Interaction Modeling and Advanced Interaction Modeling

UNIT 3

Interaction Modeling

Interaction Modeling

Introduction:

- The **class model** describes the **objects in a system and their relationships.**
- The **state model** describes **life cycle of the objects.**
- The **interaction model** describes **how the objects interact.**

Cont…

- ***The interaction model starts with use cases that are then elaborated with sequence and activity diagram.***

1. **Use cases diagram:-**
   - Focuses on functionality of a system i.e, What a system does for users?
2. **Sequence diagram:-**
   - shows the object that interact and the time sequence of their interaction.
3. **Activity diagram:-**
   - Elaborates important processing steps.

Use Case Modeling

Use Case Modeling

- **Use cases model** → **User's view of the functionality of a system**.
  - What the system does as far as the user is concerned; what it does that is of value to the user.

- It provides a means of organizing, structuring and documenting the **information** discovered during **requirements elicitation**.
- It is an ideal vehicle for discussions with the user and for clarifying the developer's understanding of the user's requirements.
- Later on technical details relating , for ex: to design of the user interface, are added for the information to the programmer.
- Use cases models are presented in a graphical form, the **Use case diagram**.

- Use case model typically proceeds as follows:

➢Find a candidate system boundary

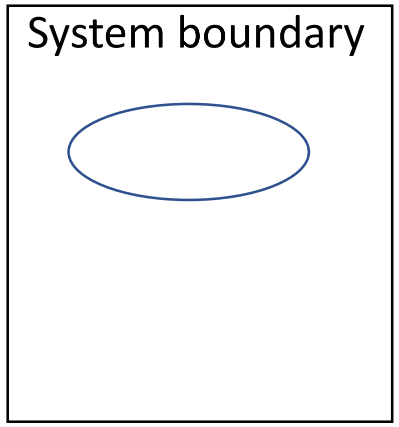
➢Find the actors

➢Find the use cases – specify system/product's functionalities

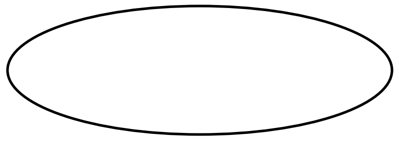
➢Iterate until use cases, actors and the system boundaries are stable.

UML:-Use Case diagram elements.

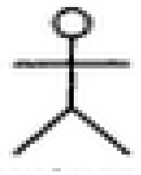
- System boundary



- Use case:



Actor:



- Communication Association:
  (use case relationship)

Use Case Models

Actor:- a stick figure labelled with the name of the actor.

Eg. Administrator, Receptionist, verification system etc.

An actor is a direct external user of a system.

- Each actor represents those objects that behave in a particular way towards the system.

- Examples:

1.Travel agency system

- Actors:-Traveler, Agent and Airline.

2.Vending machine

- Actors:-Customer, Repair technician, stock clerk.

Cont…

- An actor can be **person, device** and **other system,** anything that **interacts directly** with the system.

- An object can be bound to multiple actors if it has different facts to its behavior.

  - Eg:- Maya, Vinod, Latha may be customers of a vending machine, Vinod may also be the repair technician interacting with the machine.

Cont…

- The actor is directly connected to the system.
  - Eg.. The **dispatcher** of the **repair technician** from a **service bureau** is not an actor of a vending machine, only the repair technician interacts directly with the machine and he is an actor.

Use Cases:-

- A use case is a **coherent piece of functionality** that a system can provide **by interacting with actors**.
- We start each use case name with a verb since it represent processes.

for Ex. 'Maintain customer list' rather than 'customer list'

'Handle enquires' rather than 'enquiries'.

- Eg:

1. A customer can **buy a beverage** from a vending machine.
2. A repair technician can perform **scheduled maintenance** on a vending machine.

- Each use case involves one or more actors as well as the system itself.

  - Eg.:-In a telephone system, the use case make a call involves two actors, a caller and a receiver.

Cont…

- The actors need not be all humans.
  - Eg:-The use case make a trade on an online stock broker involves a **customer actor** and a **stock exchange** actor.
- An use case involves a sequence of messages among the system and its actors. This can be repeated several times.
  - Eg:-In the buy a beverages use case, the customer first inserts a coin and then vending machine displays the amount deposited.

Cont…

- Define a mainline behavior sequence first, then optional subsequences, repetitions and other variations.
- Eg:-A customer can deposit a variable number of coins in the "buy a beverage" use case, depending on the money inserted and the item selected the machine may or may not return change.
- Error condition are also part of a use case.
- Eg:- If a customer selects a beverage whose **supply is exhausted**, the vending machine displays a **warning menu.**

Use Case Diagram

- A system involves a **set of use cases** and **a set of actors.**

- **Set of use cases**:- shows the complete functionality of the system at some level of detail.

- **Set of actors** :- represents the complete set of objects that the system can serve.

Eg:-

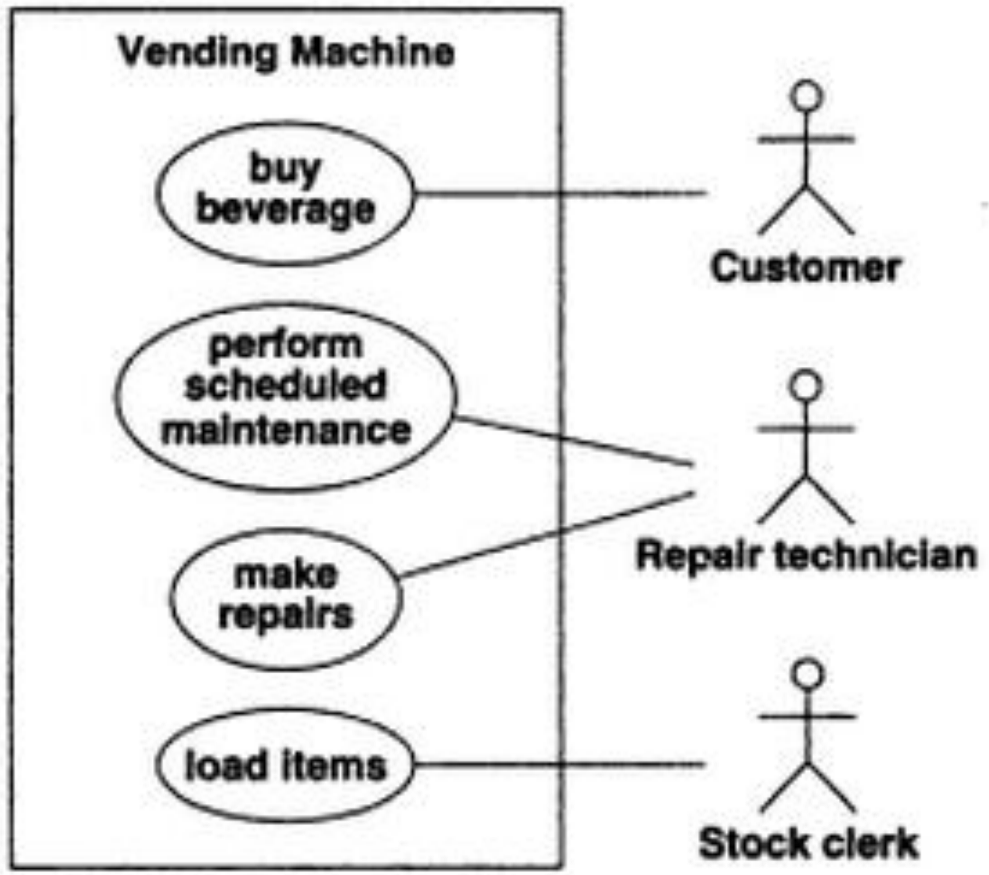


Figure 7.3 Use case diagram for a vending machine. A system involves a set of use cases and a set of actors.

Guidelines for use case models

1. First determine the system boundary:-It is impossible to identify use cases or actors if the system boundary is un clear.

2. Ensure that actors are focused:-Each actor should have a single, coherent purpose.

3. Each use case must provide value to users:-use case should represent complete transaction that provide value to user and should not define too narrowly.

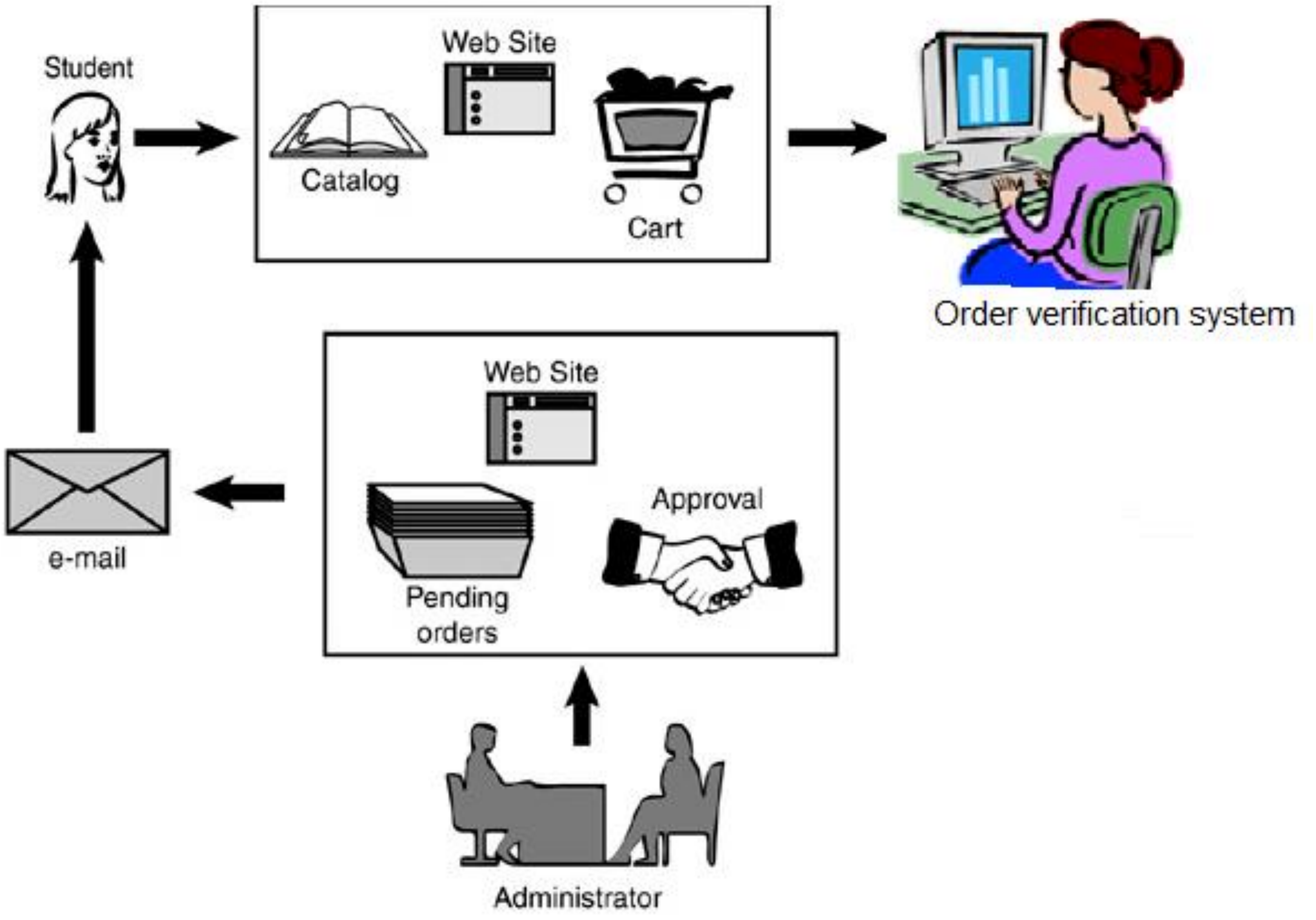
Cont…

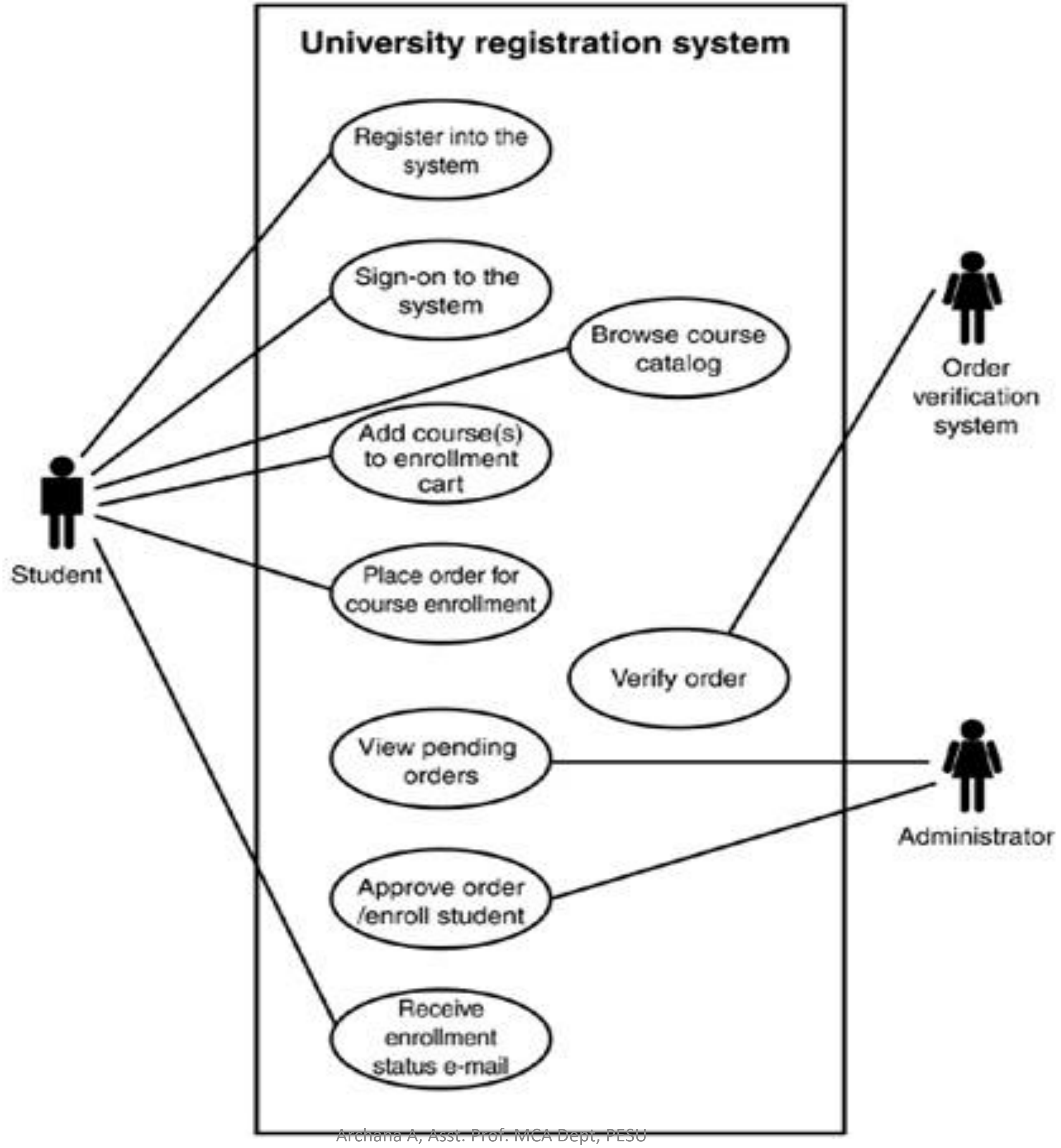
4. Relate use case and actors:-every use case should have at least one actor and every actor should participate in atleast one use case.

5. Remember that use cases are informal:-do not obsess formalism in specifying use cases.

6. Use cases can be structured:-For large systems, use cases can be built out of smaller fragments using relationship.

University Registration System




University registration system
Register into the system
Sign-on to the system
Browse course catalog
Add course(s) to enrollment cart
Place order for course enrollment
Verify order
View pending orders
Approve order /enroll student
Receive enrollment status e-mail
Student
Order verification system
Administrator

Write Use case diagram for the following

1. Online cab booking system
2. Car pooling App → Quick ride
3. Hotel Room booking application → oyo
4. e-billing
5. Platform ticket generation system

Sequence Modeling

Sequence Models

- The Sequence model elaborates the theme of use cases.

- There are two kind of sequence models
  - Scenarios.
  - Sequence diagrams - more structured format.

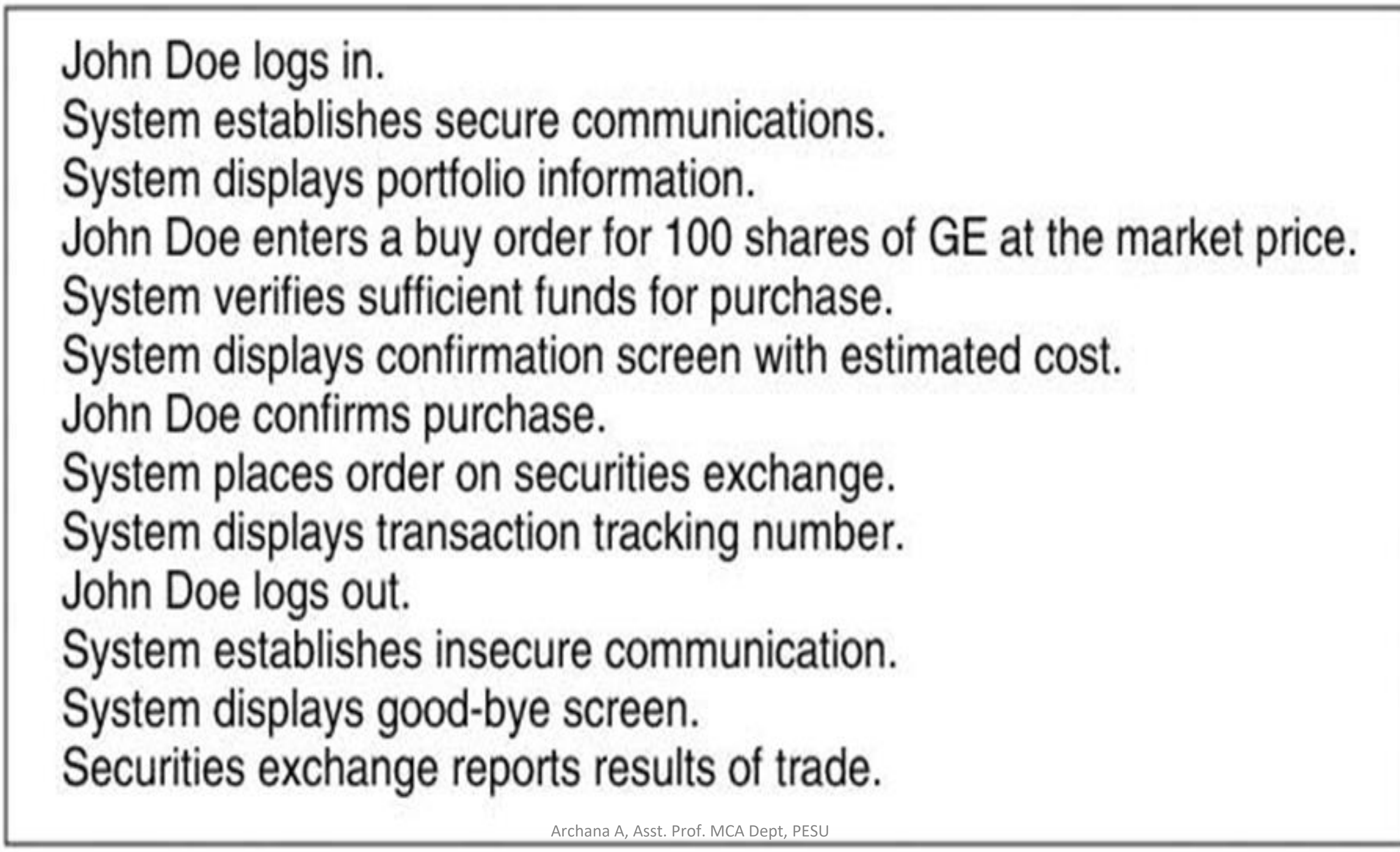
1. Scenario:

- A scenario is a sequence of events that occurs during one particular execution of a system, such as for a use case i.e. for each functionality scenario written.

- A scenario can be displayed as a list of text statements.

Scenario for a session with online Stock Broker

Go To


John Doe logs in.
System establishes secure communications.
System displays portfolio information.
John Doe enters a buy order for 100 shares of GE at the market price.
System verifies sufficient funds for purchase.
System displays confirmation screen with estimated cost.
John Doe confirms purchase.
System places order on securities exchange.
System displays transaction tracking number.
John Doe logs out.
System establishes insecure communication.
System displays good-bye screen.
Securities exchange reports results of trade.

Scenario Cont…

- At early stages of development, Scenarios are expressed at a high level, later stages we can show exact messages.

- First step of writing a scenario is to identify the objects exchanging messages.
- Determine the sender and receiver of each message and sequence of messages.
- Add activities for internal computations.

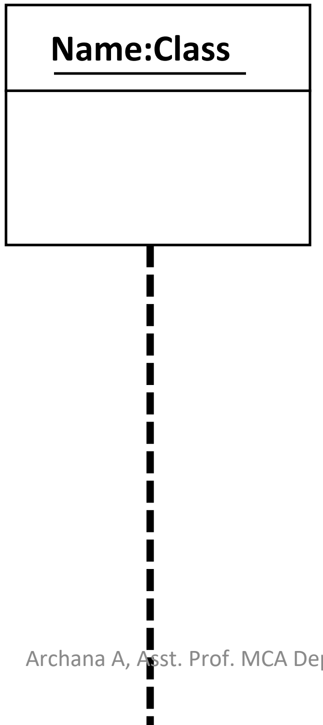
2. Sequence diagram

- A sequence diagram shows the participants in an interaction and the sequence of messages among them.

Components of sequence diagram:

Object:

- Objects are represented by rectangles and **name of the objects are underlined**
- Object life line are denoted as **dashed lines**. They are used to model the existence of objects over time

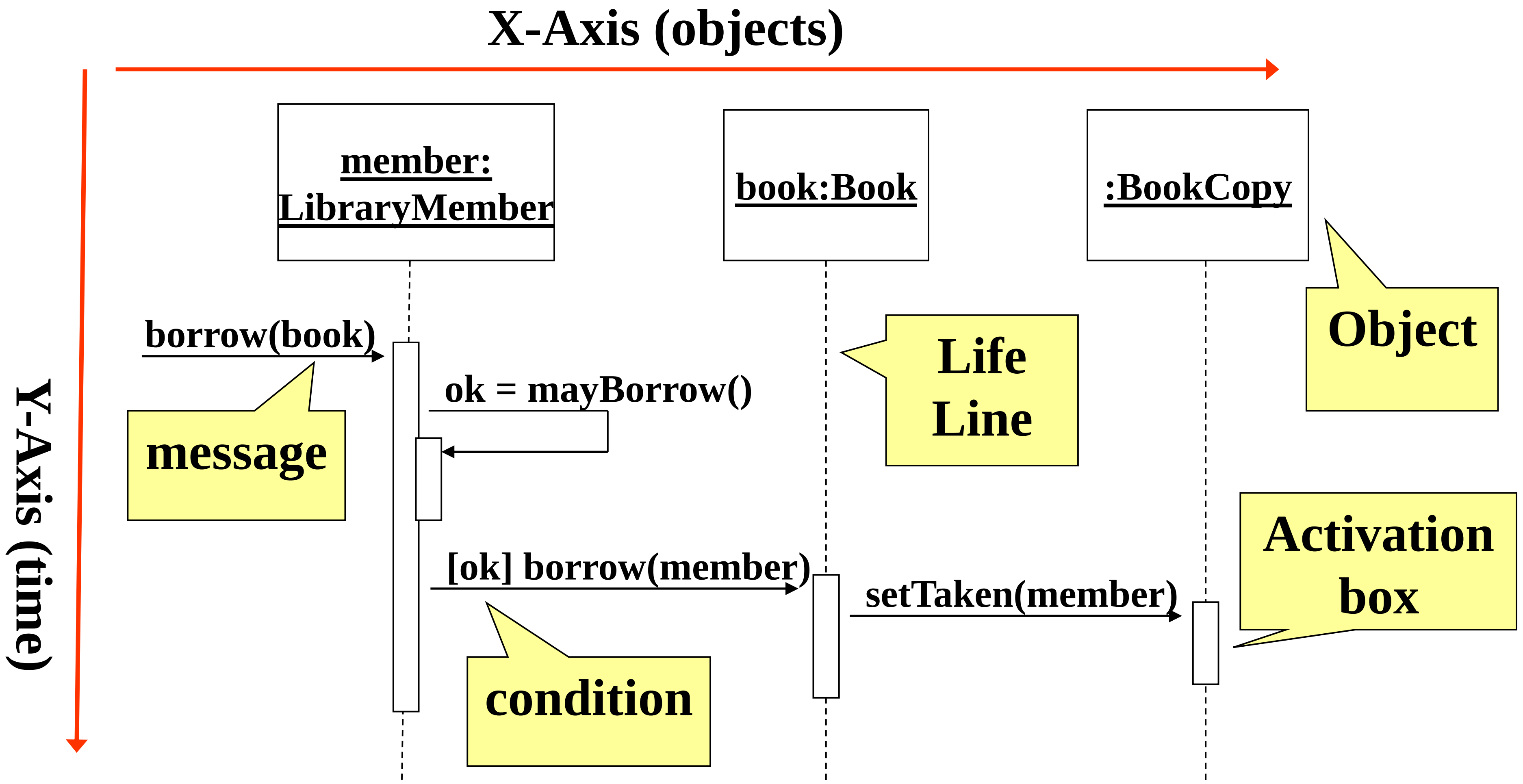


Cont…

- Each actor as well as the system is represented by a vertical line called lifeline and each message by a horizontal arrow from sender to the receiver.
- Each use case requires one or more sequence diagram to describe its behavior.
- For large scale interactions containing many independent tasks, we can draw separate sequence diagram for each task.
- Eg:-sequence diagram for a stock purchase

  - Sequence diagram for a stock quote
  - Sequence diagram for a stock purchased **that fails**.

A Sequence Diagram example



Example 2: sequence diagram

- Eg shows sequence diagram corresponding to previous stock broker scenario.

Eg:-

Go to

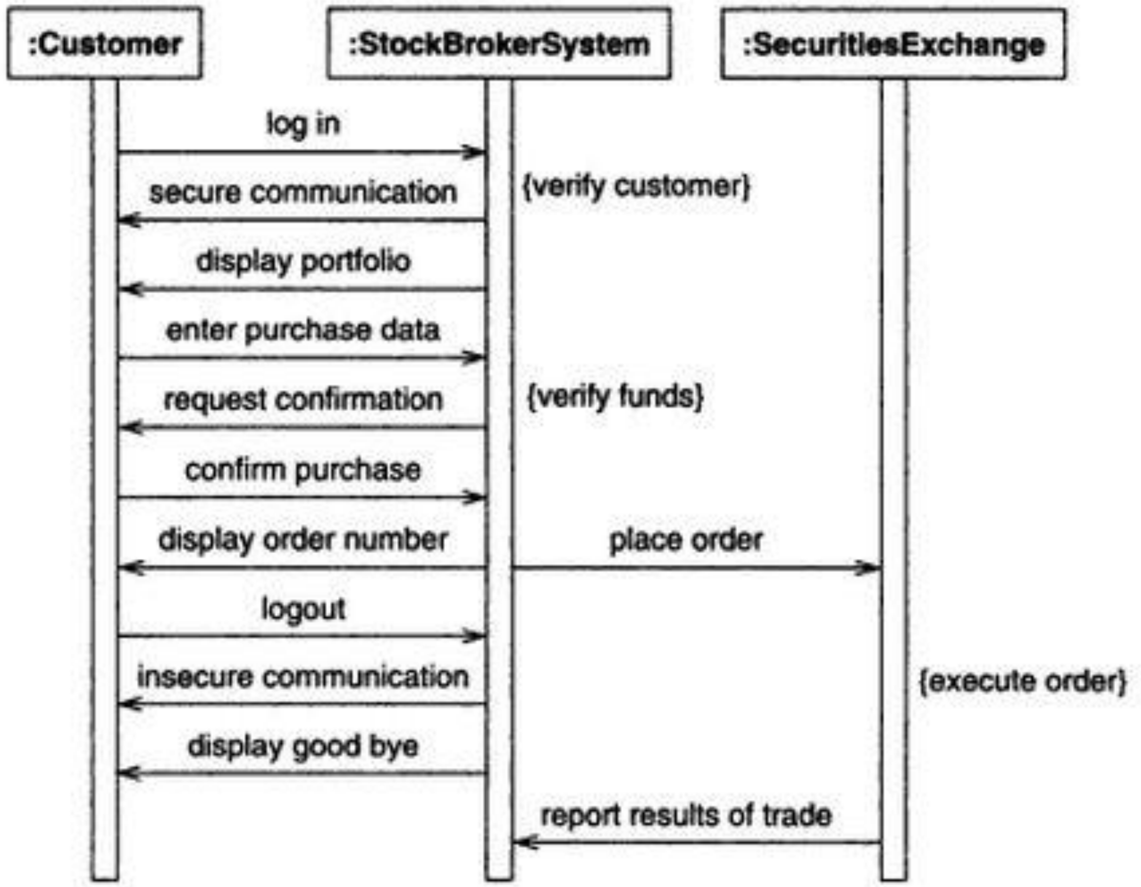


Figure 7.5 Sequence diagram for a session with an online stock broker. A sequence diagram shows the participants in an interaction and the sequence of messages among them.

MESSAGES:

- In message **square brackets** containing **a guard conditions**. This is a Boolean condition that must be satisfied to enable the message to be sent.
- May have an **asterisk followed by square brackets** containing an **iteration specification**. This specifies the number of times the message is sent.
- May have **return list** consisting of a **comma -separated** list of names that designate the values of returned by the operation.
- Must have **a name or identifier** string that represents the message.
- May have **parentheses** containing an **argument list** consisting of a comma separated list of actual parameters passed to a method.

Message Types

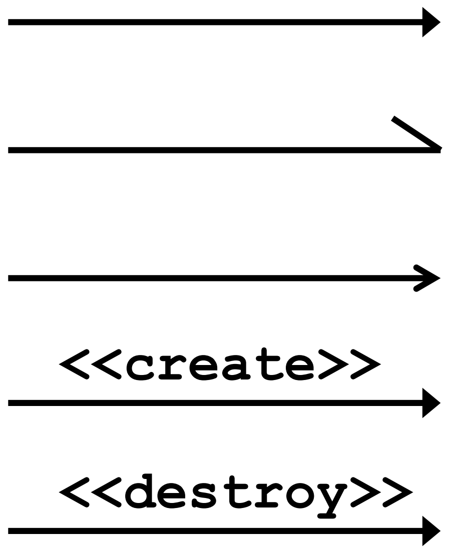
Synchronous

Asynchronous

Simple

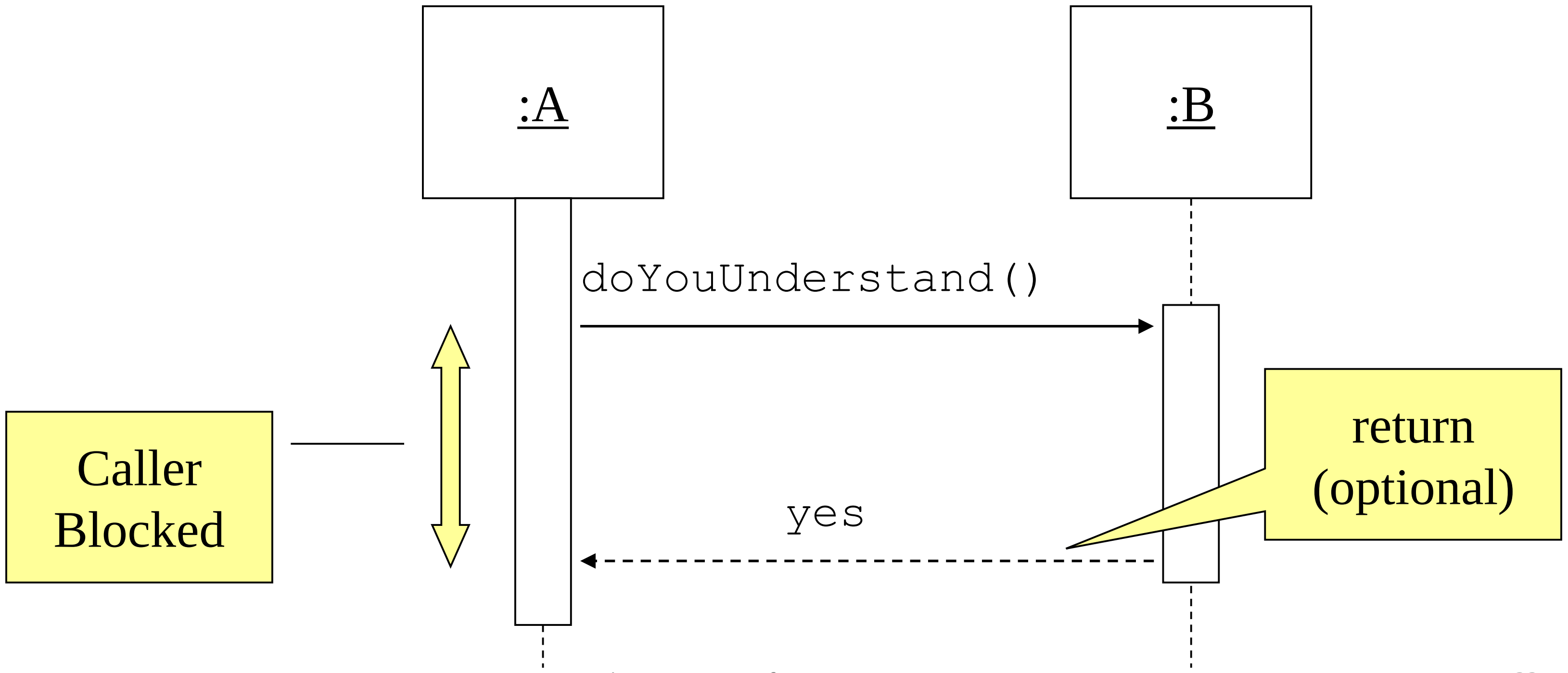
Create

Destroy



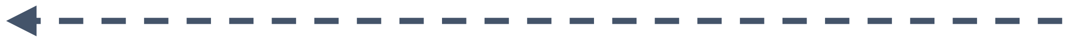
Synchronous Messages

- Nested flow of control, typically implemented as an operation call.
  - The routine that handles the message is completed before the caller resumes execution.



Return Values

- Optionally indicated using a dashed arrow with a label indicating the return value.



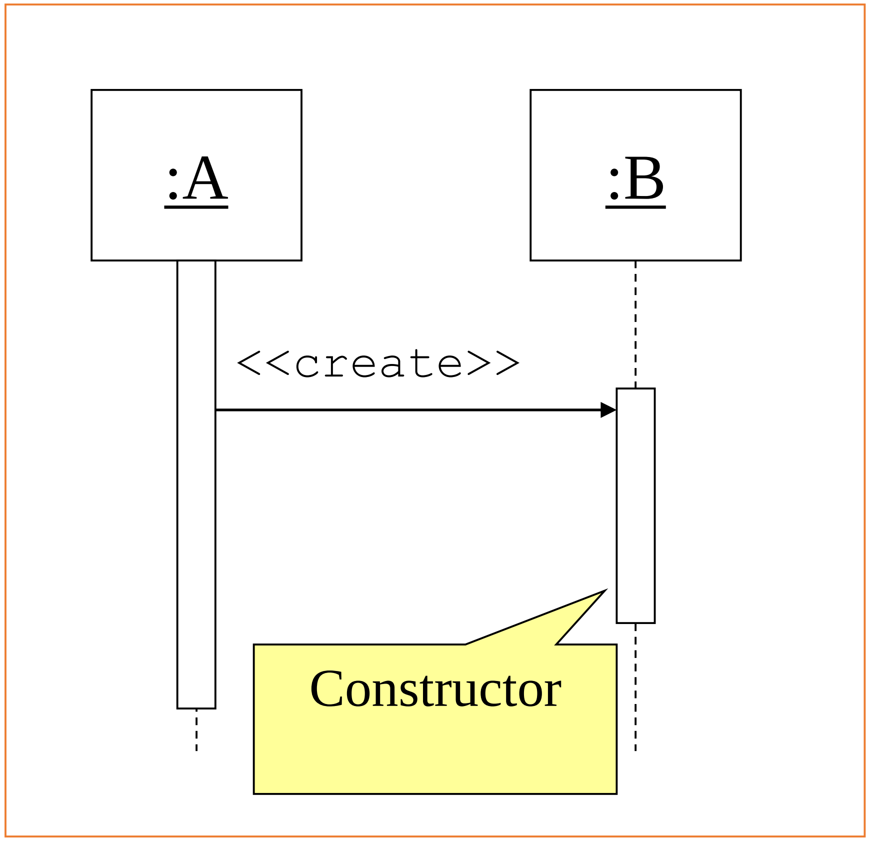
  - Don’t model a return value when it is obvious what is being returned, e.g. getTotal()

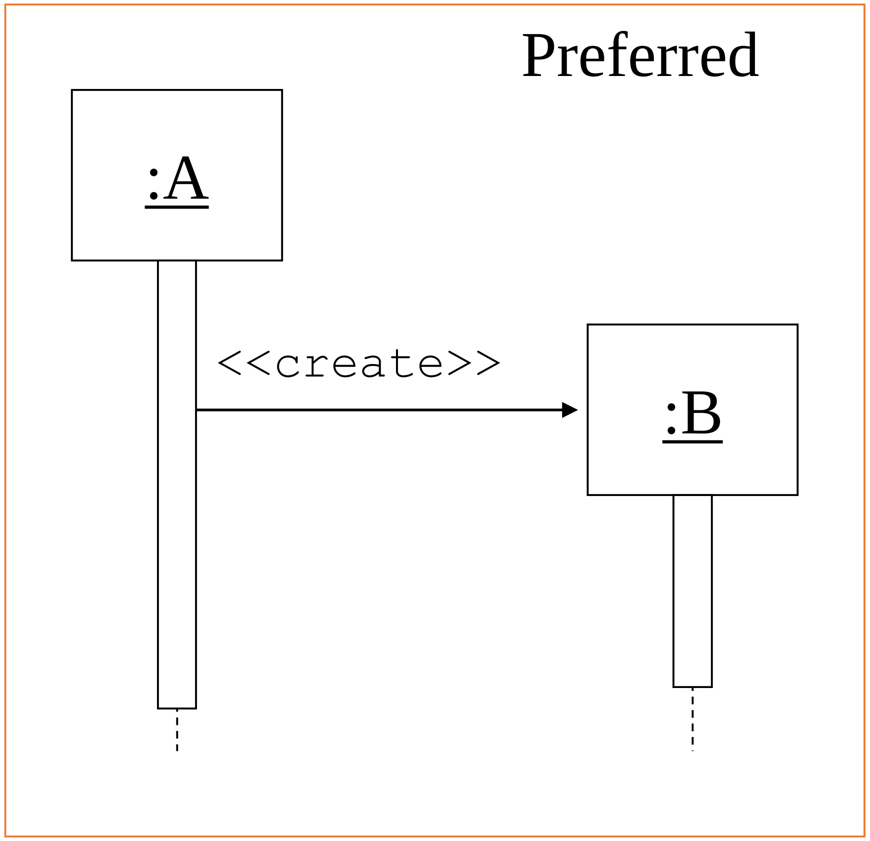
  - Model a return value only when you need to refer to it elsewhere, e.g. as a parameter passed in another message.

  - Prefer modeling return values as part of a method invocation, e.g. ok = isValid()

Object Creation

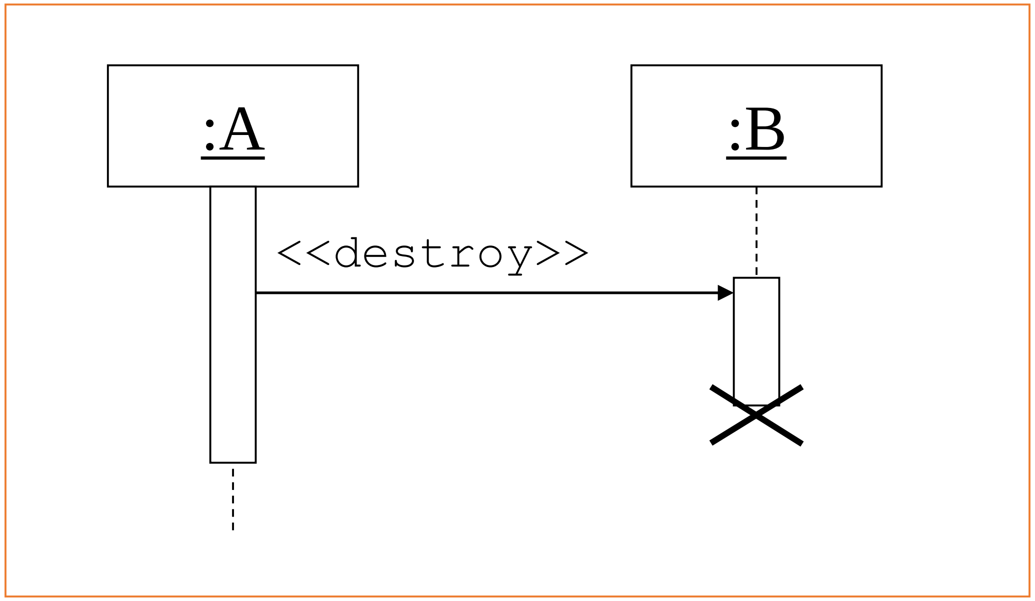
- An object may create another object via a **<<create>>** message.





Object Destruction

- An object may destroy another object via a <<destroy>> message.
  - An object may destroy itself.
  - Avoid modeling object destruction unless memory management is critical.

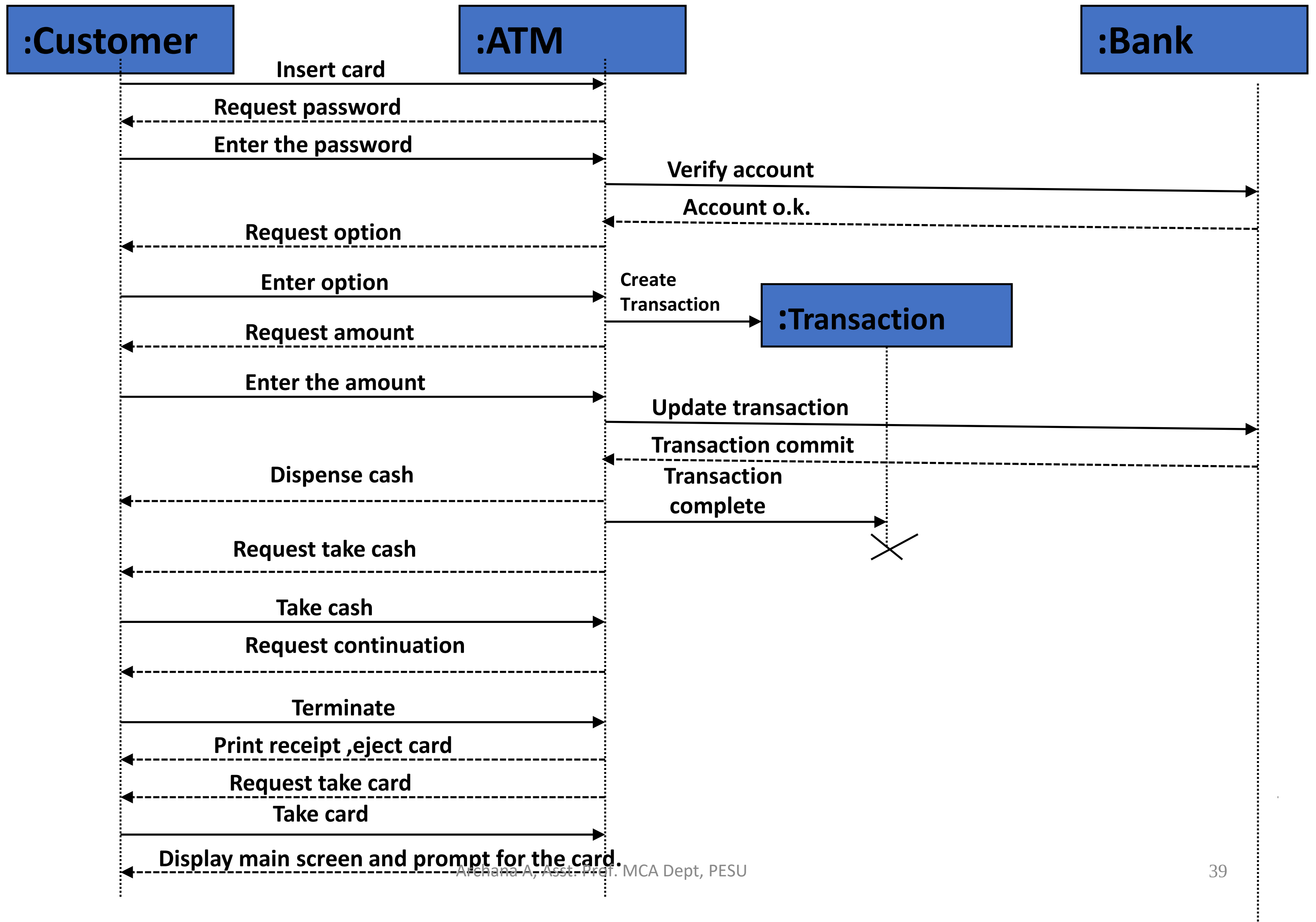


Guidelines for Sequence model

1. Prepare at least one scenario per use case:- the steps in the scenario should be logical commands, not individual button clicks.
2. Abstract the scenario into sequence diagrams:-Sequence diagrams clearly shows the contribution of each actor.
3. Divide complex interactions:-Break large interaction into their constituent tasks and prepare a sequence diagram for each of them.
4. Prepare a sequence diagram for error condition:-shows the system response to error condition.

- Draw a Sequence diagram for withdrawal of cash (normal flow) with Scenario

Sequence diagram [for withdrawal of cash, normal flow]



2. Purchase a smart phone though Online shopping App. The user has to follow a particular sequence to accomplish the task. Write a sequence diagram from the user perspective.

3. Draw a sequence diagrams for Library management system for the following cases:

a) Returning the accessed book
b) Renewal of book
c) Penalty for delay in return.

Activity Modeling

Activity Models

- Activity models are expressed through **Activity diagram**.

- An activity diagram shows **the sequence of steps that make up a complex process, such as algorithm or work flow.**

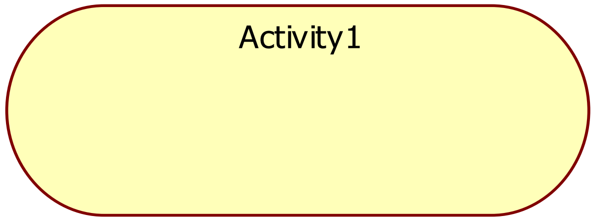
- It shows flow of control similar to sequence diagrams, but **focuses on operations rather than on objects.**

- Activity diagrams are most useful during the early stages of designing algorithms and work flow.

Components used in Activity diagram

- Elongated oval - shows activities and
- arrows - show their sequencing.
- Diamond - shows a decision point and
- heavy bar - shows splitting or merging of concurrent thread.

1. **Activities** – elongated oval

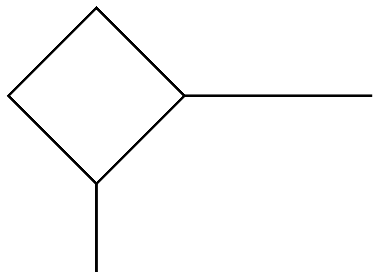


- The steps of an activity diagram are operations, specifically activities from the state model.

- Completion of activity is a completion of event and usually indicate that the next activity can be started.

- Unlabelled arrow $\longrightarrow$ shows first activity must complete before the second activity can began.

2. Branches



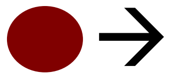
- If there is more than one successor to an activity, each arrow maybe labeled with a condition in square bracket.eg:- [failure] [success].

- All subsequent conditions are tested when an activity completes. If one condition is satisfied, its arrow indicates next activity to perform.

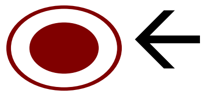
- As a notational convenience, a diamond shows a branch into multiple successors.

3. Initiation and Termination

- A solid circle with an out going arrow shows the starting point of an activity diagram.



- A bulls eye(a solid circle surrounded by a hollow circle) shows the termination point, this symbol has only incoming arrow.

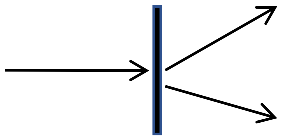


4. Concurrent activities

- Organizations and computer systems can perform more than one activity at a time.

- Eg:- One activity may be followed by another activity(sequential control), then split into several concurrent activities(a fork of control) and finally combined into a single activity(a merge of control).

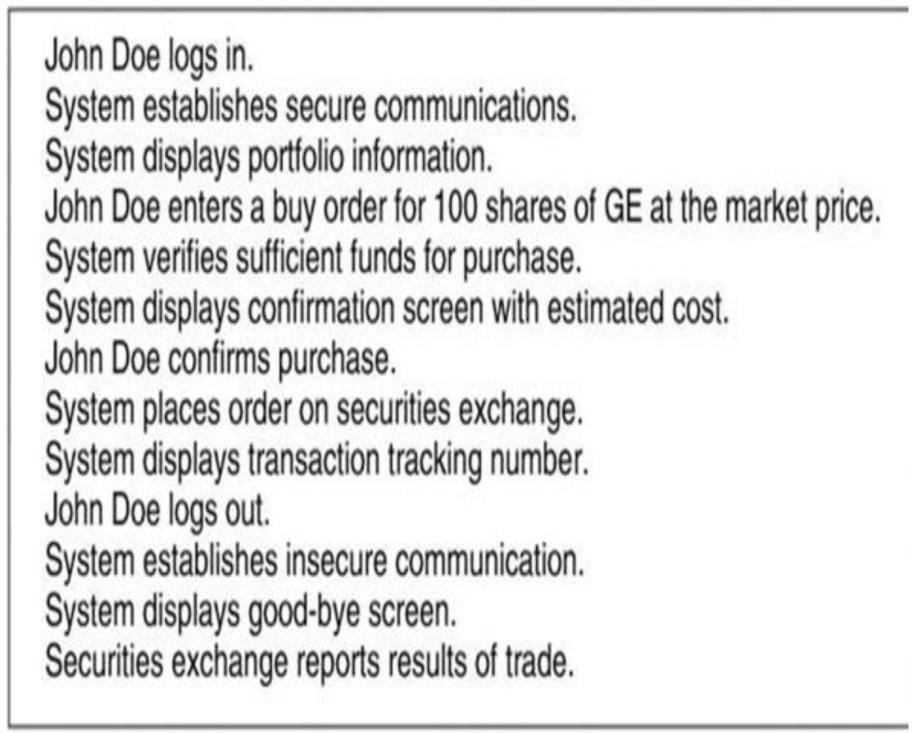
- A fork or merge is shown by a synchronization bar a heavy line with one or more input arrows and one or more output arrows.

- 

- Eg:-   Activity diagram for processing of **stock trade order that has been received** by an online stock broker.

Scenario for a session with online Stock Broker

Go To



John Doe logs in.
System establishes secure communications.
System displays portfolio information.
John Doe enters a buy order for 100 shares of GE at the market price.
System verifies sufficient funds for purchase.
System displays confirmation screen with estimated cost.
John Doe confirms purchase.
System places order on securities exchange.
System displays transaction tracking number.
John Doe logs out.
System establishes insecure communication.
System displays good-bye screen.
Securities exchange reports results of trade.

Eg:-

Go to

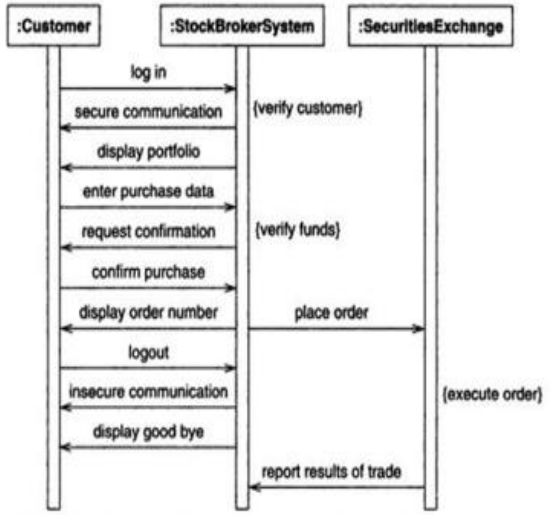


Figure 7.5 Sequence diagram for a session with an online stock broker. A sequence diagram shows the participants in an interaction and the sequence of messages among them.

30

Eg:-

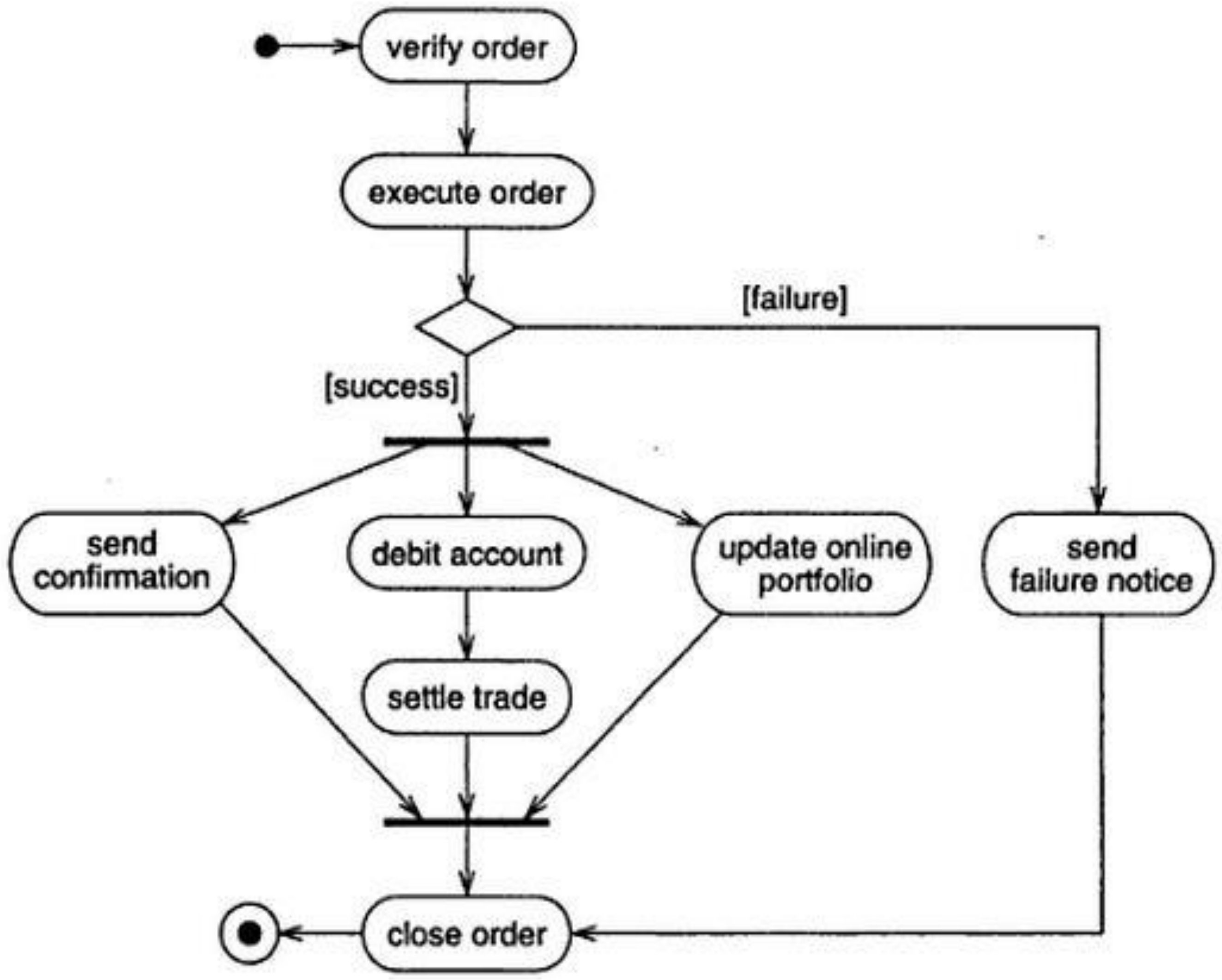


Figure 7.9 Activity diagram for stock trade processing. An activity diagram shows the sequence of steps that make up a complex process.

Guidelines for Activity Models

1. Don’t misuse activity diagram:-It should not be used as an excuse to develop software via flow chart.

2. Level diagrams:-Activities on a diagram should be at a consistent level of detail.

3. Be careful with branches and conditions :-If there are conditions, at least one must be satisfied when an activity completes, conditions using an else condition.

Executable activity diagram

- Activity diagrams are not only useful for designing the steps in a complex process, but they can also be used to show the progression of control during execution.

- An activity token can be placed on activity symbol to indicate that it is executing. When an activity completes, the token is removed and placed on the outgoing arrow. Then token moves to the next activity.

- Multiple tokens can arise through concurrency.

- Draw an Activity diagram for
  1. Account transaction (deposit, withdraw, bal enquiry, Mini statement, pin change, etc) through ATM.

  2. Book issue, return/return with penalty, book renew of Library application.

Advanced Interaction Modeling

Advanced concepts in Use Case model

Use Case Relationships

- **Include Relationship**
  - Incorporate one use case within the behavior sequence of another use case.
- **Extend Relationship**
  - Add incremental behavior to a use case.
- **Generalization**
  - Show specific variations on a general use case.

Use case Relationship…

- Complex use cases can be built from smaller pieces with the include, extend and generalization relationship.

- **Include Relationship:-**

  - The include relationship incorporates one use case within the behavior sequence of another use case.

  - An included use case is like a subroutine.

  - The **UML notation** for an included relationship is **dashed arrow from the source use case(including) to the target use case(included).**

Cont…

- **The keyword <<include>> annotates the arrow**

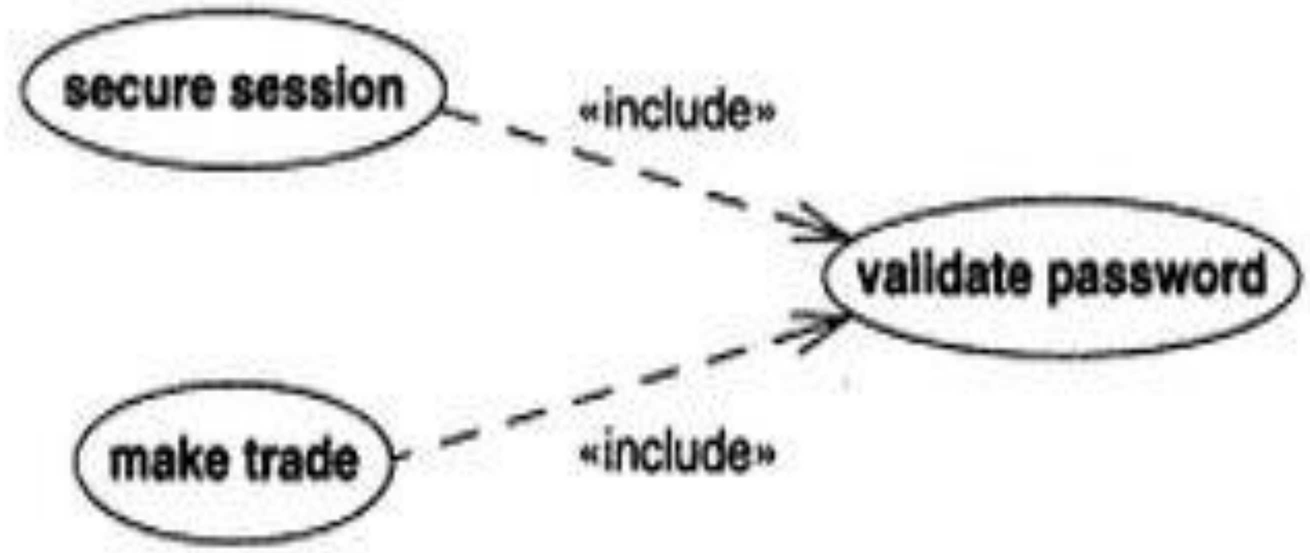


Figure 8.1 Use case inclusion. The *include* relationship lets a base use case incorporate behavior from another use case.

Extended relationship:

- The extended relationship adds incremental behavior to an use case.
- It represents the frequent situation in which some initial capability is defined and later features are added modularity.
- Eg:-A stock brokerage system might have the base use case trade stocks, which permits a customer to purchase stocks for cash on hand in the account.
- The initial behavior is inserted at the point where the purchase cost is checked against the account balance.

Cont…

- The UML notation for extended relationship is a dashed arrow from the extension use case to the base use case.
- The keyword <<extend>> annotates the arrow.

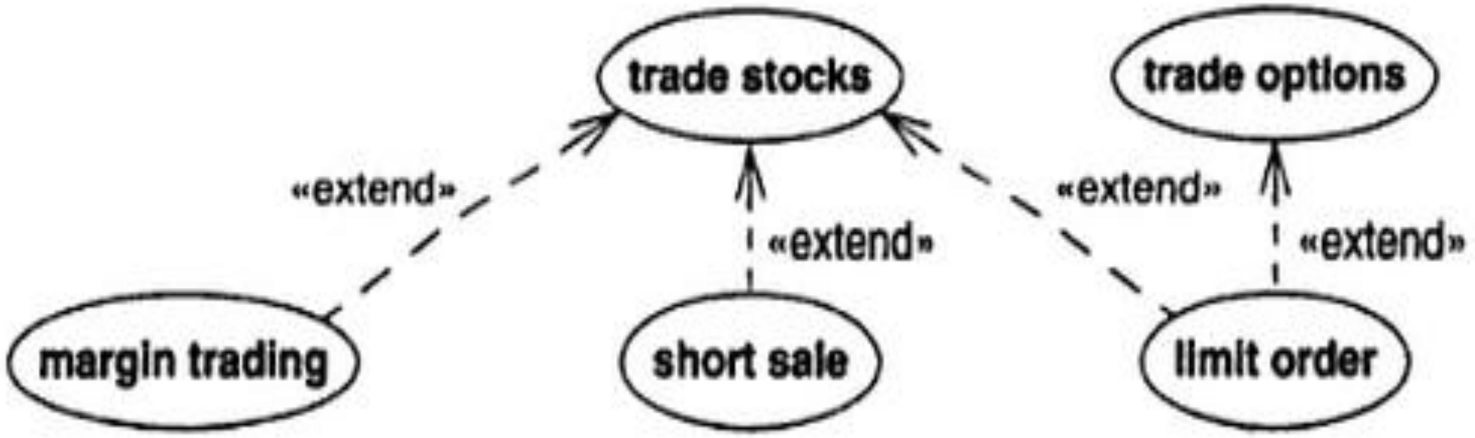


Figure 8.2 Use case extension. The *extend* relationship is like an *include* relationship looked at from the opposite direction. The extension adds itself to the base.

Cont…

- The extend relationship **connects** an extension use case to a base use case.

- The base use case must be a valid use case in the absence of any extensions.

- The behavior sequence of the extension use case occurs at the given point in the sequence.

- The extension behavior occurs only if the condition is true when control reaches the insert location.

Use cases - Generalization

- Generalization can show specific variations on a general use case, analogous to generalizations among classes.

- A parent use case represents a general behavior sequence. Child use case specialize the parent by inserting additional steps or by refining steps.

- The UML indicates generalization by **arrow** with its tail on the child use case and a triangular arrow head on the parent use case.

Eg:-

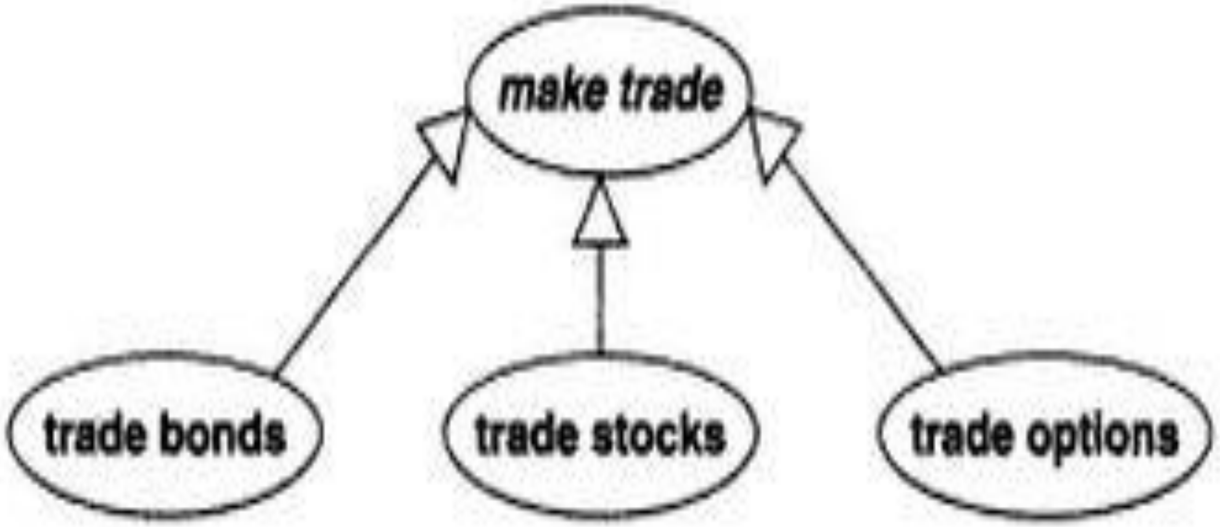


Figure 8.3 **Use case generalization**. A parent use case has common behavior and child use cases add variations, analogous to generalization among classes.

Combinations of use case relationships

- A single diagram may combine several kinds of use case relationships

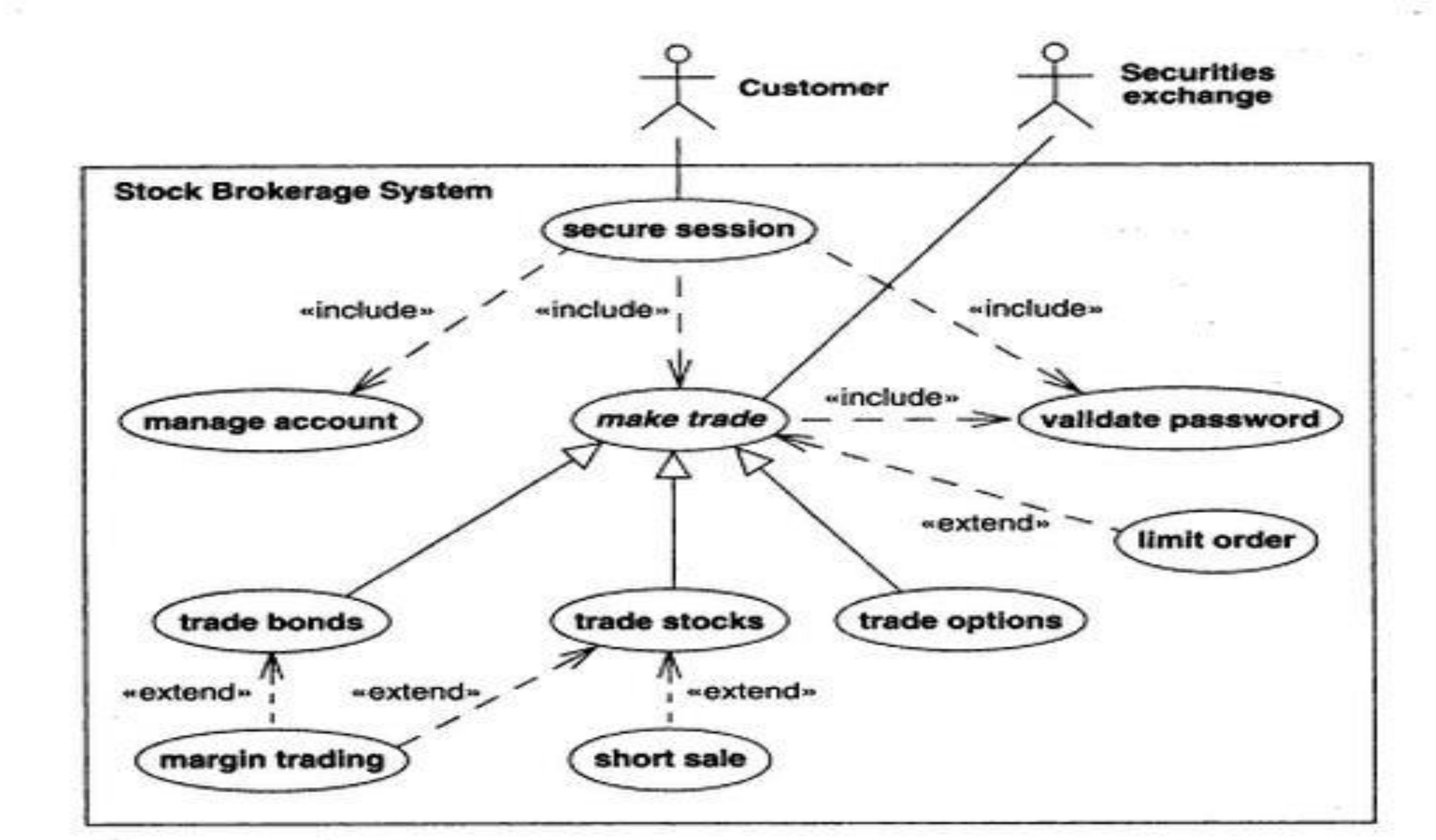


Figure 8.4 Use case relationships. A single use case diagram may combine several kinds of relationships.

Advanced concepts in sequence model

Procedural Sequence Models

- A sequence diagram shows the participants in an interaction and the sequence of messages among them.
- A sequence diagram shows the interaction of a system with its actors to perform all or part of a use case.
- Each use case requires one or more sequence diagrams to describe its behavior.

Cont…

- Sequence diagrams contains independent object.
- The UML has elaborations for sequence diagrams to show procedural calls

1. Sequence diagram with Passive objects.

2. Sequence diagram with Transient objects.

1. Sequence diagram with Passive objects

- With procedural code all objects are not constantly active.
- Most objects are passive and do not have their own threads of control.
- A passive object is not activated until it has been called.
- Once the execution of an operation completes the control returns to the caller, the passive objects become inactive.

Example…

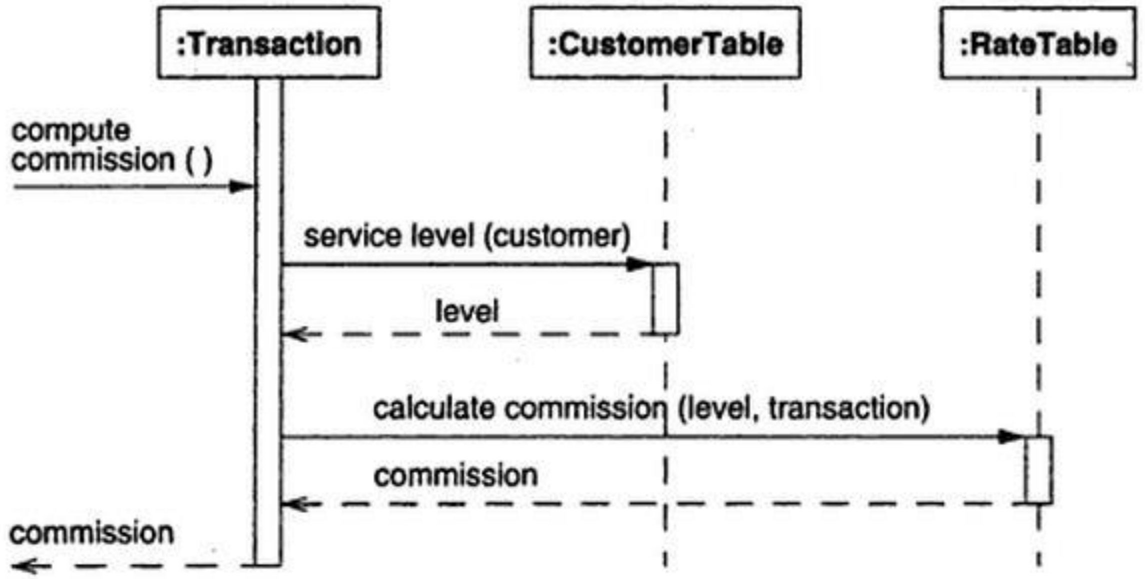


Figure 8.5 Sequence diagram with passive objects. Sequence diagrams can show the implementation of operations.

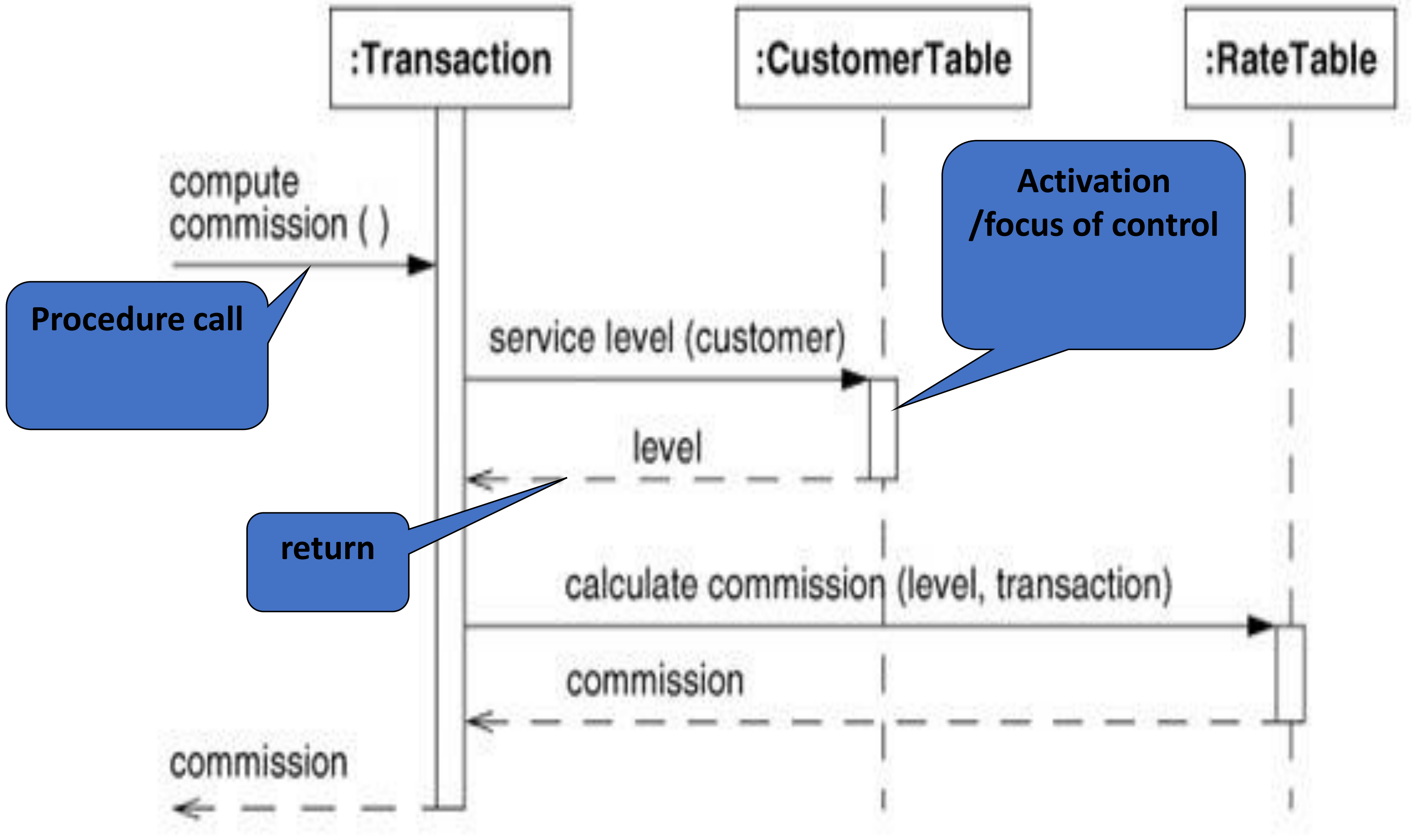


Figure 8.5 Sequence diagram with passive objects. Sequence diagrams can show the implementation of operations.

Cont…

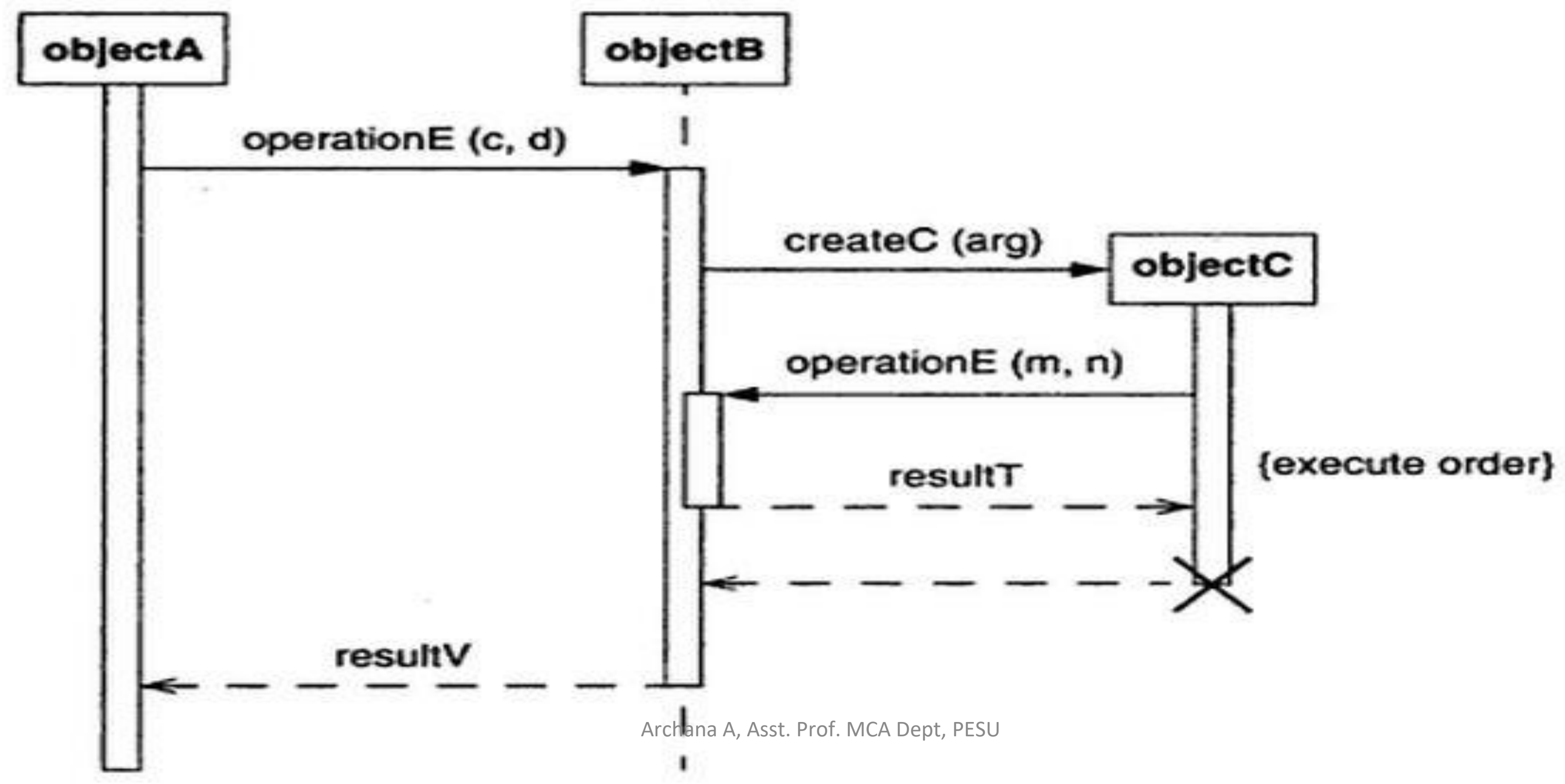
- The example calculates commission for stock brokerage transaction.
- The transaction objects receives a request to compute commission.
- It obtains customer's service level from customer table then asks the rate table to compute commission based on the service level, after which it returns commission value to the caller.

Cont…

- In the given example
- Thin rectangle shows period of time for an object execution.
- This is called activation or focus of control.
- Dashed line shows period of time when an object exists but not active.
- The entire period during which object exists is called life line.

2. Sequence diagram with Transient objects

- Consider an example, many applications have a mix of active and passive objects. They create and destroy objects.



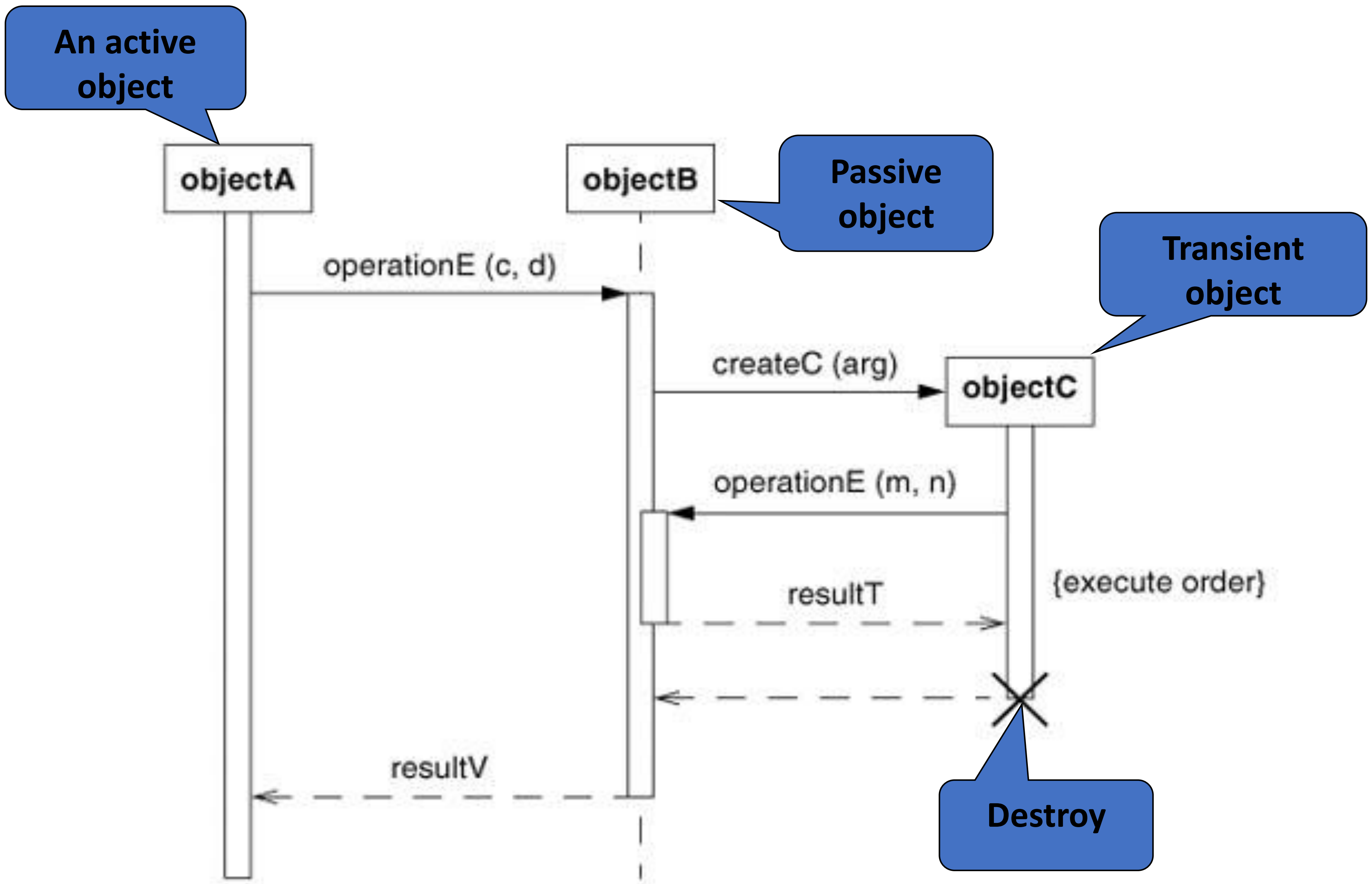


Figure 8.6 Sequence diagram with a transient object. Many applications have a mix of active and passive objects. They create and destroy objects.

Cont…

- Here objectA is an active object which initiates an operation, because it is active, its activation rectangle spans the entire time shown in diagram.
- Object B is a passive object that exists during the entire time shown in figure, but it is not active for the whole time.
- The UML shows its existence by dashed line Object B’s life line broadens into an activation rectangle when it is processing a call.

Cont…

- For some part of time, it performs a recursive operation shown by the doubled activation rectangle between the call by object C on operation E and returns the result values.

- Object C is created and destroyed during the time shown on diagram so its lifeline does not span the whole diagram.

Sequence Diagrams with Transient Objects

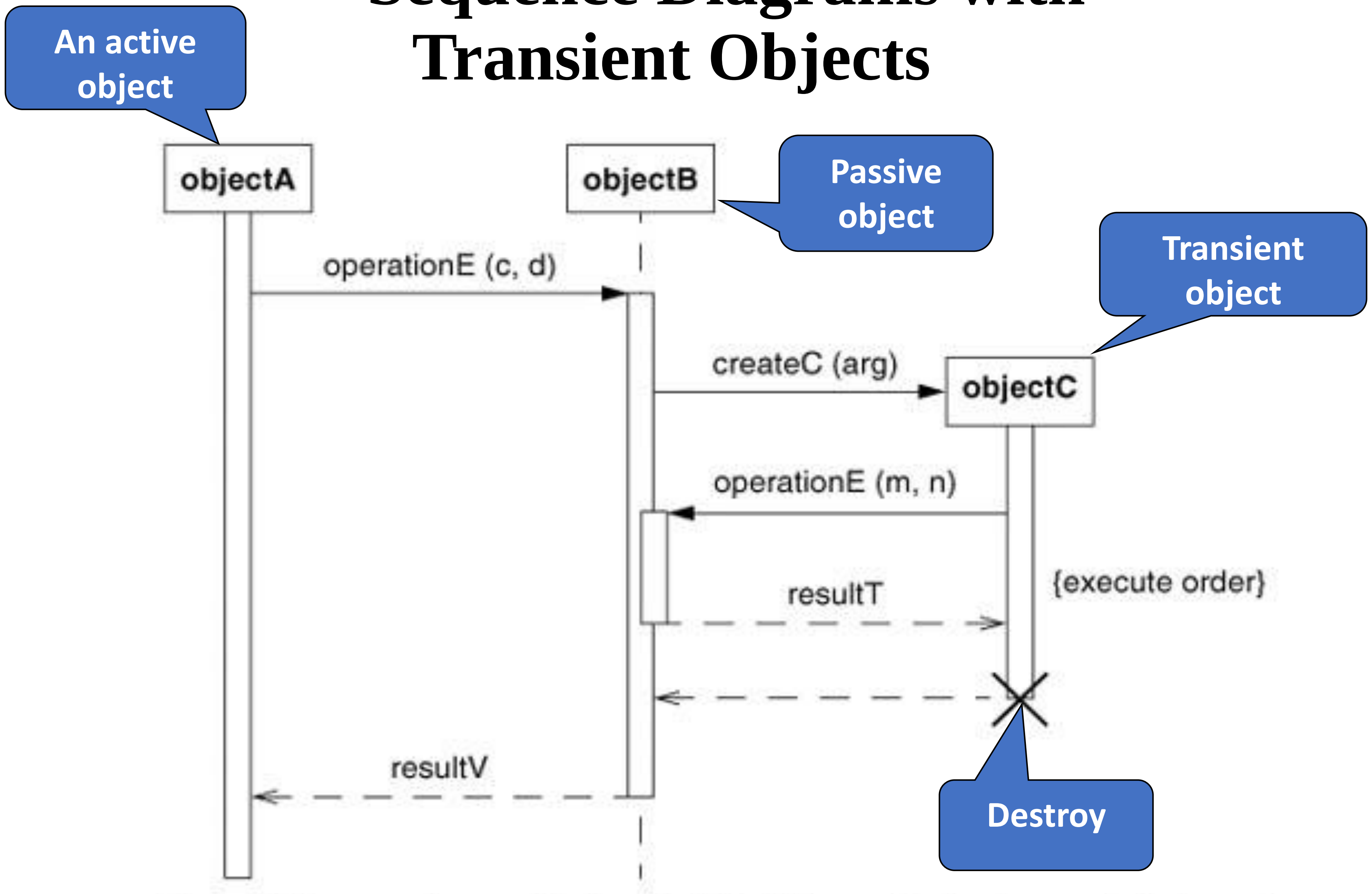


Figure 8.6 Sequence diagram with a transient object. Many applications have a mix of active and passive objects. They create and destroy objects.

Sequence Diagram For a Session

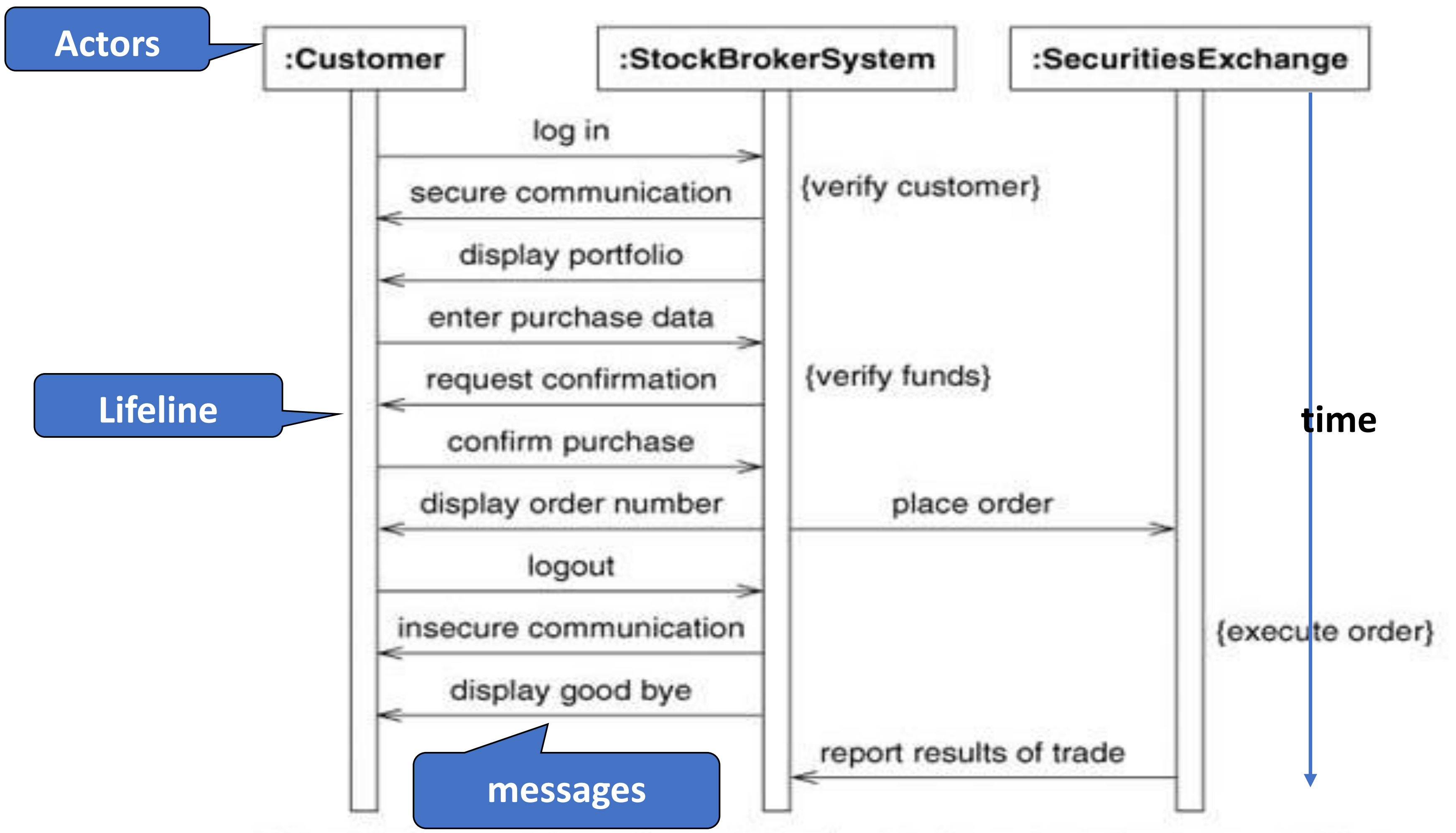


Figure 7.5 Sequence diagram for a session with an online stock broker. A sequence diagram shows the participants in an interaction and the sequence of messages among them.

- For large scale interactions containing many independent tasks, we can draw separate sequence diagram for each task.

- Eg:-sequence diagram for a stock purchase

  - Sequence diagram for a stock quote

  - Sequence diagram for a stock purchased **that fails**.

A stock purchase

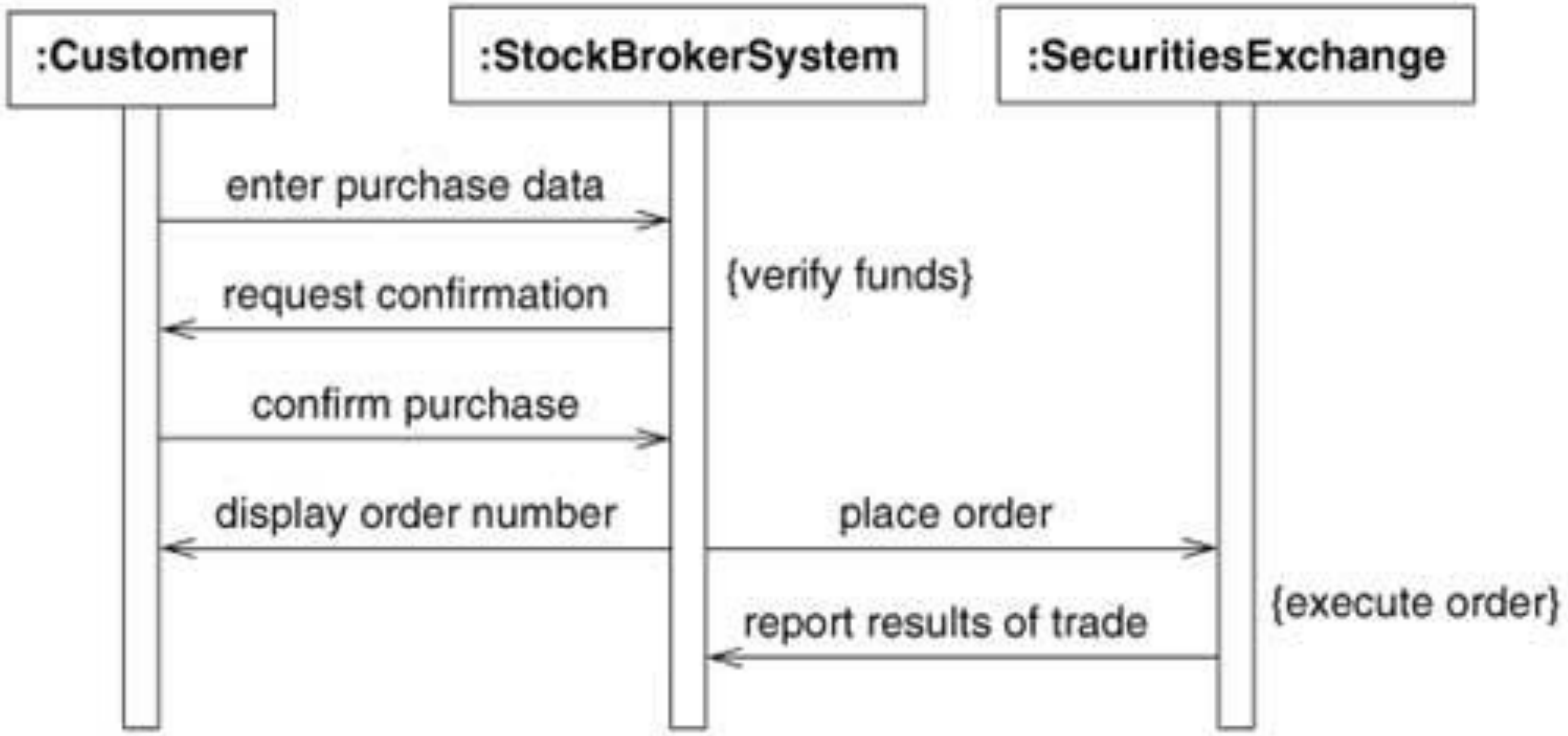


Figure 7.6 Sequence diagram for a stock purchase. Sequence diagrams can show large-scale interactions as well as smaller, constituent tasks.

A stock quote

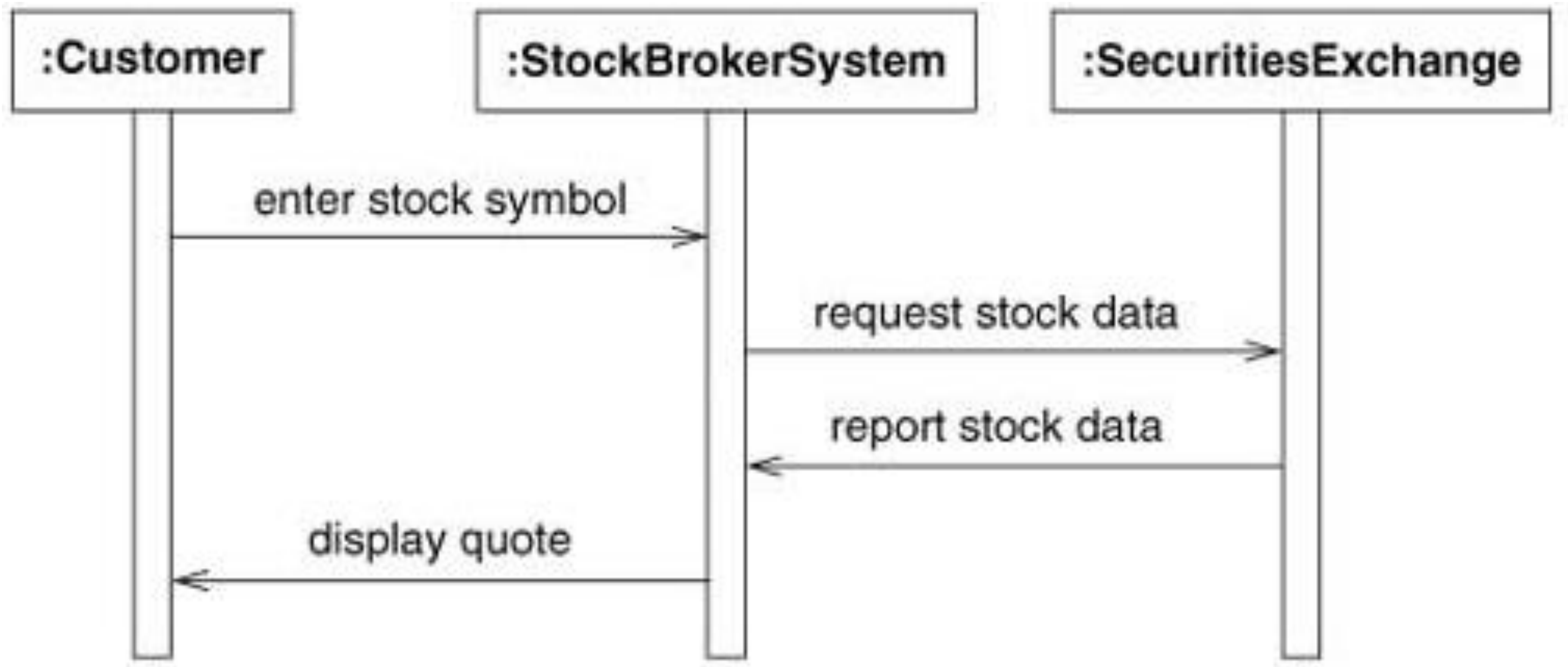


Figure 7.7 Sequence diagram for a stock quote.

A exception case

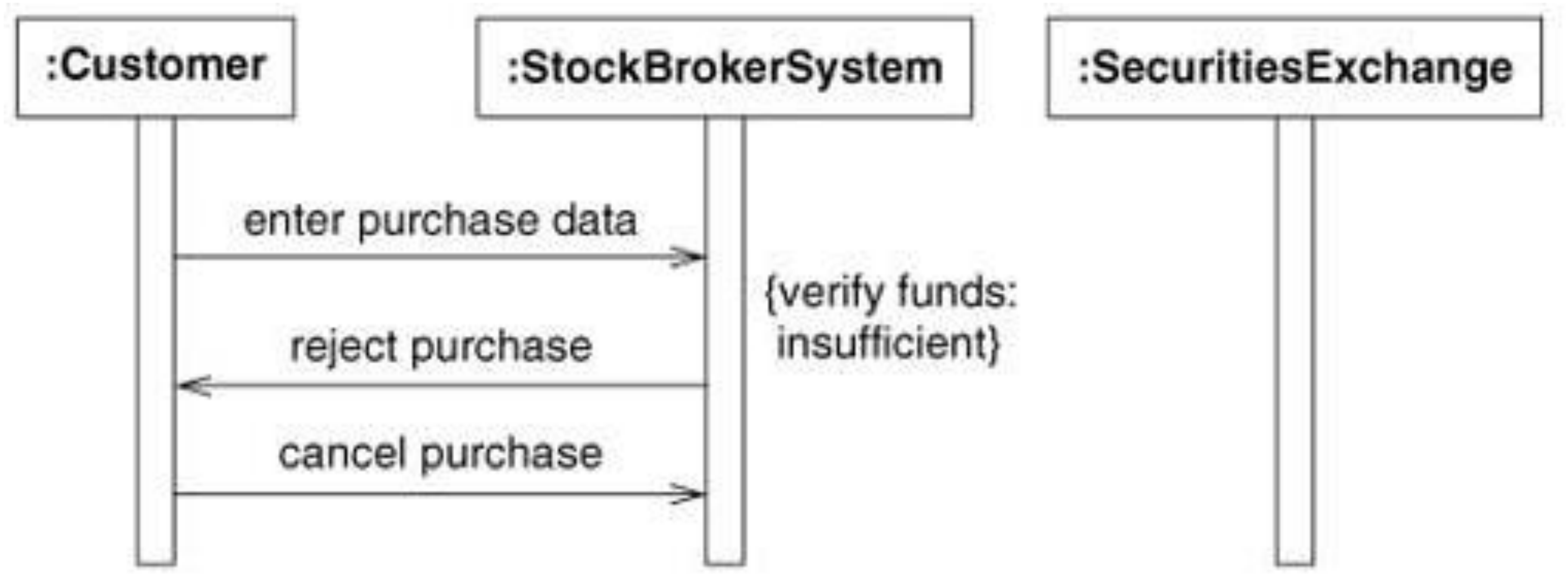


Figure 7.8 Sequence diagram for a stock purchase that fails.

Advanced concepts in activity model

Special constructs for Activity Models

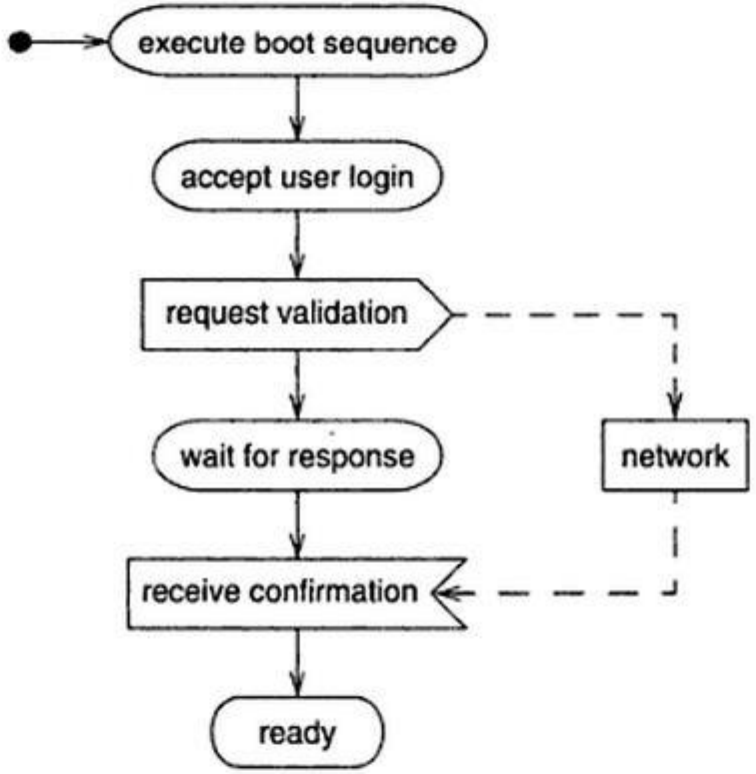
- Activity diagram have additional notation that is useful for large and complex applications.

1. Sending and receiving signals:

- Consider a work station that is turned on. It goes through a boot sequence and then requests the user login. After entry of name and password, the workstation queries the network to validate the user.

- Upon validation, the work station then finishes the startup process.

Eg…


execute boot sequence
accept user login
request validation
wait for response
network
receive confirmation
ready

Sending and Receiving Signals

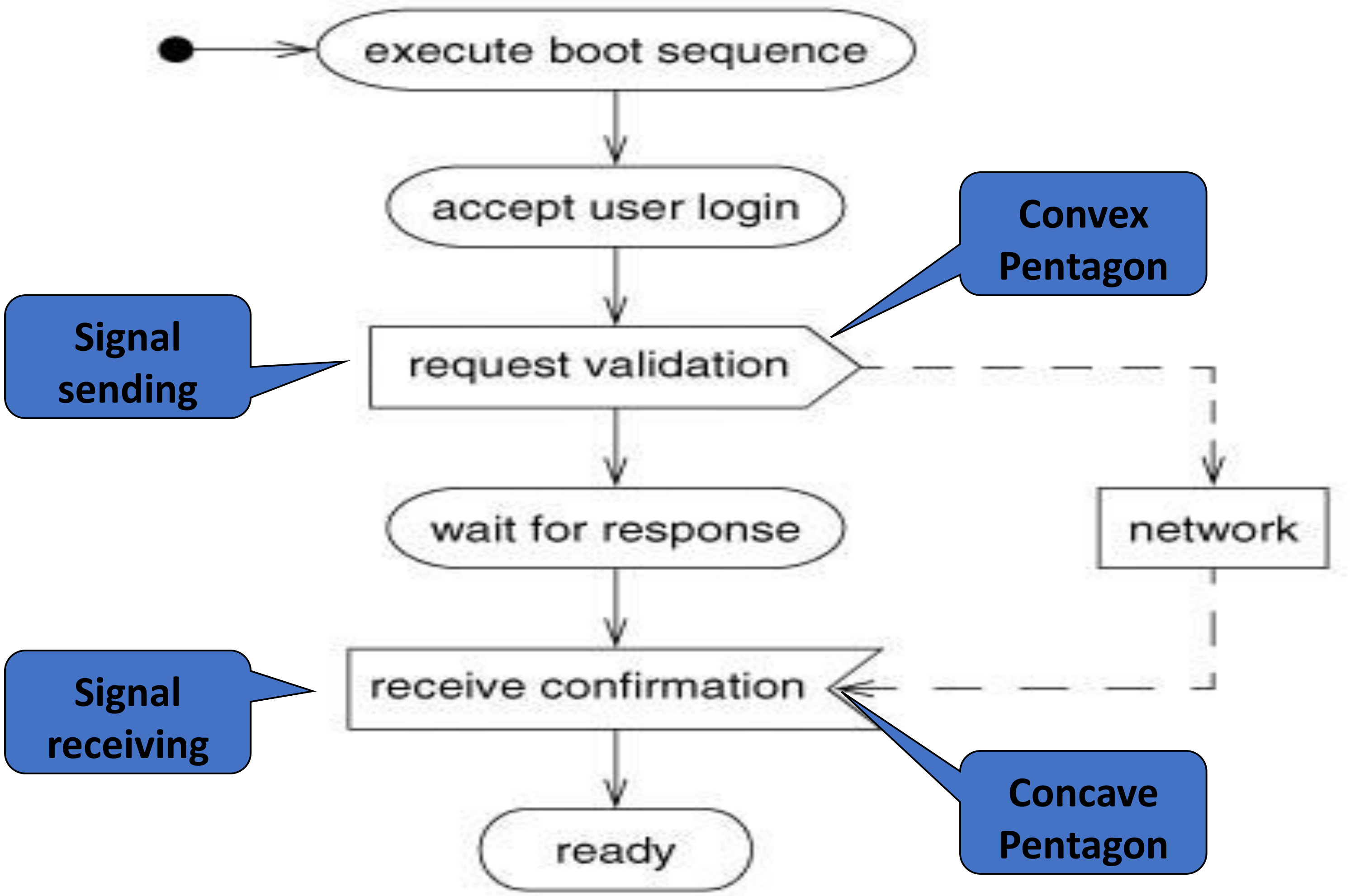


Figure 8.7 Activity diagram with signals. Activity diagrams can show fine control via sending and receiving events.

Cont…

Here UML shows,

- sending of signal as a convex pentagon.

- Receiving of signal as concave pentagon

2. Swim lanes

- In a business model, it is often useful to know which human organization is responsible for an activity.
  - Eg:-sales, marketing, purchasing.

- When the design of the system is complete, the activity will be assigned to a person.

- But at a higher level it is sufficient to partition the activities among organizations.

- You can show such a partition with an activity diagram by dividing it into columns and lines.

- Each column is called swim lane (by analogy to a swimming pool).

Cont…

- Lines across Swim lane boundaries indicate interaction among different organizations.

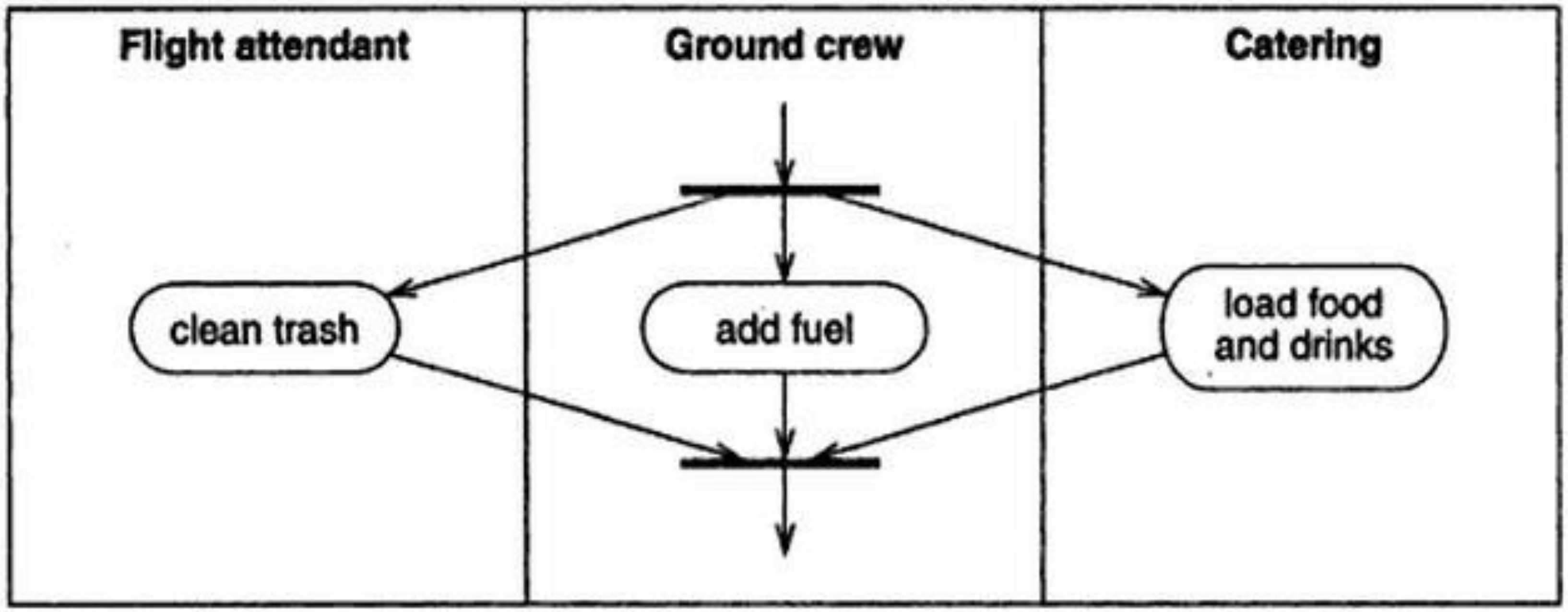


Figure 8.8 Activity diagram with swimlanes. Swimlanes can show organizational responsibility for activities.

Cont…

In the given example

- the flight attendants must clean the trash,
- the ground crew must add fuel and
- catering must load food and drink before a plane is serviced and ready for its next flight.

3.Object Flows

- **Activity diagram with object flow :** an activity diagram can show the object that are input or output of activities.
- An input or output arrows implies a control flow.
- It is unnecessary to draw control flow arrow where there is an object flow.
- Frequently the same object goes through several states during the execution of an activity diagram.

Cont…

- The same object may be an input or an output from several activities, but on closer examination an activity produces or uses an object in a particular state.

- Eg:-an airplane goes through several states as it leaves the gate , flies and then lands again.

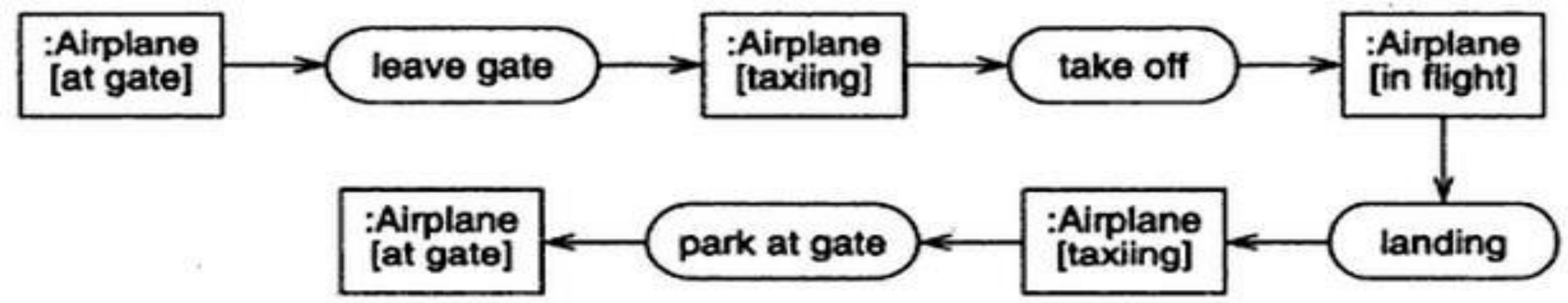


Figure 8.9 Activity diagram with object flows. An activity diagram can show the objects that are inputs or outputs of activities.

Cont…

- Activity diagram shows object flows among different object states.
- UML shows an object value in particular state by placing the state name in square brackets following the object name.

Object Flows

- Show both the control and the progression of an object from state to state as activities act on it.

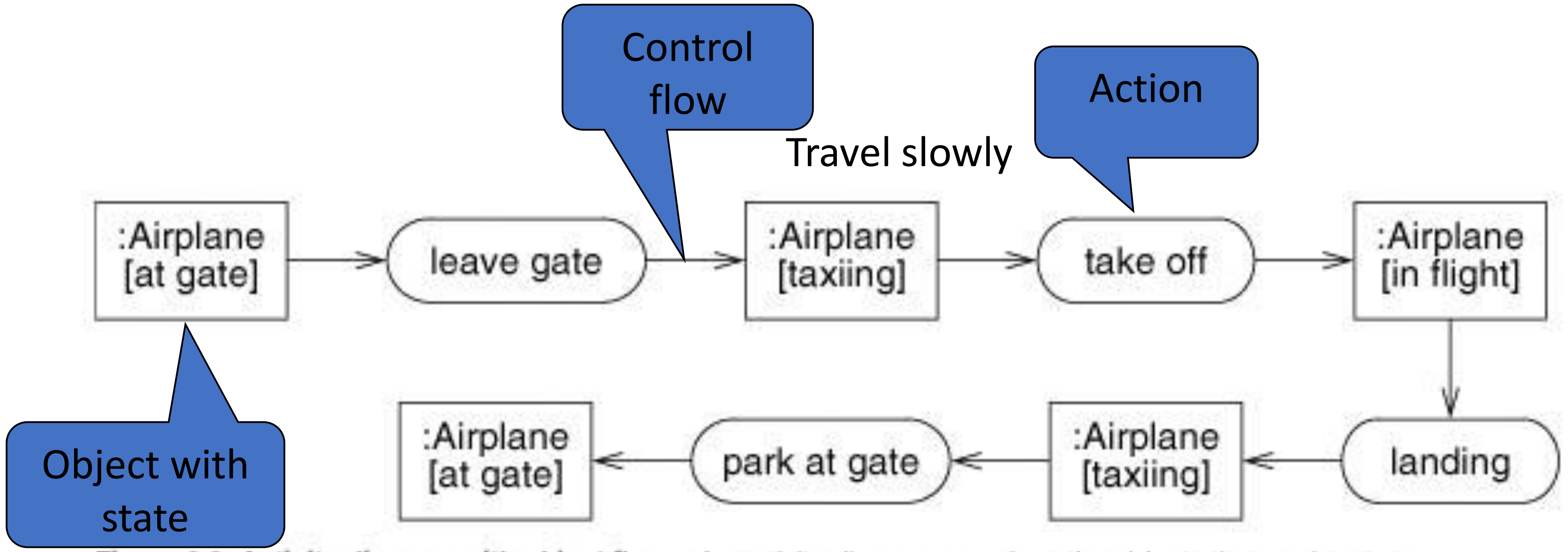


Figure 8.9 Activity diagram with object flows. An activity diagram can show the objects that are inputs or outputs of activities.

Activity diagram for stock trade processing

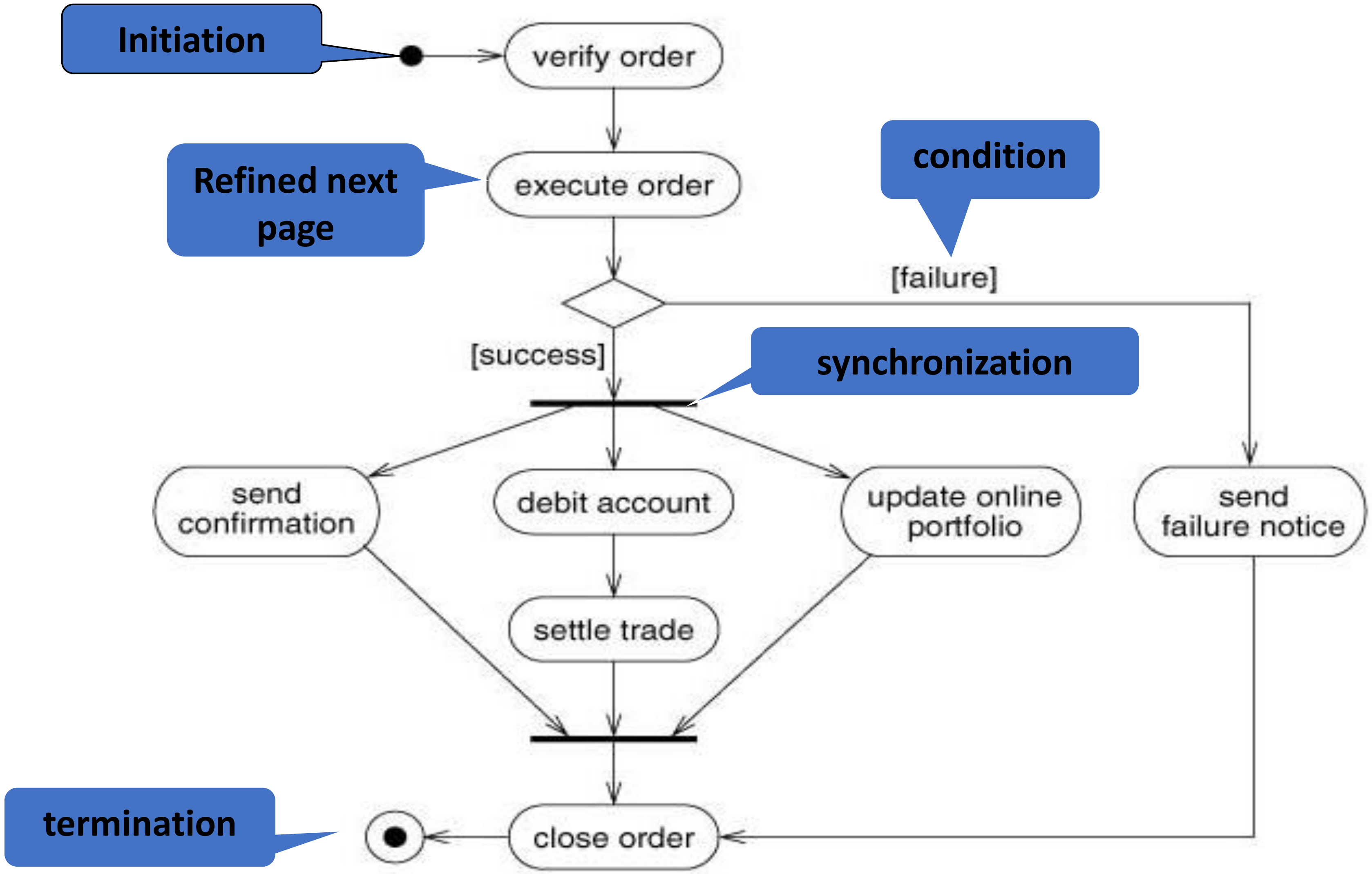


Figure 7.9 Activity diagram for stock trade processing. An activity diagram shows the sequence of steps that make up a complex process.

A Finer Activity for *execute order*

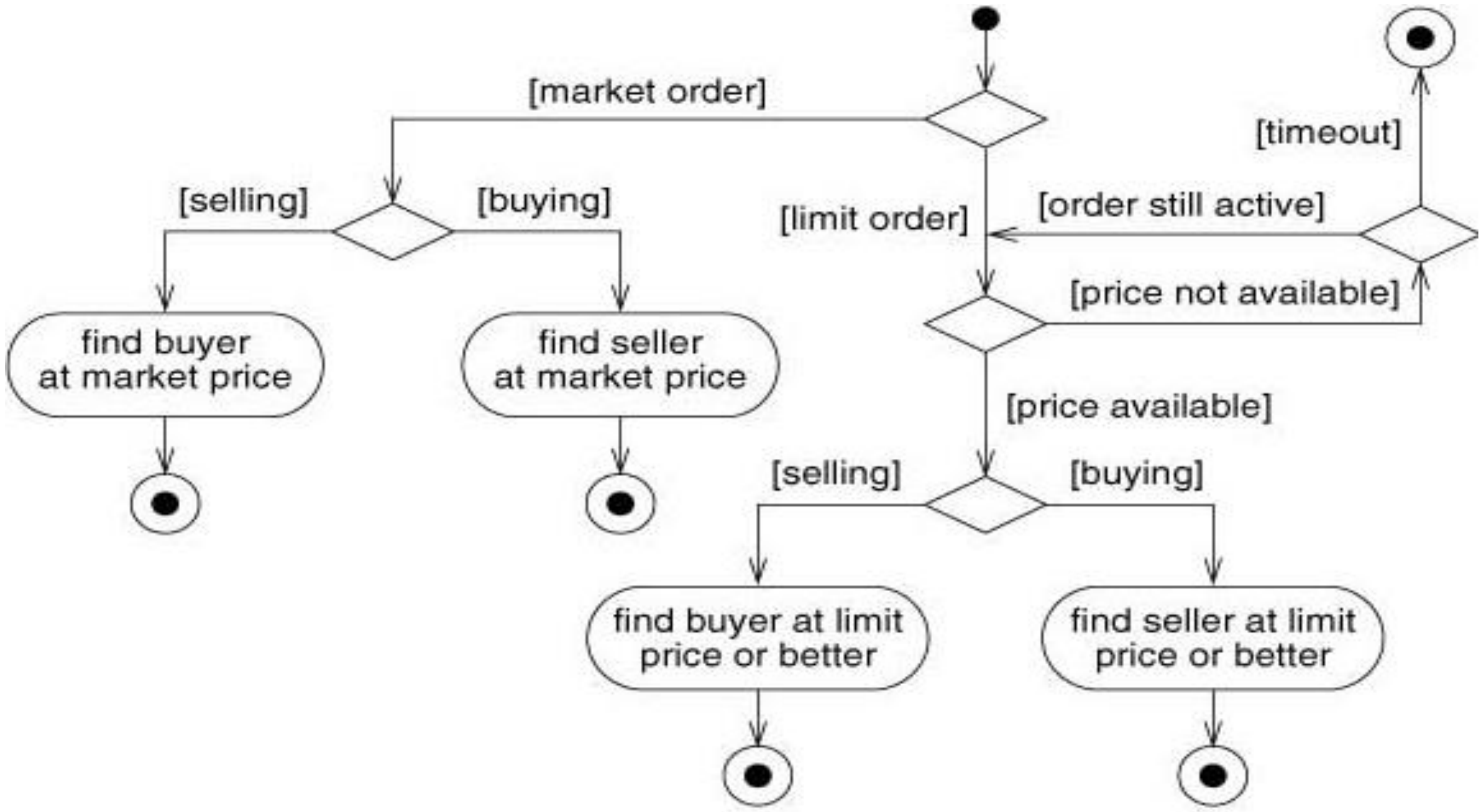


Figure 7.10 Activity diagram for *execute order*. An activity may be decomposed into finer activities.

1. Draw Use case and Sequence diagram for Online bus reservation system (consider all possibilities including error condition).

Sample Questions

1. What is interaction modeling?
2. What is use case model? Explain with examples.
3. What are the guidelines for use case models?
4. Explain sequence model with example.
5. What are the guidelines for sequence models?
6. Explain activity model with example.
7. What are the guidelines for activity model?
8. Draw use case diagram for telephone operation system.
9. Draw sequence and activity diagram for automatic vending machine.