**SEQUENCE DIAGRAMS**

* + A sequence diagram is an interaction diagram that emphasizes the time ordering of messages
  + It shows a set of objects and the messages sent and received by those objects
  + Sequence diagrams can be used to document Use Case Scenarios
  + Captures required objects early in analysis and verify object usage later in design
  + Shows the flow of messages from one object to another, and as such correspond to the methods and events supported by a class/object
  + Graphically, a sequence diagram is a table that shows objects arranged along the X axis and messages, ordered in increasing time, along the Y axis.

**Sequence Diagrams – Objects Symbols**

* An object in a sequence diagram is rendered as a box with a dashed line descending from it.

An Order Line

* The line is called the object lifeline, and it represents the existence of an object over a period of time

**Sequence Diagrams – Message Indicators**

An Order Line

A Stock Item

Check ()

[Check=”True”]

* Messages are rendered as horizontal {check()} arrows being passed from object to object as time advances down the object lifelines.
* Conditions (such as [check = “True”] indicate when a message gets passed.
* Notice that the bottom arrow is not solid, and there is no accompanying message, this indicates return from a previous message not new message

**Sequence Diagrams – Iteration Marker**

An Order Line

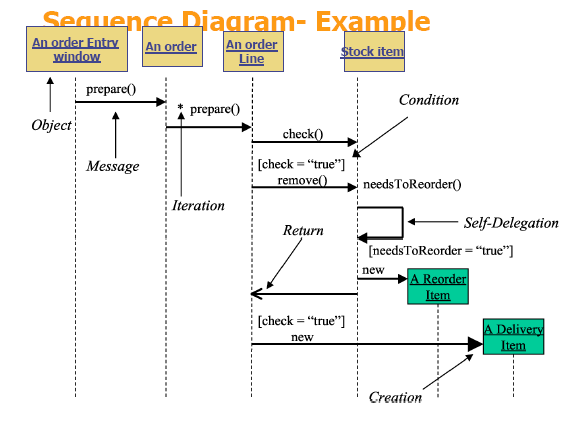
A Stock Item

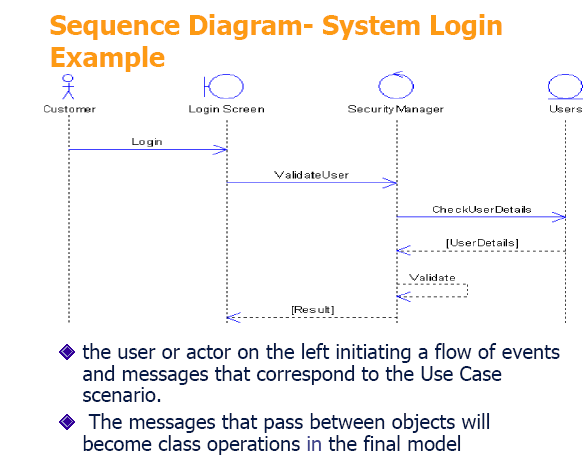
\*prepare ()

* An iteration marker, such as \* (as shown) or \*[I=1…n], indicates that a message will be repeated as indicated

Iteration Marker

**Sample Sequence Diagrams:**

**(a)**

**(b)**

**ACTIVITY DIAGRAM**

* Models the procedural flow of actions that are part of a larger activity
* Used to model system-level functions
* Focuses on the action sequence of execution and the conditions that trigger or guard those actions.
* Has a very similar notations to that of a state machine diagram

Customer Calls Ticket Office

* An action is indicated by a “capsule” shape – a rectangular object with semi-circular left and ends.
* The text inside it indicates the action (e.g., Customer Calls Ticket Office or Registration Office Opens).

First Action To Do

* The initial state is drawn as a solid circle with a transition line (arrow) that connects it to the first action in the activity’s sequence of actions.
* It is important to note that there can be only one initial state on an activity diagram and only one transition line connecting the initial state to an action.

Action 1

Action 2

The Figure above indicates *INCORRECT* rendering of an initial state within an activity diagram.

* Arrows indicate directions, the transition lines on an activity diagram show the sequential flow of actions in the modelled activity. It always point to the next action in the activity’s sequence.

**Activity Diagram – How a customer books a concert ticket.**

Customer Calls Ticket Office

Tickets Rep Asks What Event Person Wants Ticket

Customer Tells Rep Which Event

Ticket Rep Tells Customer Available Seats and Prices

Customer Tells Rep Seats

Ticket Rep Reserves Seats

Ticket Rep Asks For Credit Card and Billing Address

Customer Gives Requested Information

Ticket Rep Charges Credit Card

Ticket Rep Mails Tickets

The sample activity diagram documents the activity “Booking a Concert Ticket,” with actions in the following order:

1. Customer calls ticket office.
2. Ticket rep asks what event person wants ticket for.
3. Customer tells rep event choice.
4. Ticket rep tells customer available seats and prices.
5. Customer tells rep seating choice.
6. Ticket rep reserves seats.
7. Ticket rep asks for credit card and billing address.
8. Customer gives requested information.
9. Ticket rep charges credit card.
10. Ticket rep mails tickets.

The action order is clear from the diagram because it shows an initial state (starting point), and from that point one can follow the transition lines as they connect the activity’s action. It is possible for an activity diagram to show multiple final sates. Unlike initial state symbols, of which there can be only one on an activity.

**Decision points**

Typically, decisions need to be made throughout an activity, depending on the outcome of a specific prior action.

Customer Orders Drink

[Drink is Alcoholic]

[else]

Make Sure Customer is at Least 21 years Old

Get Drink for Customer

* Each transition line involved in a decision point must be labelled with text above it to indicate “guard conditions,” commonly abbreviated as guards.
* Guard condition text is always placed in brackets—for example, [guard condition text]
* A guard condition explicitly tells when to follow a transition line to the next action

**Merge points**

Sometimes the procedural flow from one decision path may connect back to another decision path. In these cases we connect two or more action paths together using the same diamond icon with multiple paths pointing to it, but with only one transition line coming out of it. This indicate a *merge* decision point.

Tell Customer to Order Non-Alcoholic Drink

[Customer’s Age <21]

Make Sure Customer is at Least 21 years Old

Customer Orders Drink

[else]

[Customer’s Age >=21]

[Drink is Alcoholic]

Get Drink for Customer

A partial activity diagram, showing two decision points:

(“Drink is alcoholic” and “Customer’s Age <21”) and one merge

(“else” and “Customer’s Age >=21”)