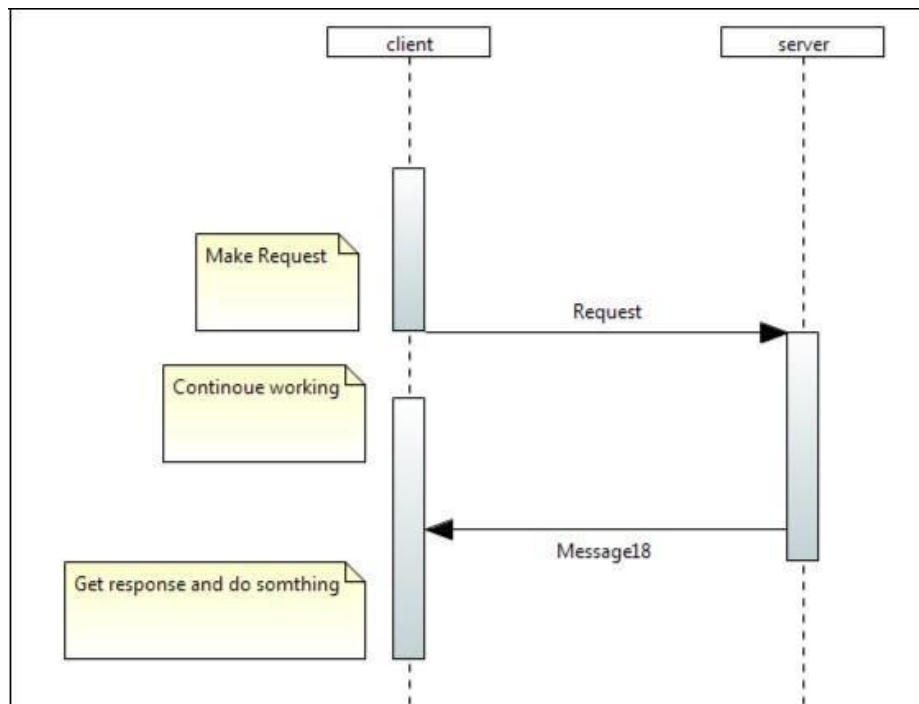
## Synchronous Calls

**Synchronous messages** only request operations, not signals. The message type is synchCall. Synchronous messages wait for a response from the operation before resuming their behavior. Synchronous messages are displayed as an arrow with a filled arrowhead.



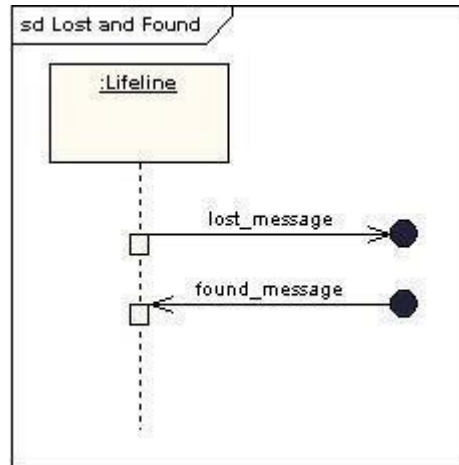
## Asynchronous Calls

**Asynchronous messages** of the type (MessageSort) **asynchCall**, connect to an operation and trigger its execution. With asynchronous messages, the system does not wait for a response from the recipient, but continues its processes without interruption. Operation parameters and message attributes must match.



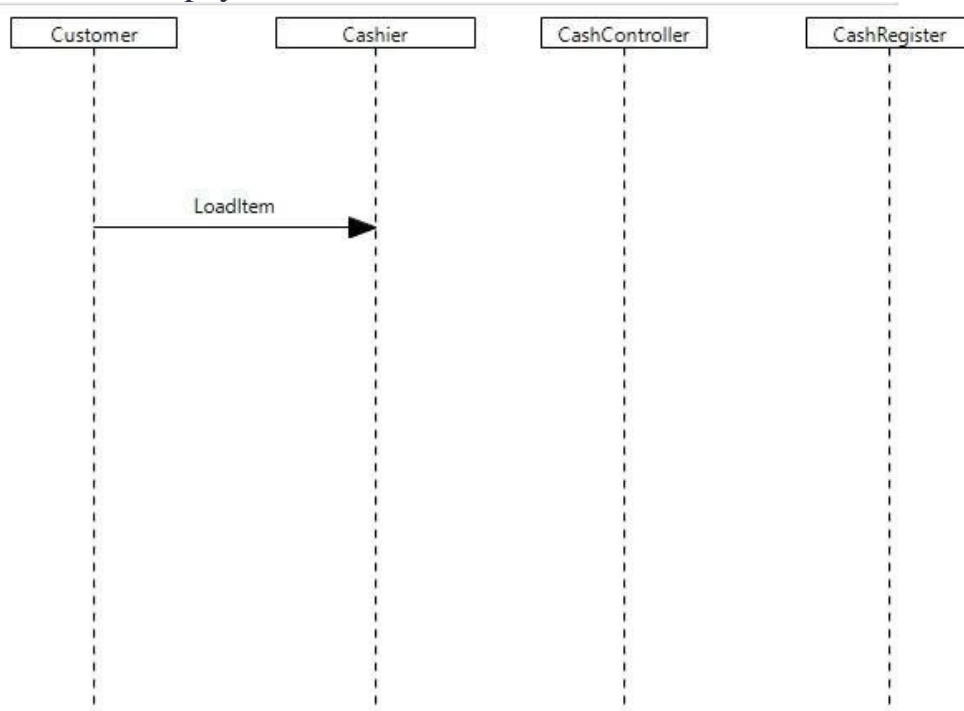
## Lost and Found Messages

Lost messages are those that are either sent but do not arrive at the intended recipient, or which go to a recipient not shown on the current diagram. Found messages are those that arrive from an unknown sender, or from a sender not shown on the current diagram. They are denoted going to or coming from an endpoint element



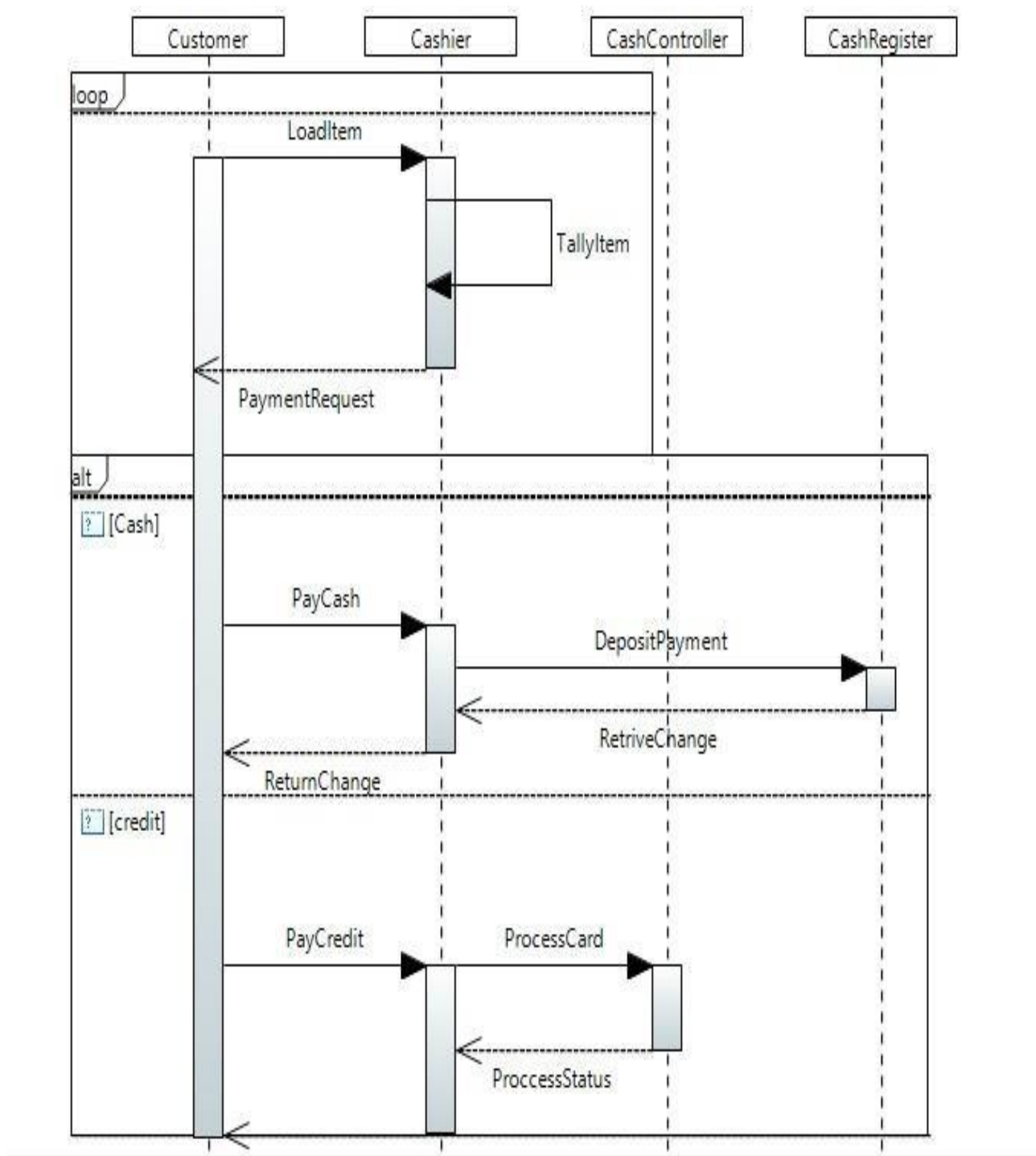
## Example: Customer Cashier

This Sequence diagram illustrates the use of Combined Fragments in modelling a simplified purchasing process. Cashiers scan items ensure that prices are quantities are correct, and collect payments.

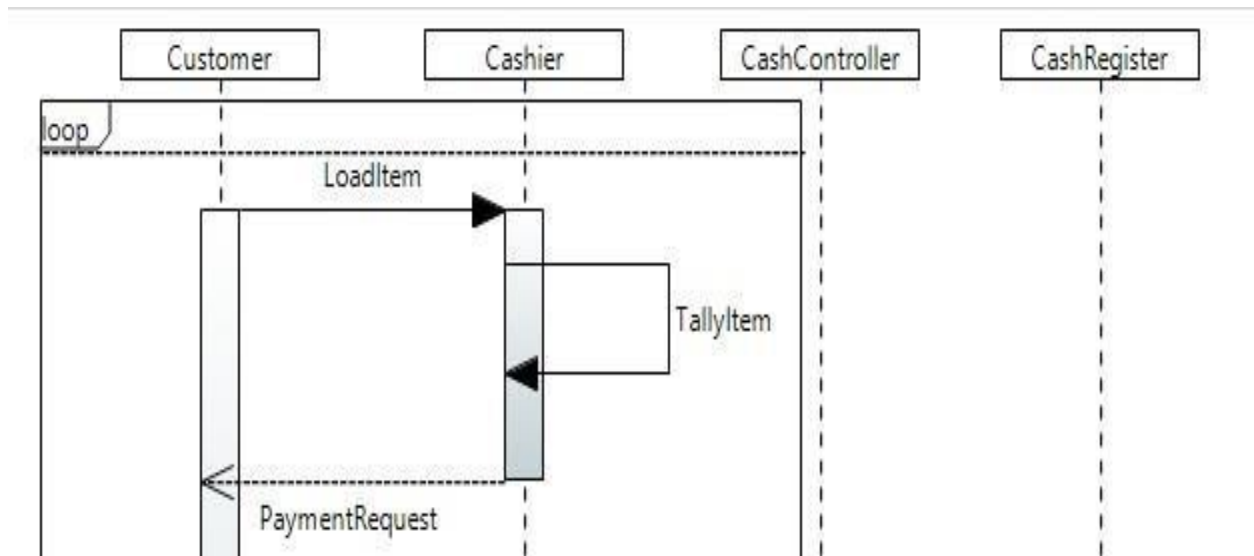


## Alternative Combined Fragment

A Combined Fragment reflects one or more aspects of interaction (called interaction operands) controlled by an interaction operator, with corresponding Boolean conditions known as interaction constraints. The Fragment displays as a transparent window, divided by horizontal lines for each operand.



**Loop Combined Fragment.** A loop fragment represents iteration through an unknown number of items for purchase, after which the cashier requests payment.



## System Sequence Diagram:

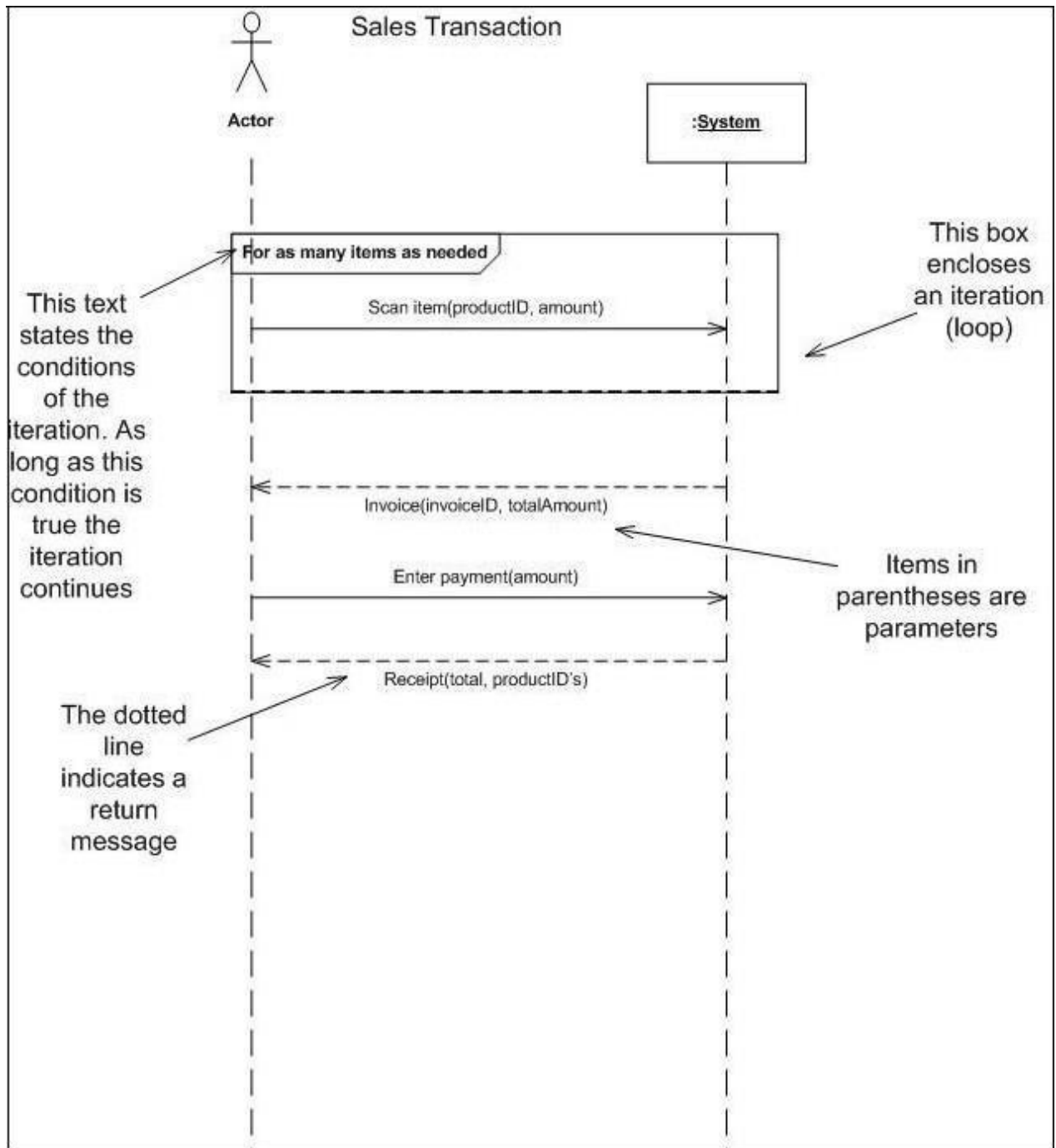
A system sequence diagram (SSD) is a sequence diagram that shows, for a particular scenario of a use case, the events that external actors generate, their order, and possible inter-system events.

### Overview

System sequence diagrams are visual summaries of the individual use cases. All systems are treated as a black box; the diagram places emphasis on events that cross the system boundary from actors to systems. A system sequence diagram should be done for the main success scenario of the use case, and frequent or complex alternative scenarios.

A system sequence diagram should specify and show the following:

- **External actors**
- **Messages (methods) invoked by these actors**
- **Return values (if any) associated with previous messages**
- **Indication of any loops or iteration area**



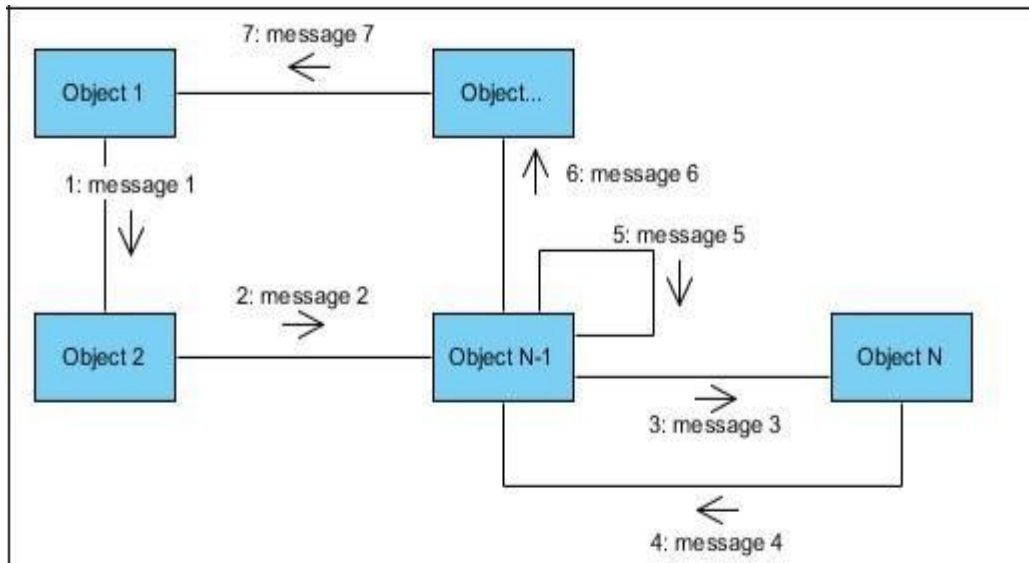
## Collaboration/Communication Diagram

Communication diagram is a kind of UML diagram that is designed for illustrating the dynamic view of the system. It emphasizes the structural organization of the objects' send and receive messages. In this example of the notation for a communication diagram, objects (actors in use cases) are represented by rectangles. In the example (generic communication diagram):

- The objects are Object1, Object2, Object..., ObjectN-1 ..., and ObjectN.

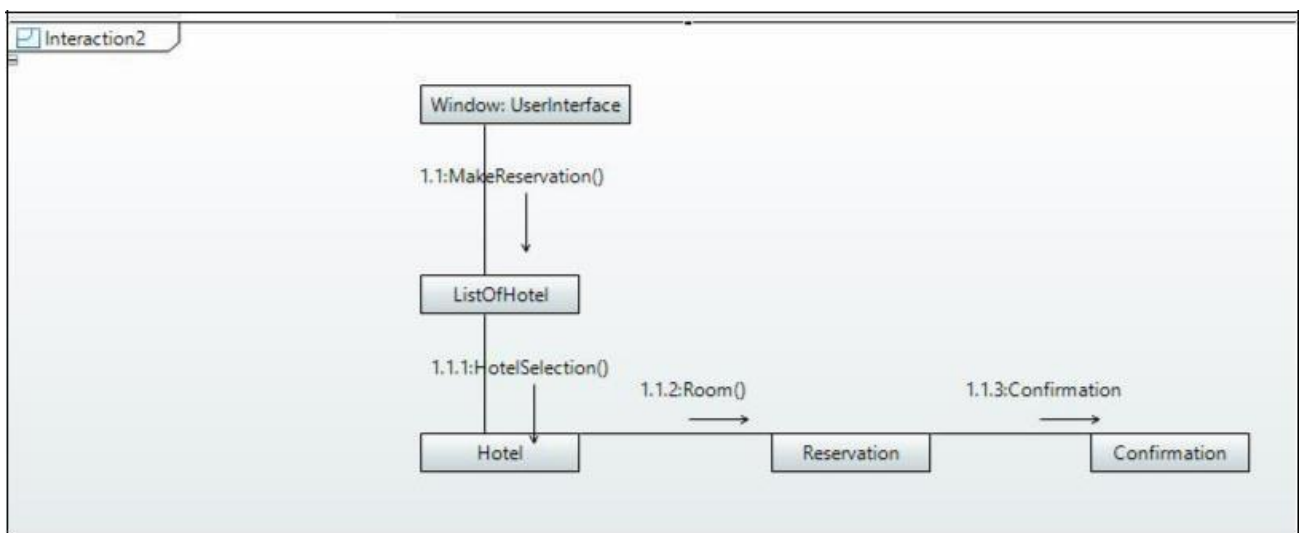


- Messages passed between objects are represented by labeled arrows that start with the sending object (actor) and end with the receiving object.
- The sample messages passed between objects are labeled 1: message1, 2: message2, 3: message3, etc., where the numerical prefix to the message name indicates its order in the sequence.
- Object1 first sends Object2 the message message1, Object2 in turn sends ObjectN-1 the message message2, and so on.



### Example-01 Communication Example - Hotel Reservation

- Each message in a communication diagram has a sequence number.
- The top-level message is numbered 1.
- Messages sent during the same call have the same decimal prefix, but suffixes of 1, 2, etc. according to when they occur.



## Example-02 Library Item Overdue

A Library Management System is a software built to handle the primary housekeeping functions of a library. Libraries rely on library management systems to manage asset collections as well as relationships with their members. Library management systems help libraries keep track of the books and their checkouts, as well as members' subscriptions and profiles.

Library management systems also involve maintaining the database for entering new books and recording books that have been borrowed with their respective due dates.

