



NSCET E-LEARNING PRESENTATION

LISTEN ... LEARN... LEAD...





COMPUTER SCIENCE AND ENGINEERING

III YEAR / V SEMESTER

CS8592 OBJECT ORIENTED ANALYSIS AND DESIGN

A.DURAIMURUGAN , M.E, (Ph.d)

ASSISTANT PROFESSOR

Nadar Saraswathi College of Engineering & Technology,

Vadapudupatti, Annanji (po), Theni – 625531.





UNIT IV

DESIGN PATTERNS



GRASP

- The critical design tool for software development is a mind well educated in design principles. ☐It is not the UML or any other technology.
- A learning aid to help in the design of object-oriented software.
- Deciding : Object ☐What ☐Where ☐How

GRASP

- It stands for:
- General
- Responsibilities
- Assignment
- Software
- Patterns (Principles)

GRASP

- Learning Basic Object Design
- Creator 4 Learning Aid To understand object design To apply design Based on Patterns of assigning responsibilities
- A contract or obligation of a classifier Responsibilities are related to the obligations of an object in terms of its behavior. Knowledge Private state Computed state Behavior Send messages itself and modify its private state Instantiate another objects Send messages to another objects

Sequence Diagram Notations

“Doing” responsibilities Doing something itself, such as creation an object or doing a calculation. Initiating action in other objects ☐ Controlling and coordinating activities in other objects. “knowing” responsibilities Knowing about encapsulated data. Knowing about related objects. Knowing about things it can derive or calculate.

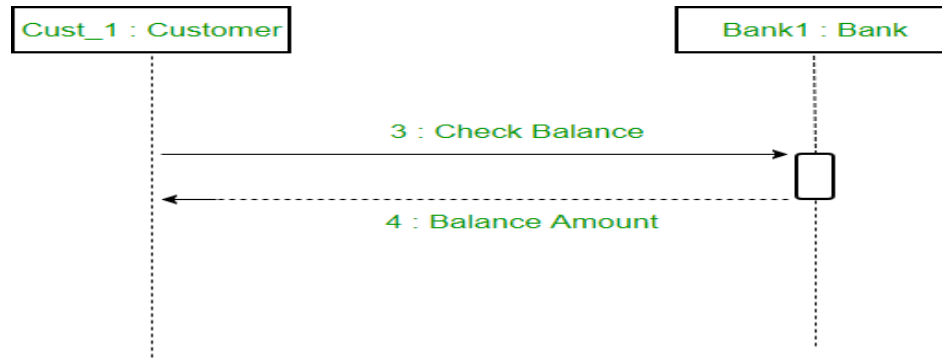
Sale is responsible for creating an object Payment ? Doing ? Sale is responsible for knowing its total Knowing

All patterns have suggestive names Naming a pattern, technique or principle has the following advantages:

- Facilities communication
- Support chunking and incorporating concepts into Understanding and memory

Difference between a lifeline and an actor

- A lifeline always portrays an object internal to the system whereas actors are used to depict objects external to the system. The following is an example of a sequence diagram:



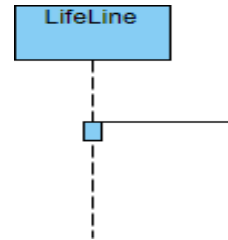
4)Activation Box

- Represents the time needed for an object to complete a task. The longer the task will take, the longer the activation box becomes.



MAPPING DESIGNS TO CODE

- An activation is represented by a thin rectangle on a lifeline represents the period during which an element is performing an operation. The top and the bottom of the of the rectangle are aligned with the initiation and the completion time respectively



Design Class Diagrams DCDs contain class or interface names, classes, method and simple attributes.

These are sufficient for basic class definitions.

- Elaborate from associations to add reference attributes.

REFERENCE ATTRIBUTES

Reference Attributes

An attribute that refers to another complex objects. ☐ Reference Attributes are suggested by associations and navigability in a class diagram.

Example:

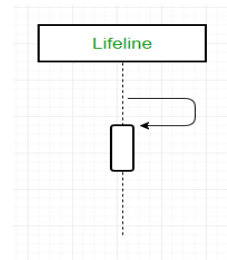
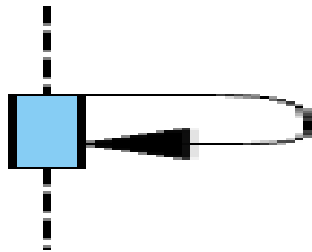
A product specification reference on a Sales Line Item. So here we can use product spec as a complex reference attribute to sales line item class.

- Asynchronous Messages
- An asynchronous message does not wait for a reply from the receiver.
- The interaction moves forward irrespective of the receiver processing the previous message or not. We use a lined arrow head to represent an asynchronous message.



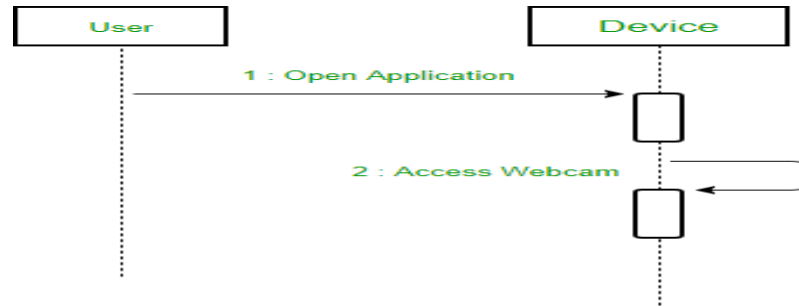
Message

- Certain scenarios might arise where the object needs to send a message to itself.
- Such messages are called Self Messages and are represented with a U shaped arrow.



Self Message

- For example – Consider a scenario where the device wants to access its webcam. Such a scenario is represented using a self message.

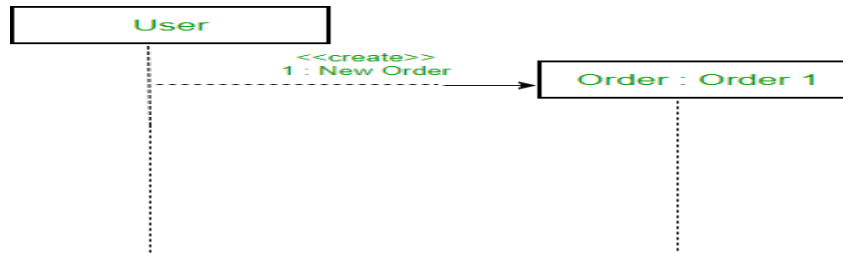


ROLE NAMES

- Each end of an association is a role.
- Reference Attributes are often suggested by role names. (use role names as the names of reference attributes).

Create Message example

- For example – The creation of a new order on a e-commerce website would require a new object of Order class to be created.



Destroy/ Delete Message example

- For example – In the scenario below when the order is received by the user, the object of order class can be destroyed.



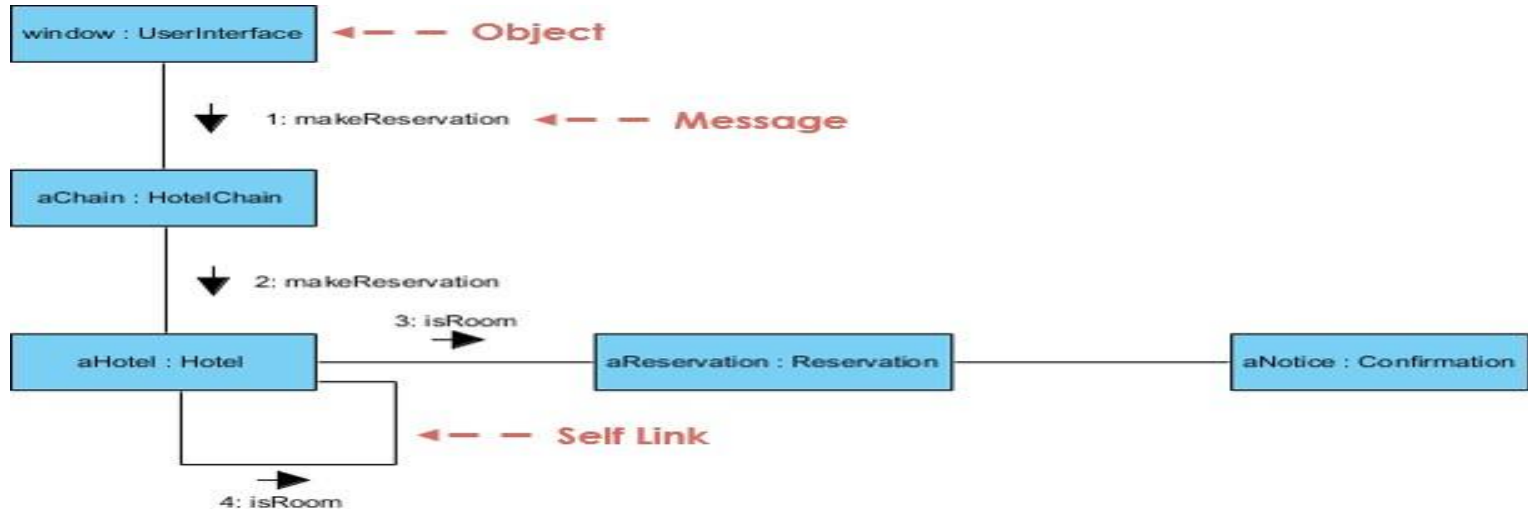
Why Collaboration Diagram?

- Unlike a sequence diagram, a collaboration diagram shows the relationships among the objects.
- Sequence diagrams and collaboration diagrams express similar information, but show it in different ways.
- The sequence diagram is used when time sequence is main focus.
- A collaboration diagram represents a collaboration, which is a set of objects related in a particular context, and interaction, which is a set of messages exchanged among the objects within the collaboration to achieve a desired outcome.

- Identify behavior whose realization and implementation is specified
- Identify the structural elements (class roles, objects, subsystems) necessary to carry out the functionality of the collaboration
 - Decide on the context of interaction: system, subsystem, use case and operation
- Model structural relationships between those elements to produce a diagram showing the context of the interaction
- Consider the alternative scenarios that may be required
 - Draw instance level collaboration diagrams, if required.
 - Optionally draw a specification level collaboration diagram to summarize the alternative scenarios in the instance level sequence diagrams

Collaboration Diagram?

The diagram



UML Statechart Diagram

- A state diagram describes the behavior of a single object in response to a series of events in a system.

Following are the main purposes of using Statechart diagrams:

- To model dynamic aspect of a system.
- To model life time of a reactive system.
- To describe different states of an object during its life time.
- Define a state machine to model states of an object.