Lecture 6
Activity Diagram

Chapter 20
The Unified Modeling Language User Guide
SECOND EDITION
By Grady Booch, James Rumbaugh, Ivar Jacobson

Session 14
UML Weekend Crash Course
Thomas A. Pender

Introduction

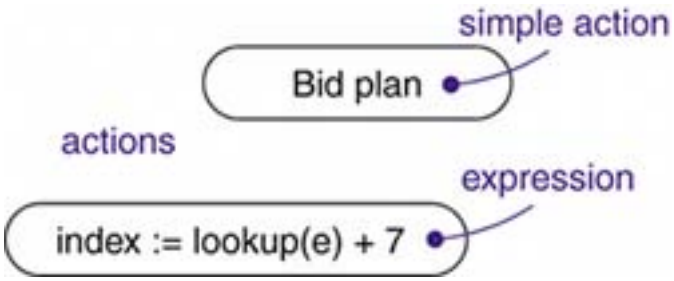
- An activity diagram is essentially a flowchart, showing flow of control from activity to activity.
- Unlike a traditional flowchart, an activity diagram shows concurrency as well as branches of control.
- You use activity diagrams to model the dynamic aspects of a system. For the most part, this involves modeling the sequential and concurrent steps in a computational process.
- An **activity** is a behavior that specifies the sequence of steps a computational process performs.
- Behavioral thing that focuses on the flows among computational steps without regard to which object performs each step.
- A step of an activity is called an **action**.

Elements

- Activity diagrams commonly contain:
  - Actions
  - Activity nodes
  - Flows
  - Object values

Actions and Activity Nodes

- Actions: Atomic computations are called actions, like, call an operation on an object, send a signal to an object, or even create or destroy an object. Actions can't be decomposed (actions are atomic), meaning you can't execute part of an action; either it executes completely or not at all.

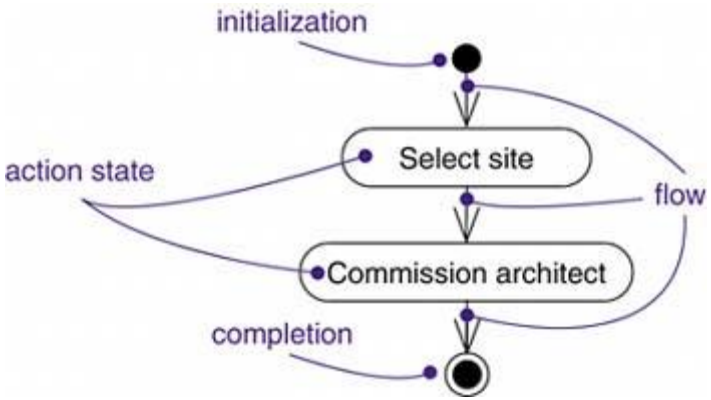


Actions and Activity Nodes

- Activity: An activity node is an organizational unit within an activity. In general, activity nodes are nested groupings of actions or other nested activity nodes. Like, evaluate some expression that sets the value of an attribute or that returns some value.

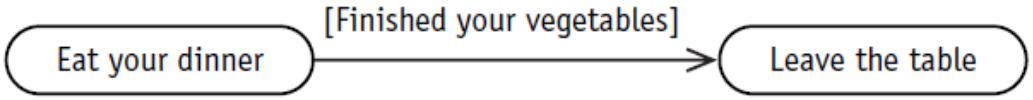
Control Flow

- Control Flow: A flow is represented as a simple arrow from the predecessor action to its successor, without an event label. A flow of control has to **start** and **end** someplace.

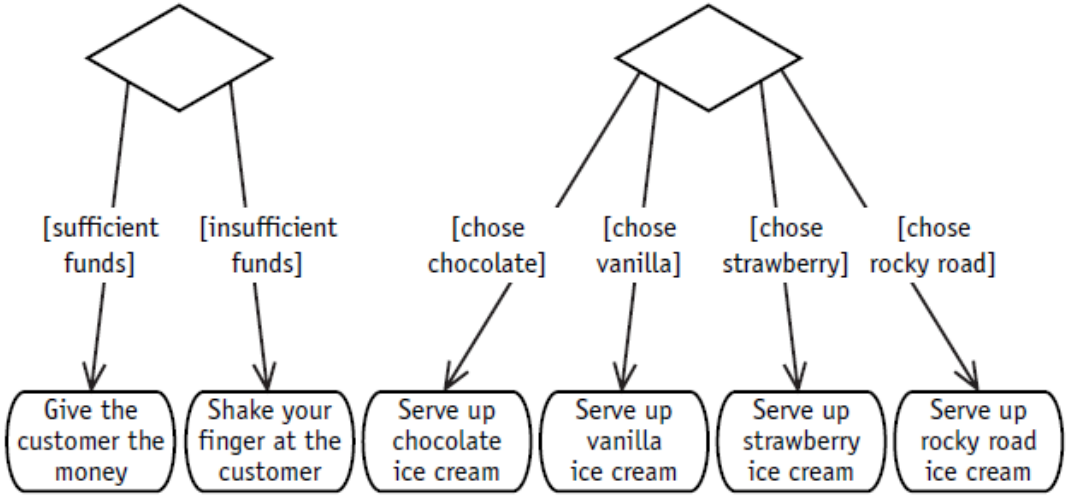


Guard condition and decision

- Guard Condition:

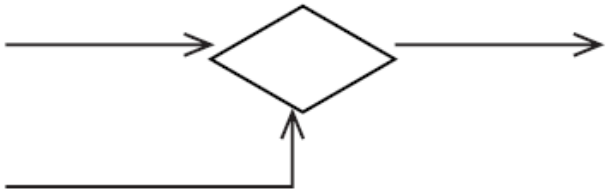


- Decision: (One incoming and multiple alternate outgoing)



Merge Point, Start and End

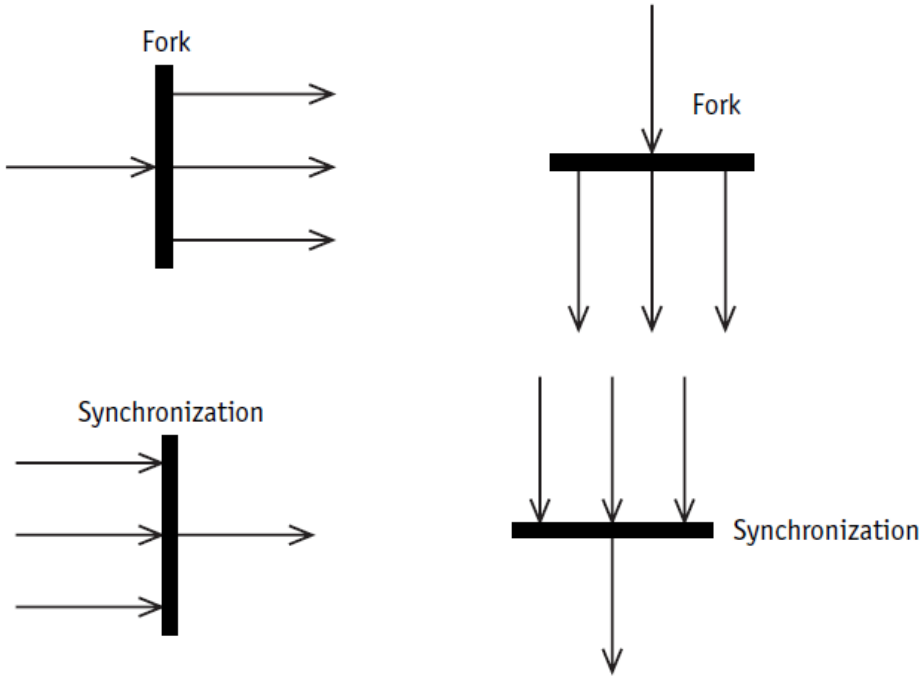
- Merge Point: (Multiple alternate incoming and one outgoing)



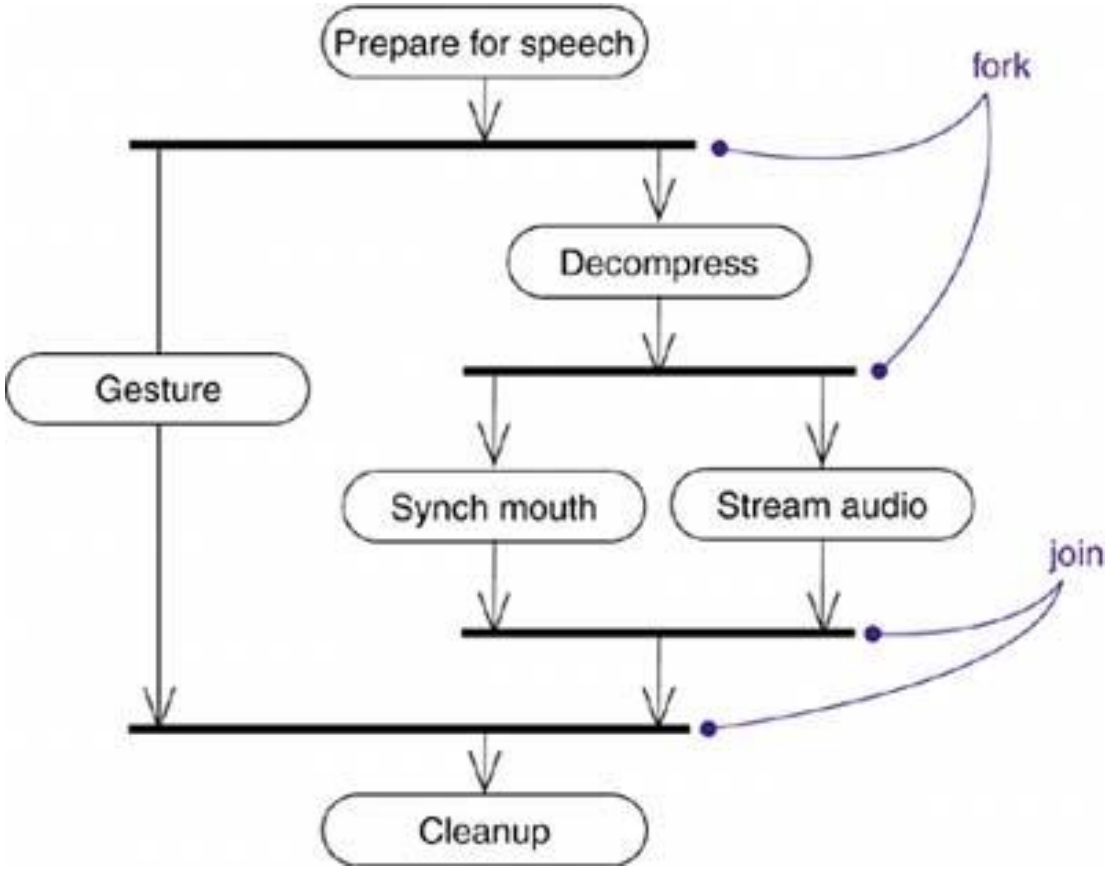
- Start and End (must be present in activity diagram):

Concurrency

- Forking and Synchronization: Number of outgoing concurrent flows out of a forking must match with the number of incoming concurrent flows in its respective synchronization (joining).
- Forking: (One incoming and multiple parallel outgoing)
- Synchronization/ Joining: (Multiple parallel incoming and one outgoing)



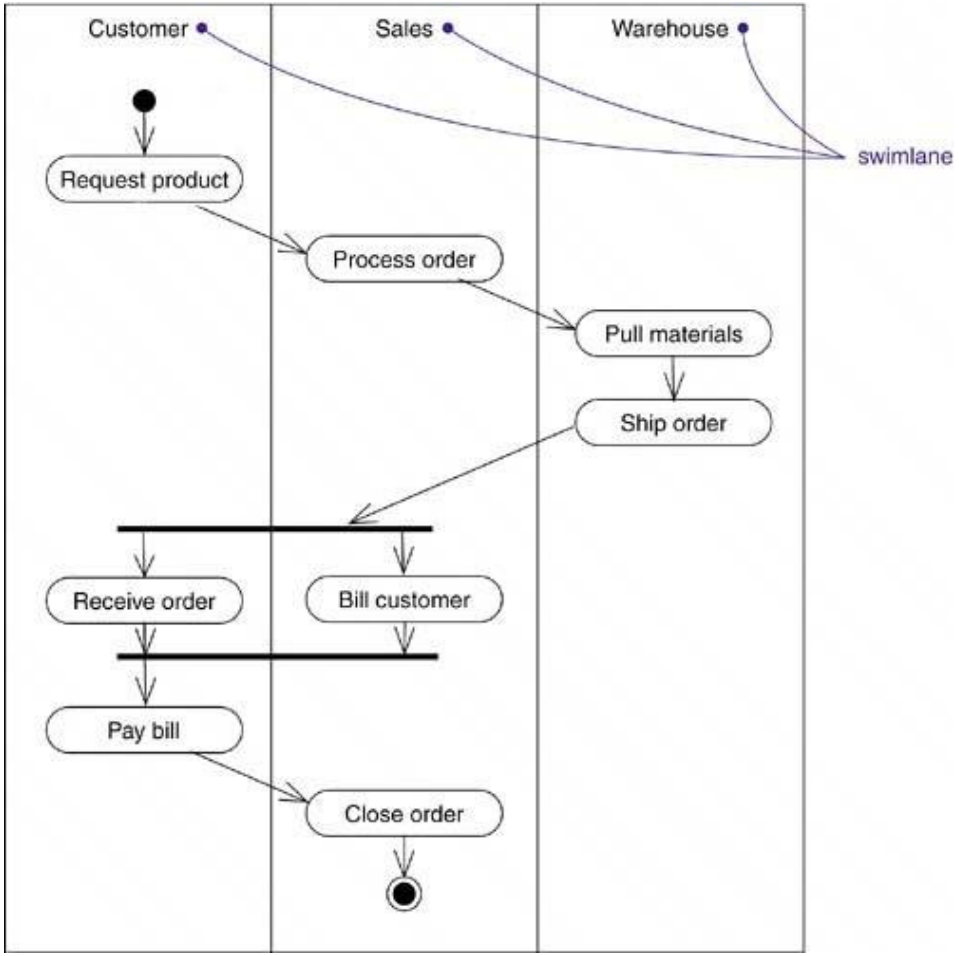
Concurrency (Example)



Swimlane

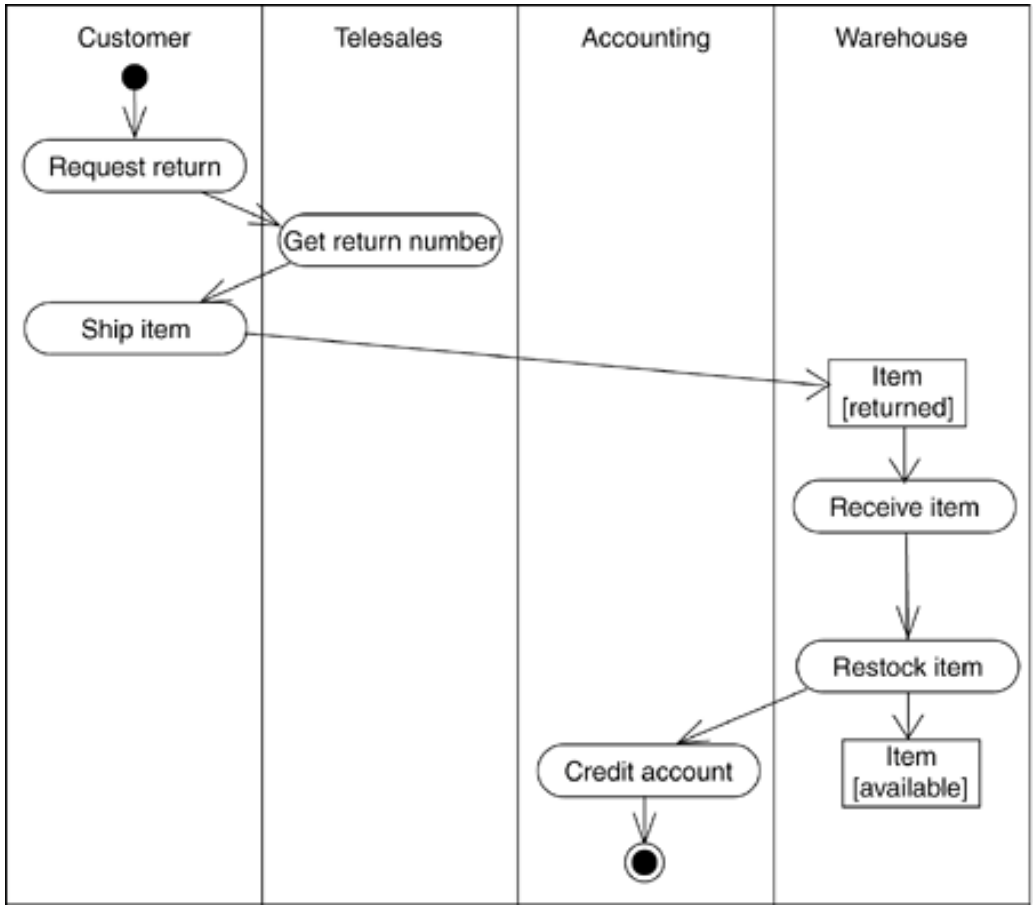
- Swimlane: Modeling workflows of business processes, to partition the activity states on an activity diagram into groups, each group representing the business organization responsible for those activities. Each swimlane has a name unique within its diagram. A swimlane really has no deep semantics, except that it may represent some real-world entity, such as an organizational unit of a company.

Swimlane (Example)



Object Flow

- Objects may be involved in the control flow associated with an activity. Usually how the **state of object changes** is shown with object flow.

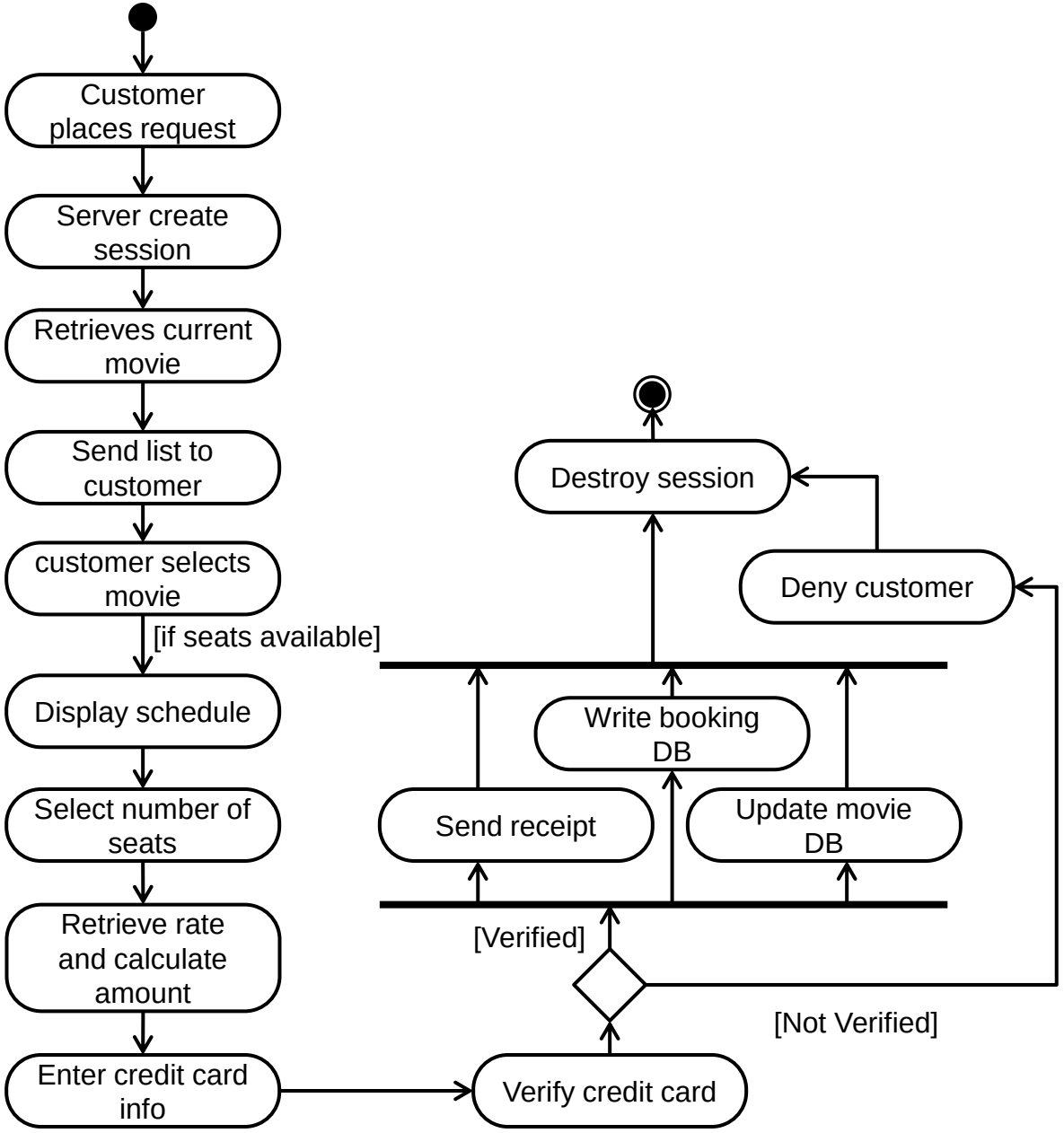


Case Studies

- **Case 1**
- In an online movie ticket booking system a customer books ticket using an interface. After the customer places a request for current movies, the server creates a Session object which then requests a list of currently running movies from the movie database. The list is then sent to customer. Customer then selects the desired movie and the Session object retrieves the available timing of the selected movie. Before displaying the time to the customer each scheduled time of the movie is checked in the Booking Database for the availability of the seats. Only the time schedules in which seats are available are displayed to the customer. The customer then selects the number of seats he wants to book. The rate of the ticket is taken from the Movie Database and the due amount is calculated by the Session object. Then the customer enters his credit card detail which is verified by a creditCardAgent. If the card is verified the session object sends a receipt to the customer, writes in the booking database and update the movie database at the same time. If the card is not verified the customer request is denied. The server finally destroys the session object.

Case 1: Solution

In an online movie ticket booking system a customer books ticket using an interface. After the customer places a request for current movies, the server creates a Session object which then requests a list of currently running movies from the movie database. The list is then sent to customer. Customer then selects the desired movie and the Session object retrieves the available timing of the selected movie. Before displaying the time to the customer each scheduled time of the movie is checked in the Booking Database for the availability of the seats. Only the time schedules in which seats are available are displayed to the customer. The customer then selects the number of seats he wants to book. The rate of the ticket is taken from the Movie Database and the due amount is calculated by the Session object. Then the customer enters his credit card detail which is verified by a creditCardAgent. If the card is verified the session object sends a receipt to the customer, writes in the booking database and update the movie database at the same time. If the card is not verified the customer request is denied. The server finally destroys the session object.



Case Studies

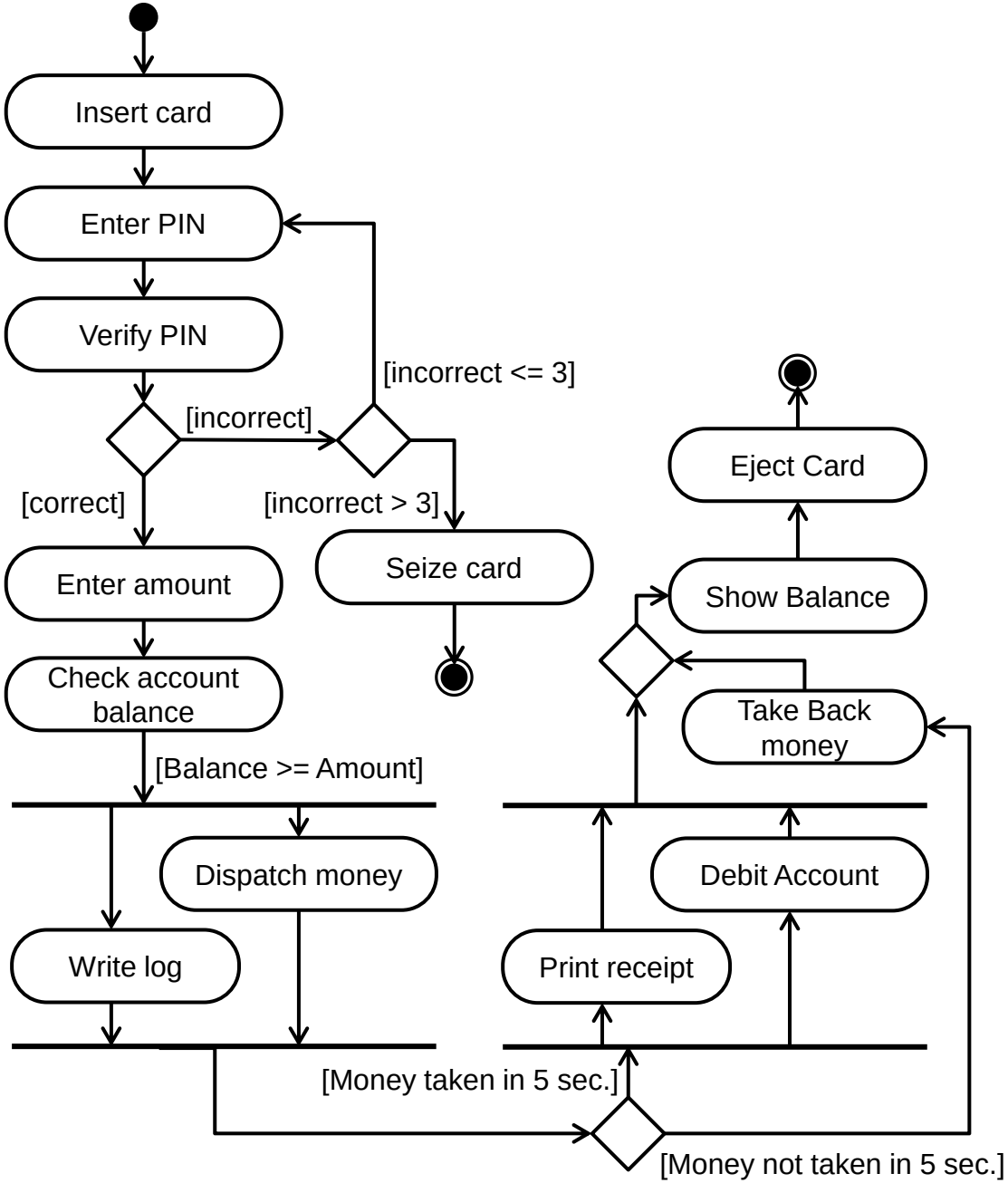
- **Case 2**
- In an online airlines ticket booking system, a customer places a request to the system selecting the departure and arrival destinations and also the date of departure. The system then gets all the carrier names and their time schedule from the flight database and displays to the customer. If the customer likes any one of the options, he selects the option. But, if the customer doesn't like any option he can go back and select a new date of departure or he can close the application. After selecting an option, the system asks for customer details and credit card information. The system verifies the credit card. If the credit card is valid, the system writes all the booking information in the booking database and sends customer information to the carrier simultaneously. If the verification of the credit card fails, the system sends an error message to the customer and requests for credit card information again. The verification process is done again. If the credit card verification fails more than three times the system cancels the booking process and sends a warning message to the credit card agent at the same time. Once the booking is done the system passes the message to the interface and the interface generates a bill and sends it to the customer.

Case Studies

- **Case 3**
- In an ATM machine a customer starts a withdrawal transaction by inserting the card. Then he enters the pin, which is verified by the bank. If the pin is incorrect the machine requests for the pin again and the customer enters pin number. The verification repeated for 3 times for an incorrect pin number. If the pin is incorrect even in 4^{th} attempt, the card is seized by the machine and the transaction is closed. If the pin is correct customer enters the amount he wishes to withdraw. The bank then checks the account balance of the customer. If the balance is greater than or equal to the withdrawal amount, then money is dispatched through the machine and the information is written in the log concurrently. If the money is taken form the slot within 5 seconds, the account is debited, and a transaction receipt is printed simultaneously. If money is not taken in 5 seconds, it is taken back by the machine. Then the balance is shown to the customer. The card is then ejected, and the transaction is completed.

Case 3: Solution

In an ATM machine a customer starts a withdrawal transaction by inserting the card. Then he enters the pin, which is verified by the bank. If the pin is incorrect the machine requests for the pin again and the customer enters pin number. The verification repeated for 3 times for an incorrect pin number. If the pin is incorrect even in 4th attempt, the card is seized by the machine and the transaction is closed. If the pin is correct customer enters the amount he wishes to withdraw. The bank then checks the account balance of the customer. If the balance is greater than or equal to the withdrawal amount, then money is dispatched through the machine and the information is written in the log concurrently. If the money is taken form the slot within 5 seconds, the account is debited, and a transaction receipt is printed simultaneously. If money is not taken in 5 seconds, it is taken back by the machine. Then the balance is shown to the customer. The card is then ejected, and the transaction is completed.



Case Studies

- **Case 4 (Activity Diagram with Object Flow)**
- A regular customer requests for product. The sales department processes the order by creating a new order object in empty state. Materials of the order are pulled out of the inventory at the warehouse and the order is set as filled. Products are then shipped by the warehouse and sales department prepares the bill at the same time. If the customer pays for the bill within a week sales department closes the order in paid state. If the customer doesn’t pay the bill within one week the sales department sends warning to the customer changing his state to defaulter and closes the order in unpaid state.

Case 4: Solution

(Activity Diagram with Object Flow)

A regular customer requests for product. The sales department processes the order by creating a new order object in empty state. Materials of the order are pulled out of the inventory at the warehouse and the order is set as filled. Products are then shipped by the warehouse and sales department prepares the bill **at the same time**. If the customer pays for the bill within a week sales department closes the order in paid state. If the customer doesn't pay the bill within one week the sales department sends warning to the customer changing his state to defaulter and closes the order in unpaid state.