Session 7 – More on UML and CASE

Dao Tran Hoang Chau

Modified from Introduction to Object-Oriented Analysis and Design with UML and Unified Process, Stephen R. Schach

Chapter Overview

- UML Is *Not* a Methodology
- Class Diagrams
- Notes
- Use-Case Diagrams
- Stereotypes
- Interaction Diagrams
- Statecharts
- Activity Diagrams
- Packages
- Component Diagrams

Chapter Overview (contd)

- Deployment Diagrams
- Review of UML Diagrams
- UML and Iteration

The Current Version of UML

- Version
- OMG

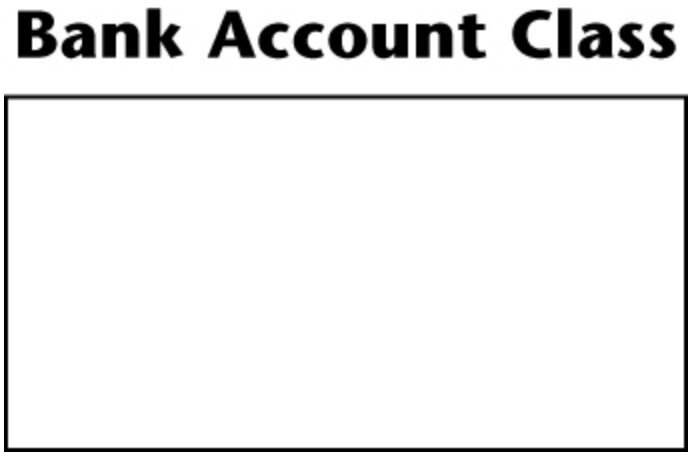
What you learn about UML in this course

- Focus on main workflow
- Focus on IS

Class Diagrams

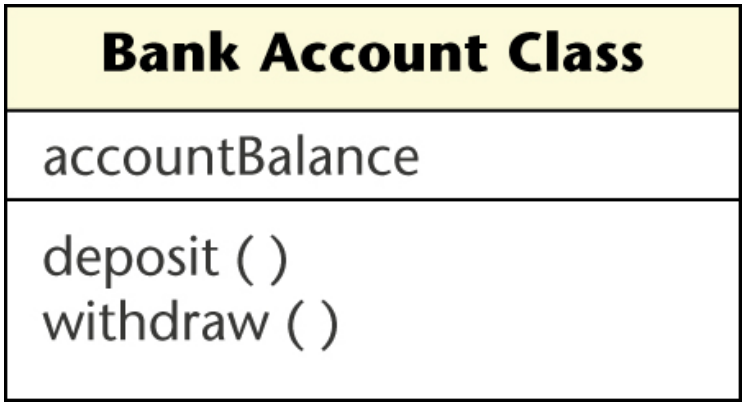
- A class diagram depicts classes and their interrelationships

- Here is the simplest possible class diagram



Class Diagrams (contd)

- Class diagram showing more details of Bank Account Class



- Add as many (or as few) details as appropriate for the current iteration and incrementation

Class Diagrams: Notation (contd)

- Freedom of notation extends to objects
- Example:
  - bank account : **Bank Account Class**
- Bank account is an object, an instance of a class **Bank Account Class**
  - The underlining denotes an object
  - The colon denotes “an instance of”
  - The bold face and initial upper case letters in **Bank Account Class** denote that this is a class

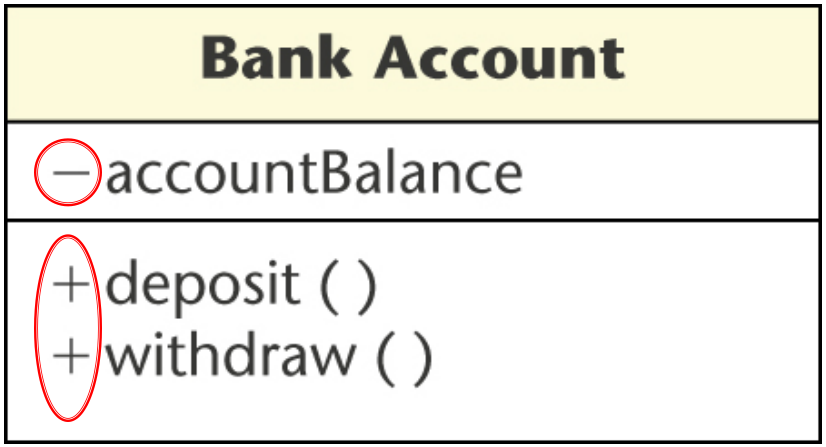
Class Diagrams: Notation (contd)

- UML allows a shorter notation when there is no ambiguity
  - bank account

Class Diagrams: Visibility Prefixes (contd)

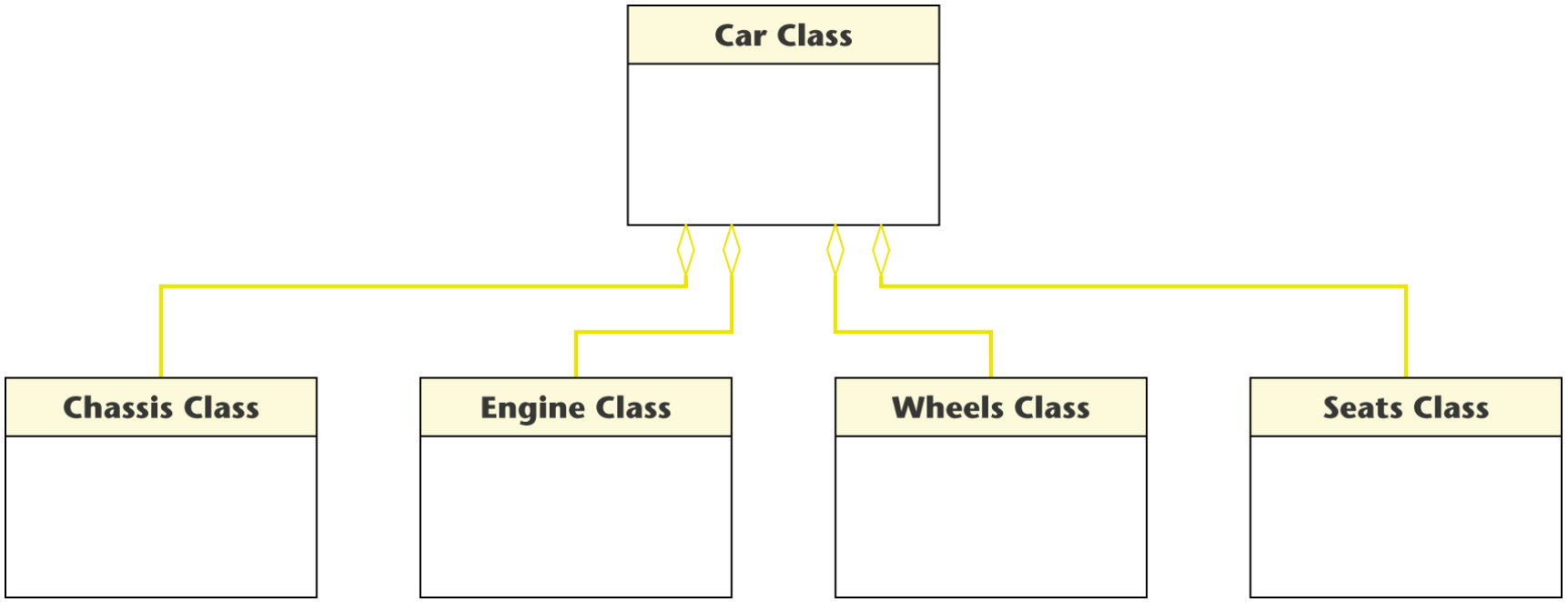
- UML visibility prefixes (used for information hiding)
  - Prefix + indicates that an attribute or operation is public
    - Visible everywhere
  - Prefix – denotes that the attribute or operation is private
    - Visible only in the class in which it is defined
  - Prefix # denotes that the attribute or operation is protected
    - Visible either within the class in which it is defined or within subclasses of that class

Question ?



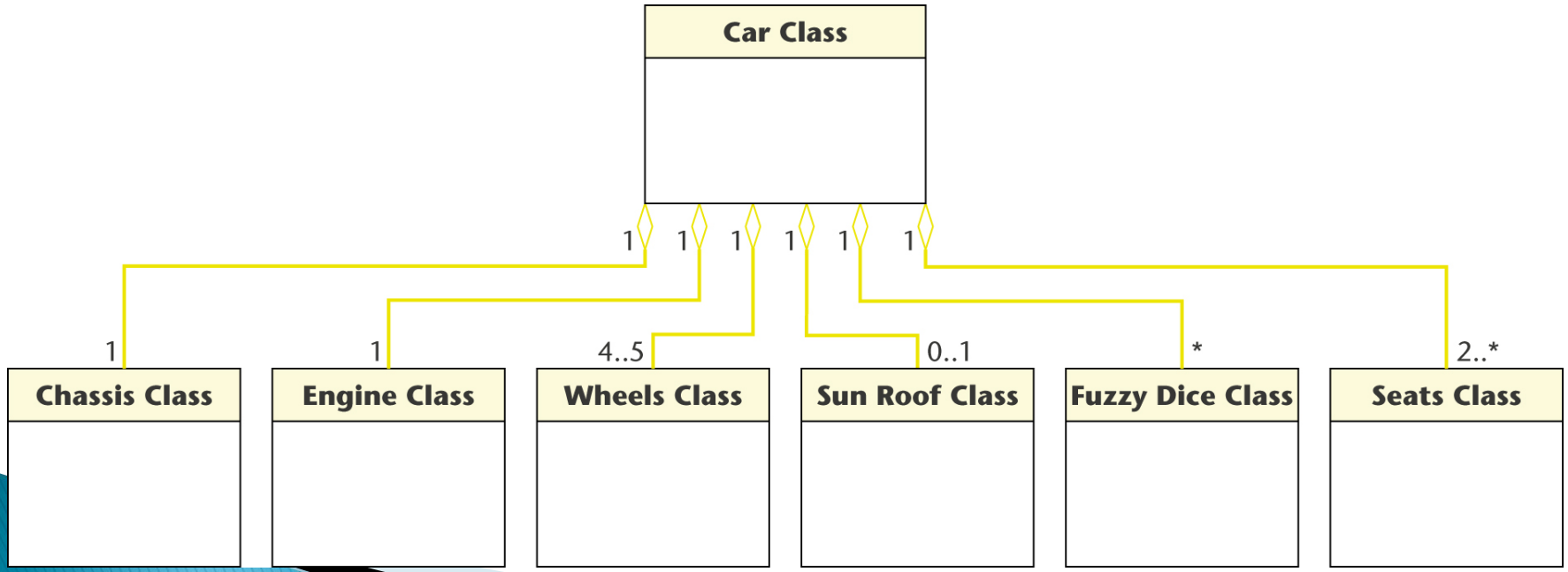
Aggregation

- Example: “A car consists of a chassis, an engine, wheels, and seats”



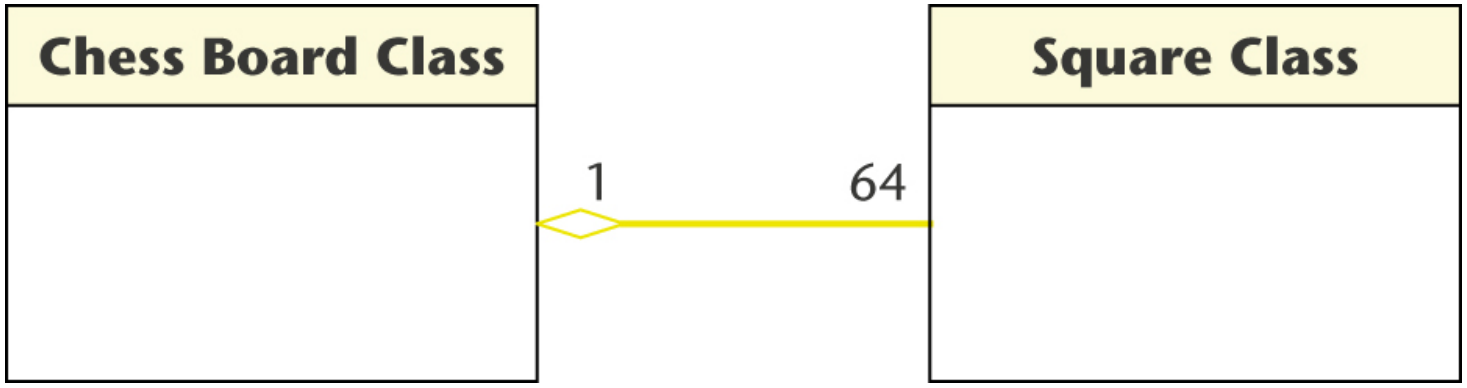
Multiplicity

- Example: “A car consists of one chassis, one engine, 4 or 5 wheels, an optional sun roof, zero or more fuzzy dice hanging from the rear-view mirror, and 2 or more seats”



Composition

- chess board consists of 64 squares



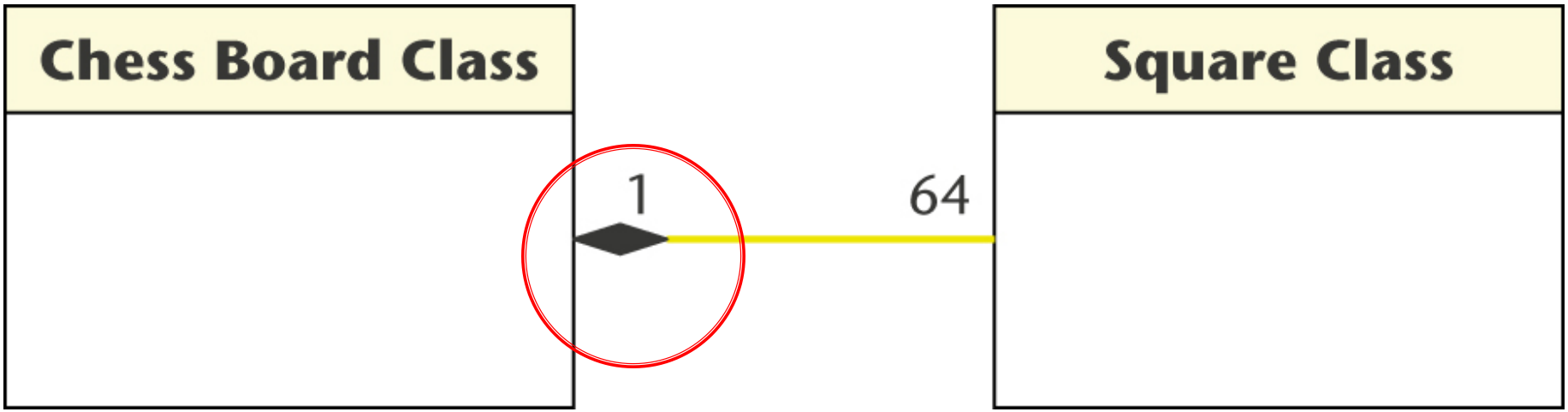
  - a stronger form of aggregation

Composition (contd)

- Aggregation
  - Models the part-whole relationship
- Composition
  - Also models the part-whole relationship
  - In addition, every part may belong to only one whole
  - If the whole is deleted, so are the parts

Composition – notation

- Composition is depicted by a solid diamond

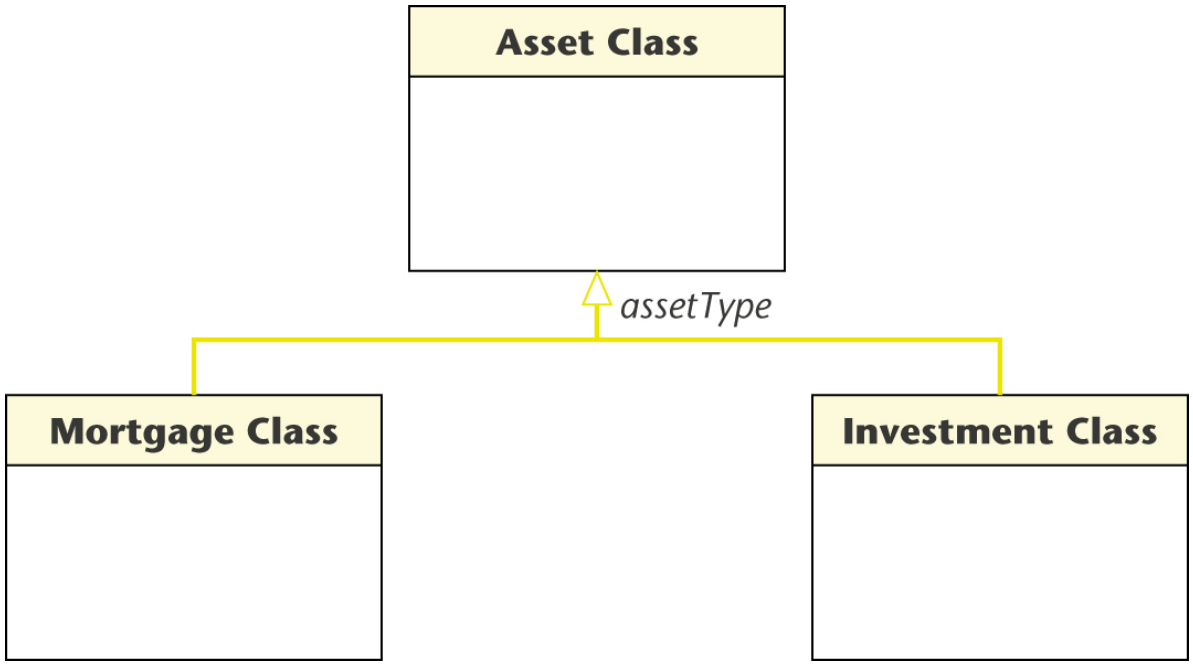


Inheritance

- Inheritance is a required feature of object orientation
- Inheritance is a special case of generalization

Generalization

- Every instance of **Asset Class** or its subclasses has an attribute assetType (the discriminator)



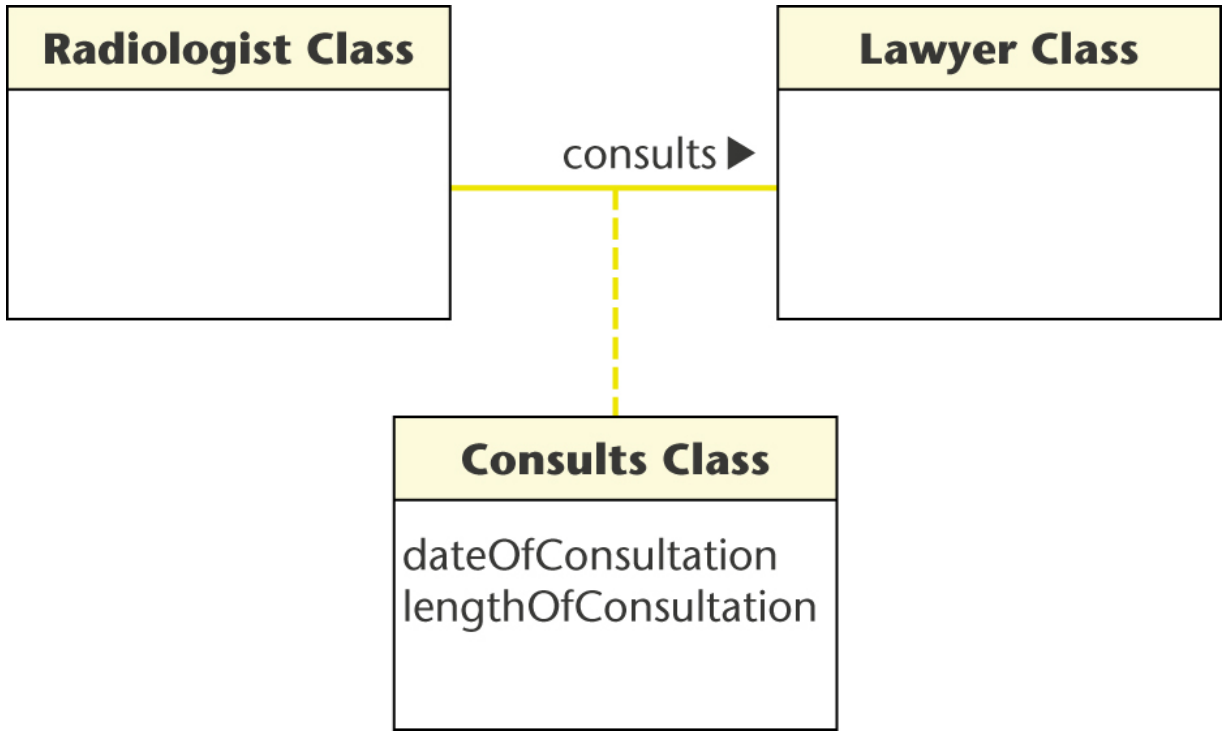
Association

- Example of association:

Association (contd)

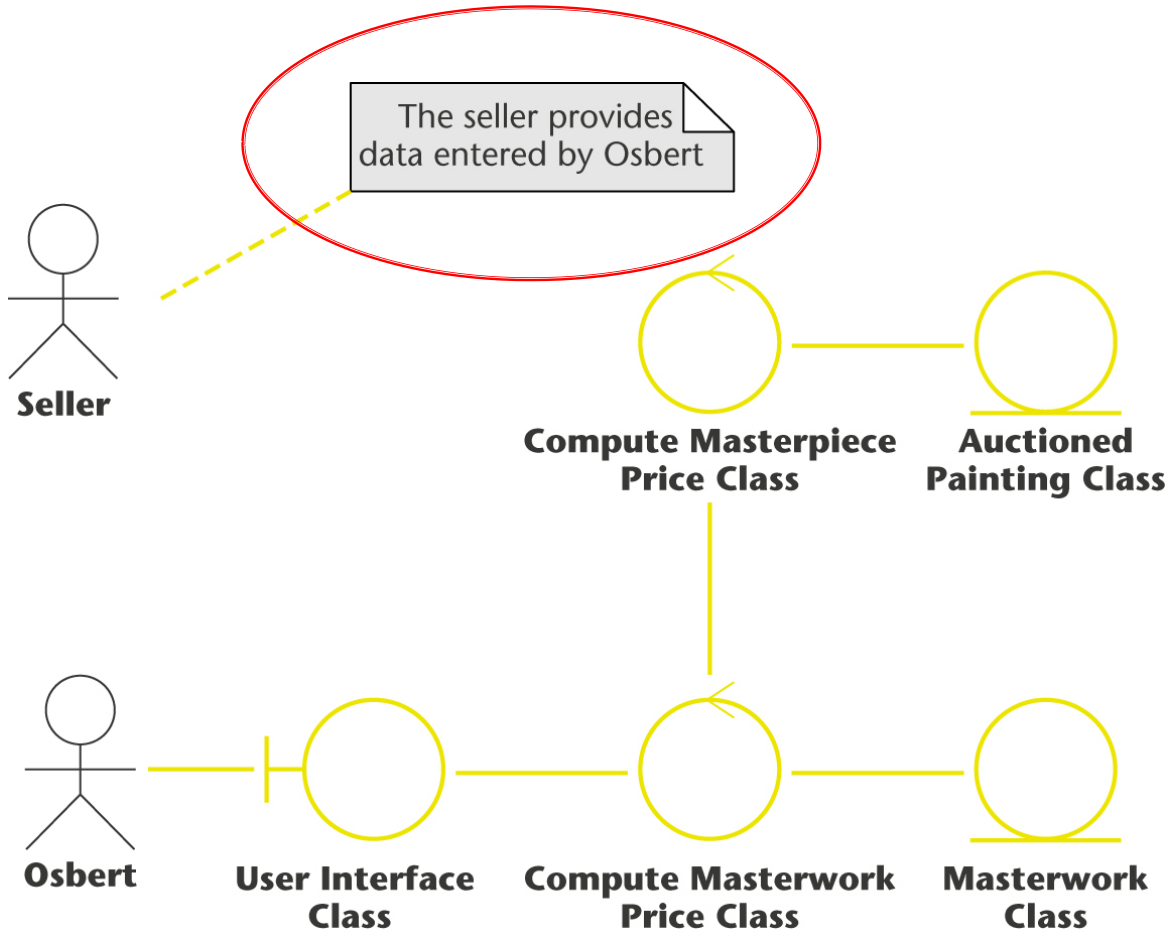
- The association between the two classes may be modeled as a class
  - Example: Suppose the radiologist consults the lawyer on a number of occasions, each one for a different length of time

Association class



Notes

- A comment in a UML diagram is called a note
  - A rectangle with the top right-hand corner bent over
  - A dashed line is drawn from the note to the item to which the note refers

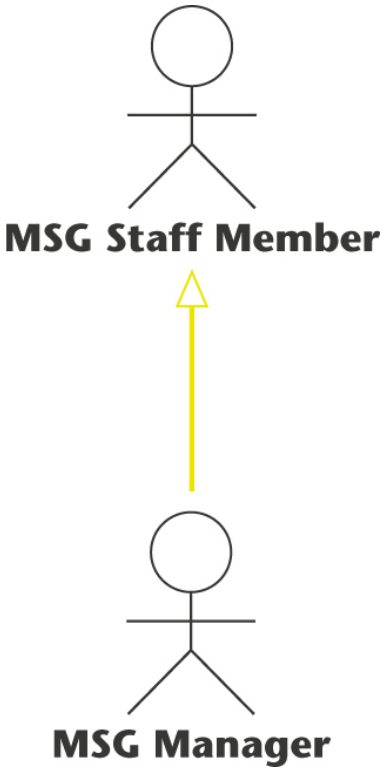


Use-Case Diagrams

- A use case is a model of the interaction between
  - External users of an information system (actors) and
  - The information system itself
    - More precisely, an actor is a user playing a specific role
- A use-case diagram is a set of use cases

Use-Case Diagrams (contd)

- Generalization of actors is supported
  - The open triangle points toward the more general case

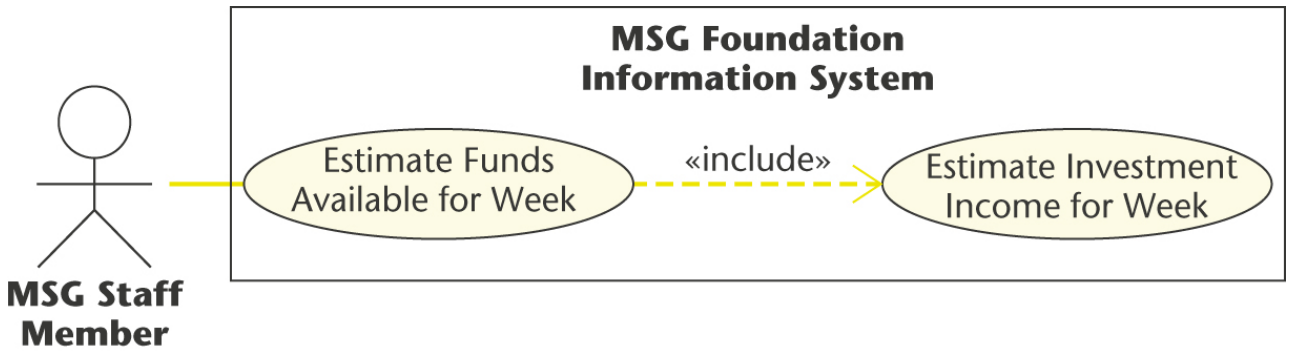


Stereotypes

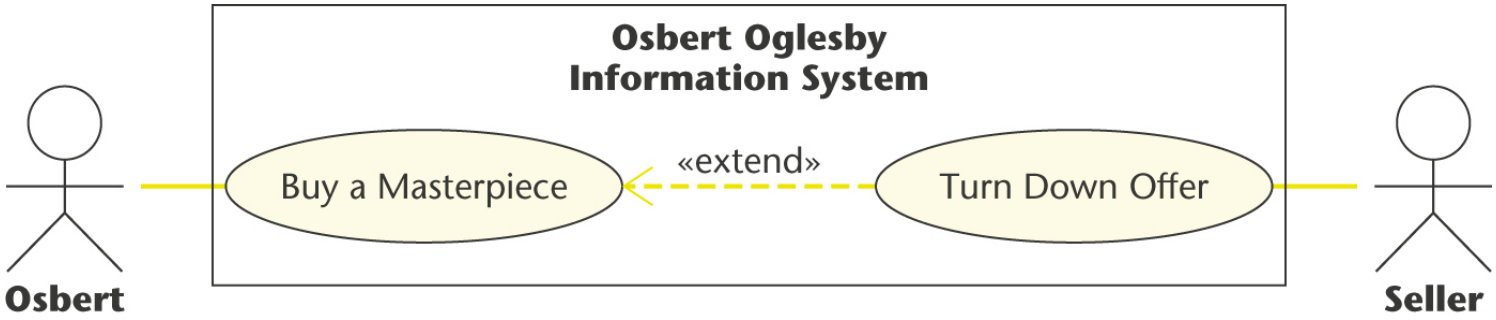
- A stereotype in UML is a way of extending UML
- Stereotypes already encountered include
  - Boundary, control, and entity classes, and
  - The «include» stereotype
- The names of stereotypes appear between guillemets
  - Example: «This is my own construct»

Stereotypes (contd)

- Example: Use case Estimate Funds Available for Week incorporates use case Estimate Investment Income for Week



Stereotypes (contd)



Interaction Diagrams

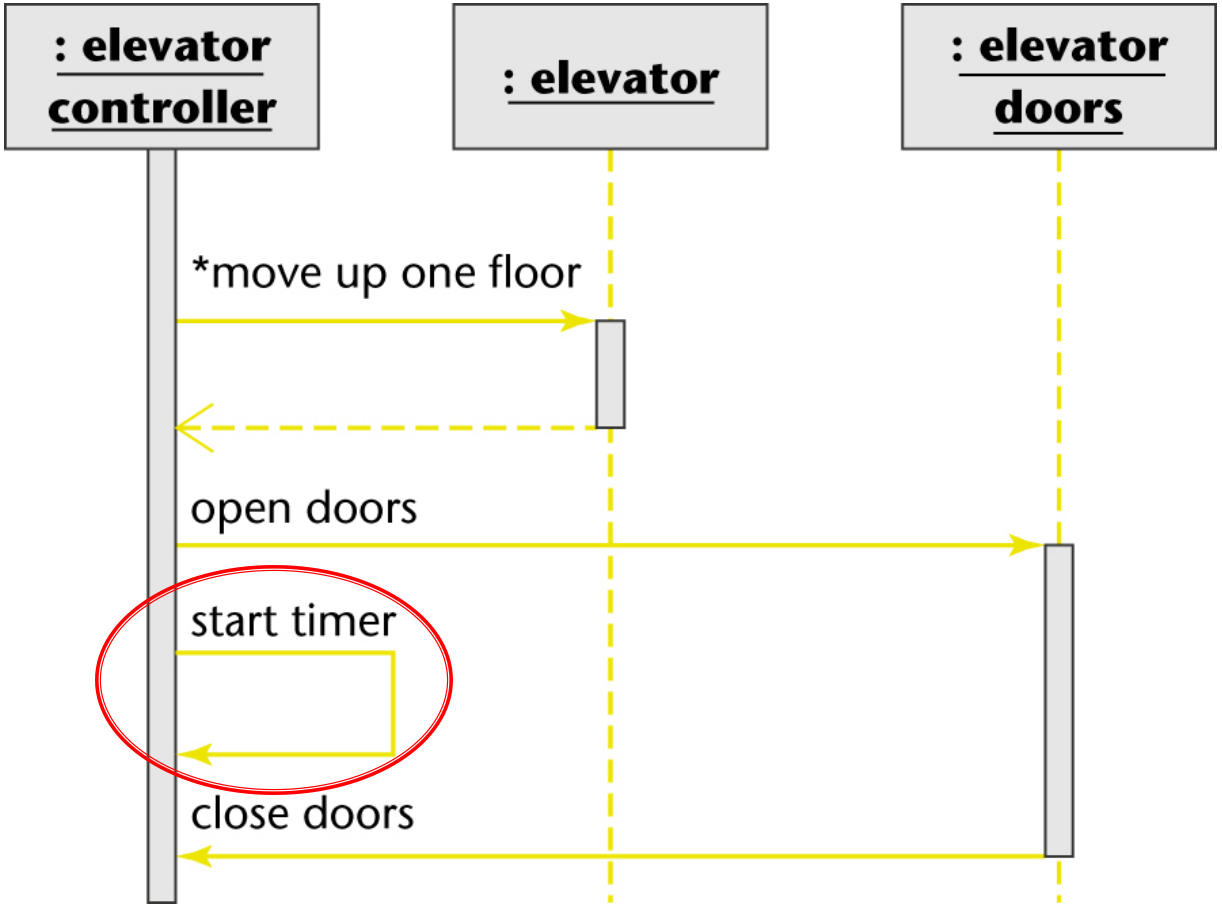
- show how objects interact with one another
- two types of interaction diagrams

Return message

- Optional
- No need to describe return message
- Guard on message

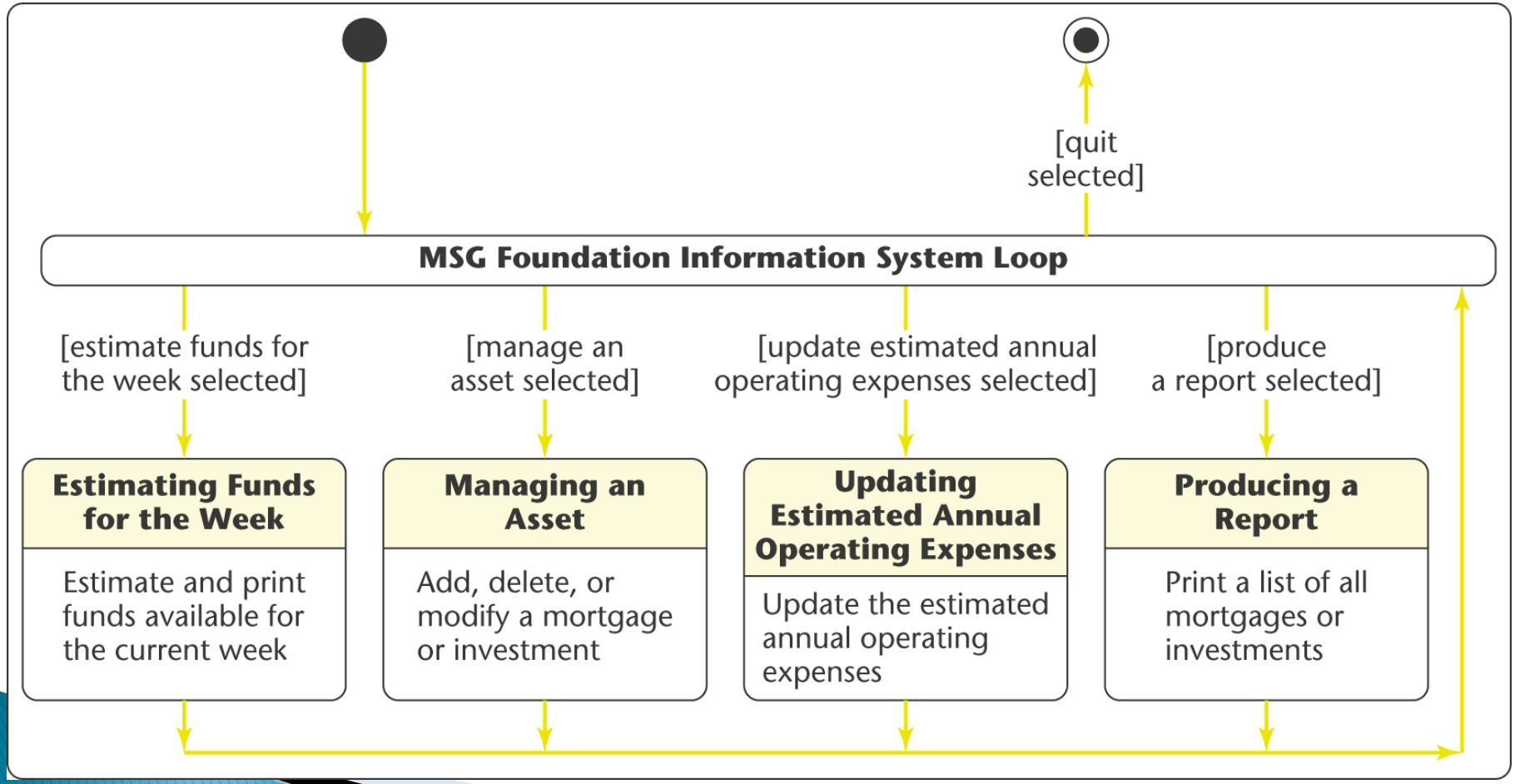
Iteration in sequence diagram

- Where ?

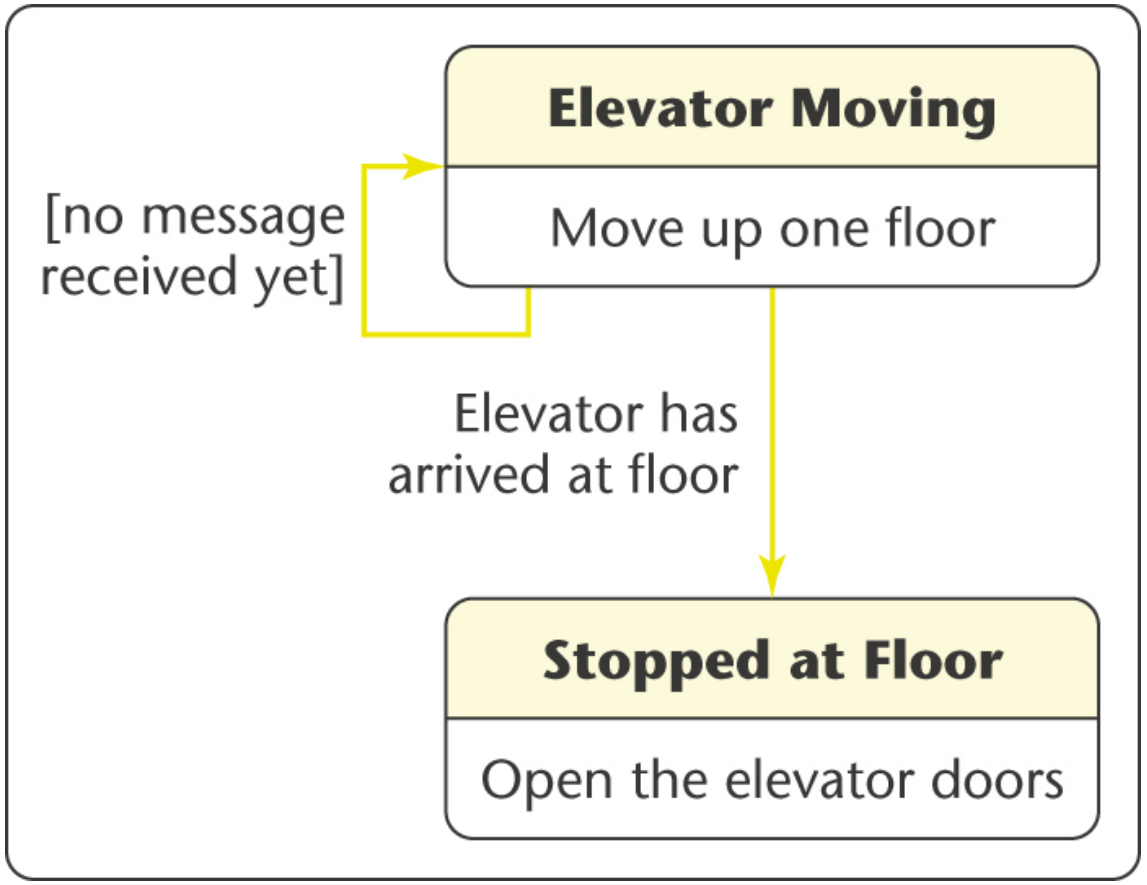


Statecharts

- Statechart with guards



Statechart – event, guard, activity



Statecharts (contd)

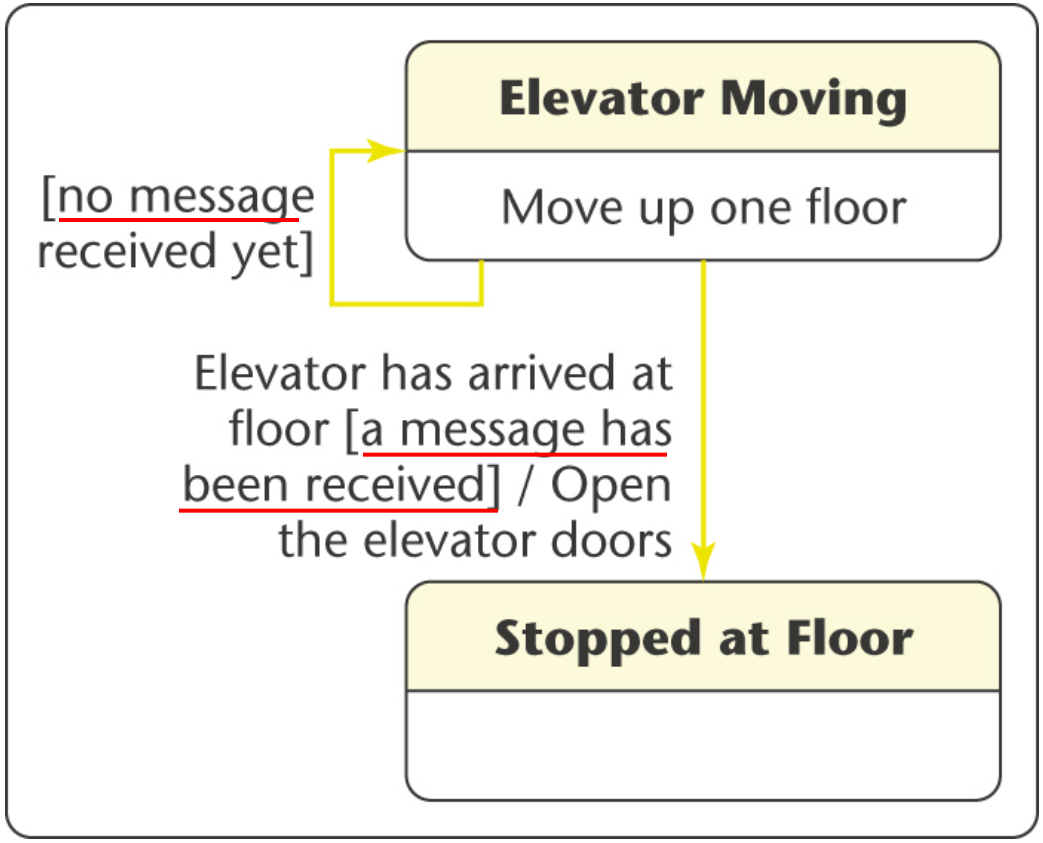
- The most general form of a transition label is
  - event [guard] / action
  - If
    - event

    has taken place and
    - [guard]

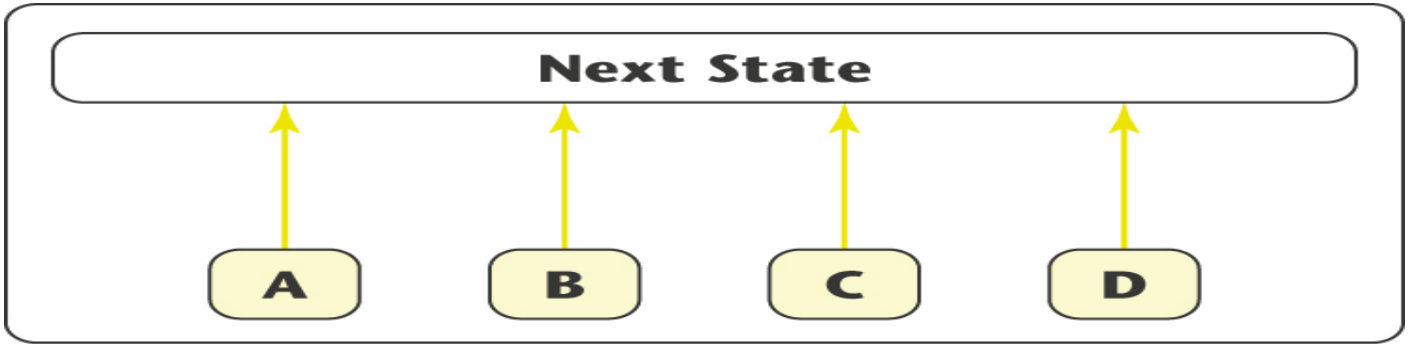
    is true, the transition occurs, and, while it is occurring,
    - action

    is performed

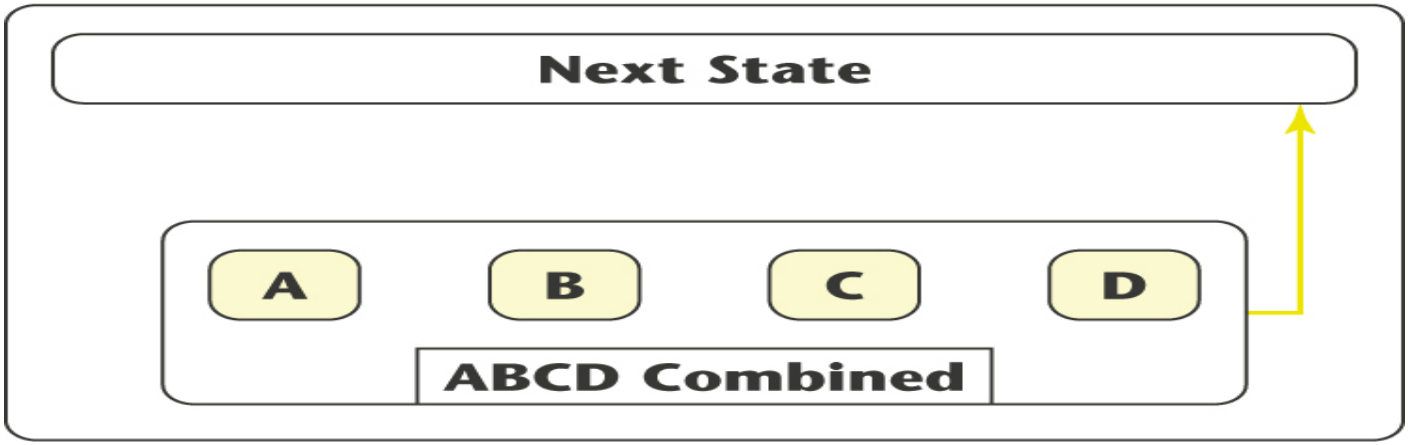
Statecharts – event [guard] / action



Statecharts (contd)

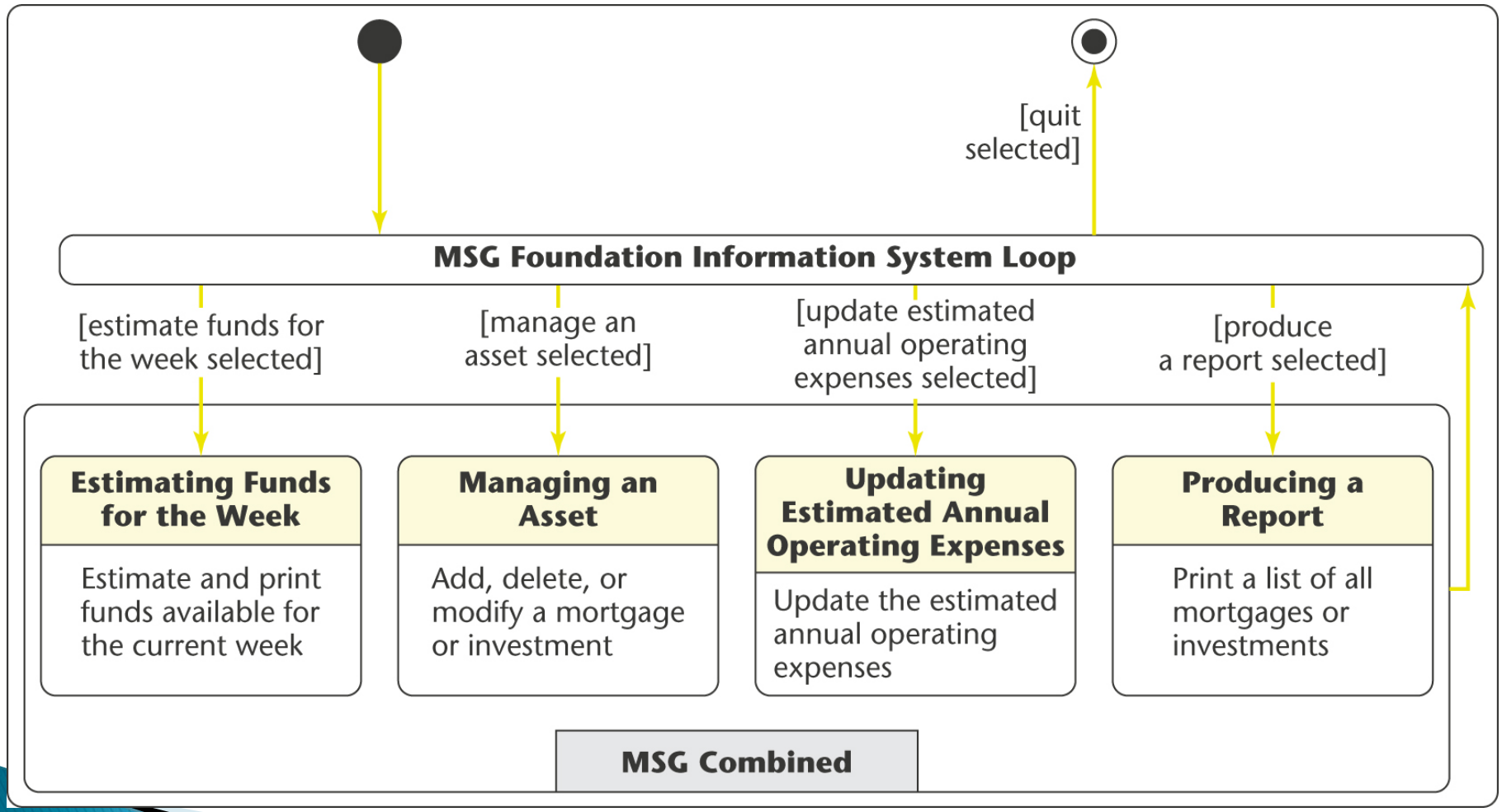


(a)



(b)

Statecharts (contd)

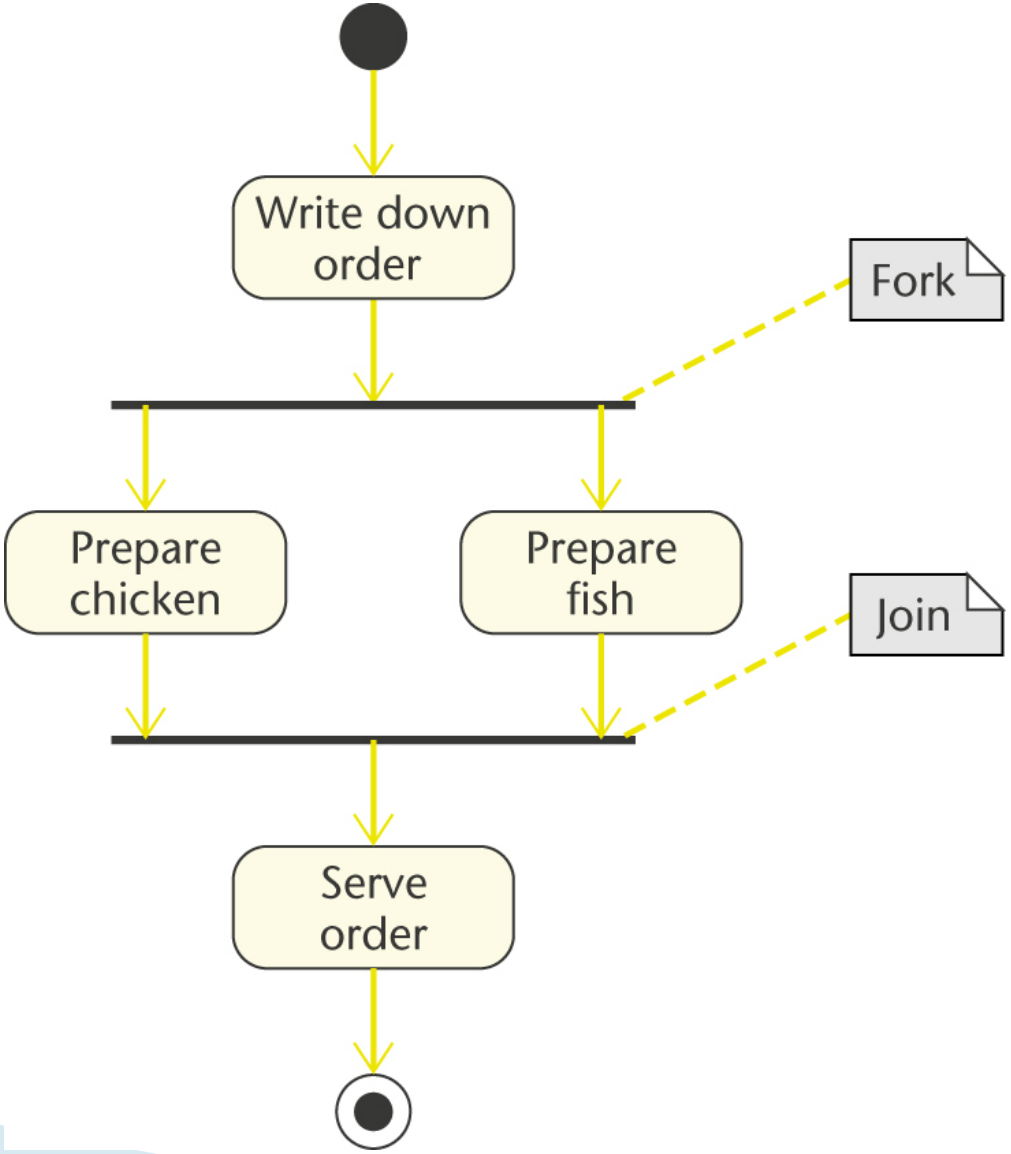


Activity Diagrams

- Activity diagrams show how various events are coordinated
  - Used when activities are carried on in parallel

- Example:
  - One diner orders chicken, the other fish
  - The waiter writes down their order, and hands it to the chef
  - The meal is served only when both dishes have been prepared

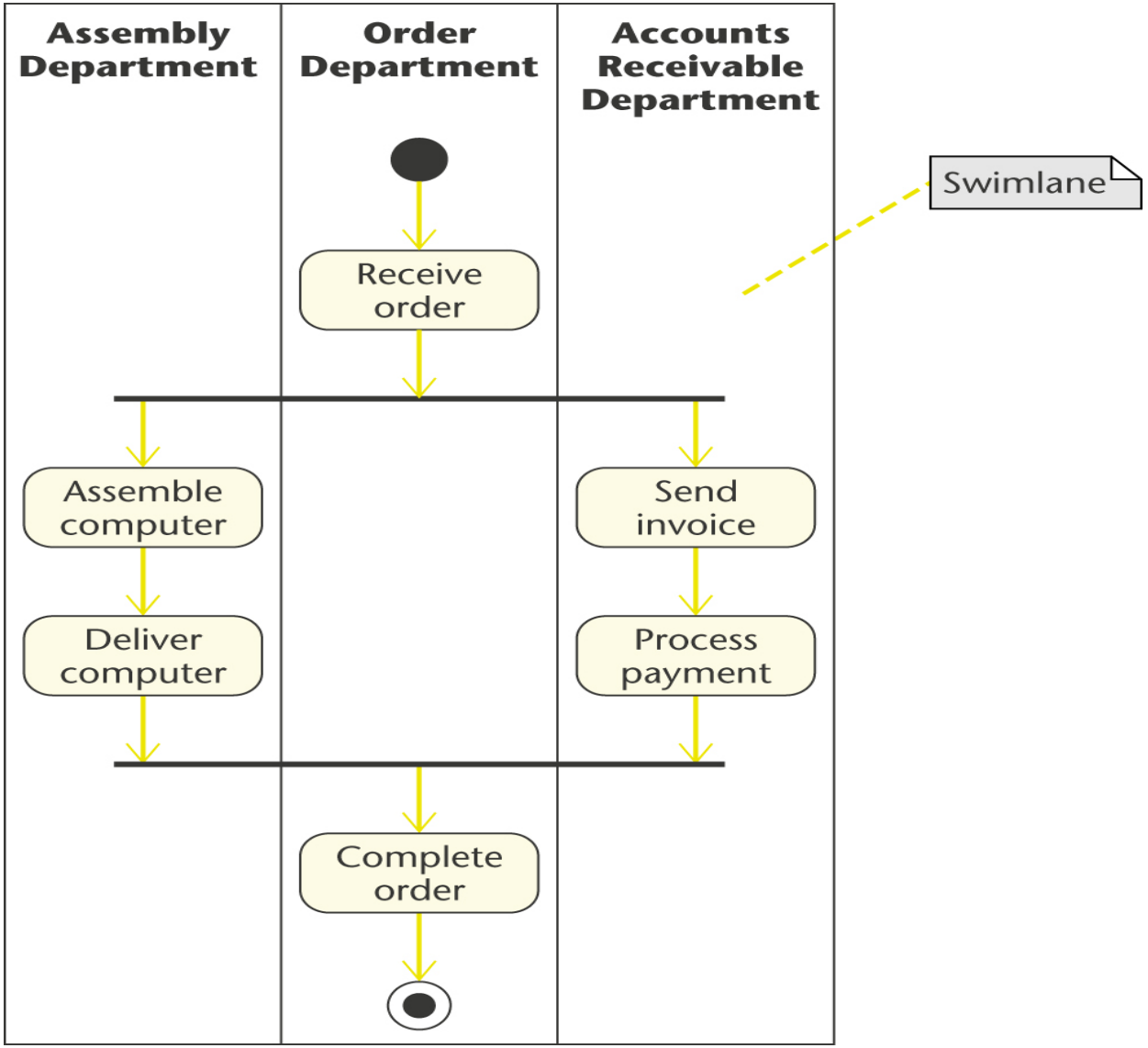
Activity Diagrams (contd)



Activity Diagrams (contd)

- A *fork* has
  - One incoming transition, and
  - Many outgoing transitions, each of which starts an activity to be executed in parallel with the other activities

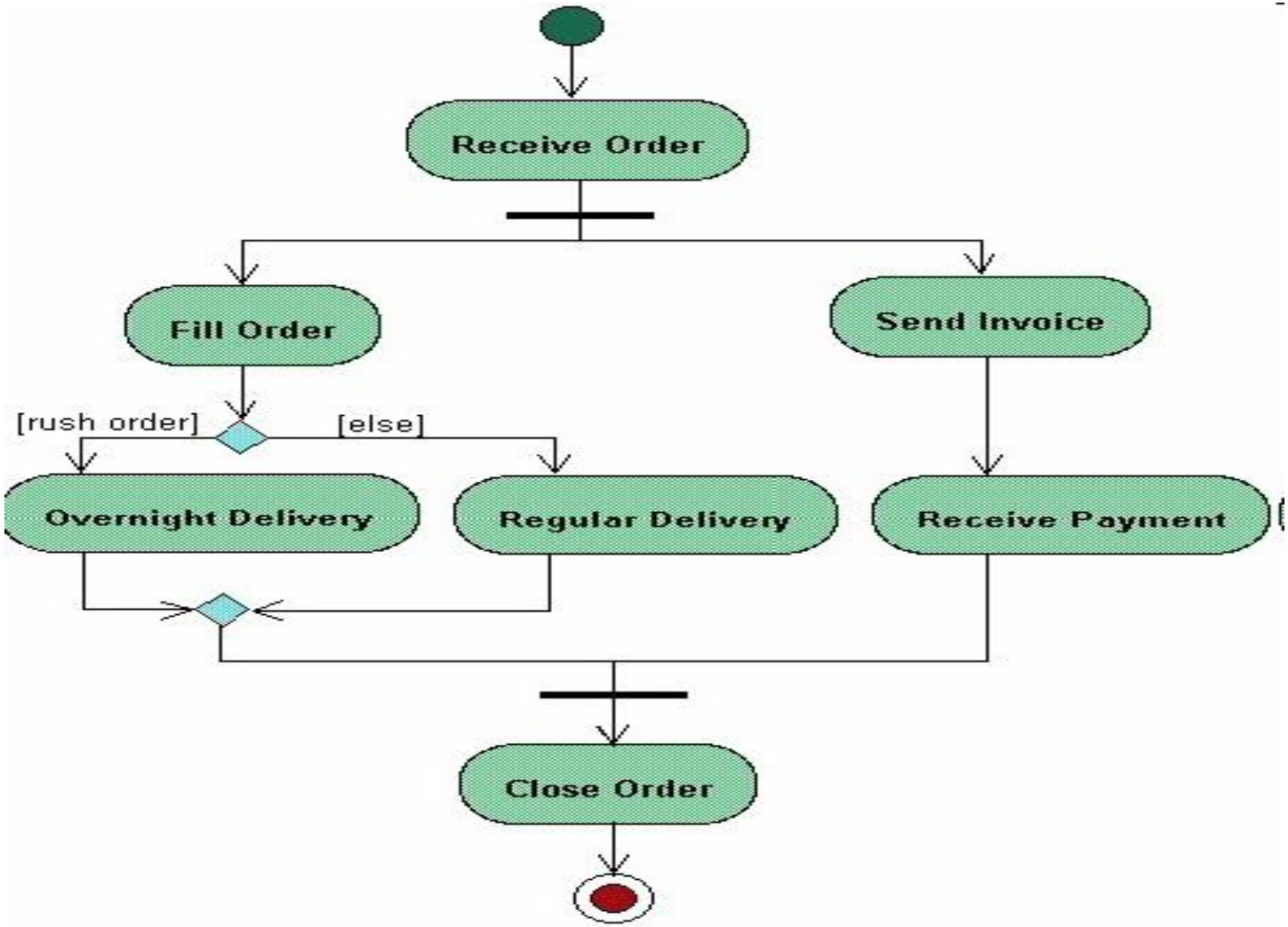
- A *join* has
  - Many incoming transitions, each of which lead from an activity executed in parallel with the other activities, and
  - One outgoing transition that is started when all the parallel activities have been completed


Assembly Department
Order Department
Accounts Receivable Department
Swimlane
Receive order
Assemble computer
Deliver computer
Send invoice
Process payment
Complete order

Activity Diagrams (contd)

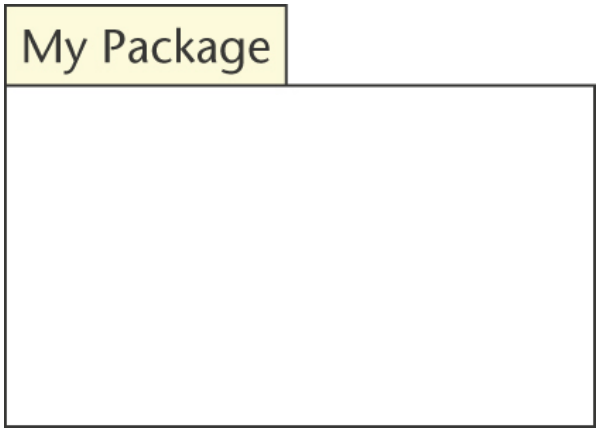
- The three departments involved
  - Assembly Department
  - Order Department
  - Accounts Receivable Department

  are each in their own *swimlane*


Receive Order
Fill Order
Send Invoice
[rush order]
[else]
Overnight Delivery
Regular Delivery
Receive Payment
Close Order

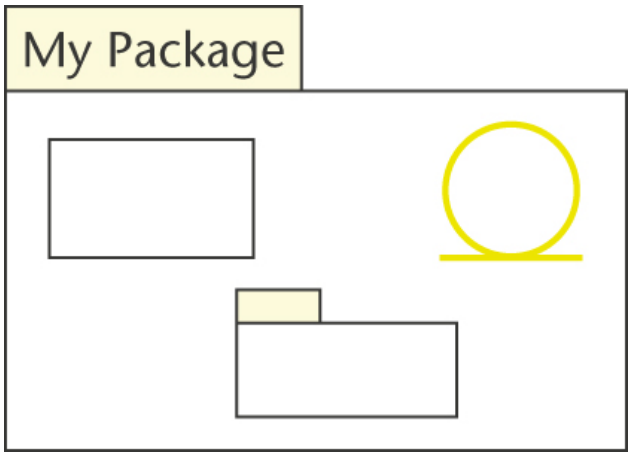
Packages

- A large information system is decomposed into relatively independent packages

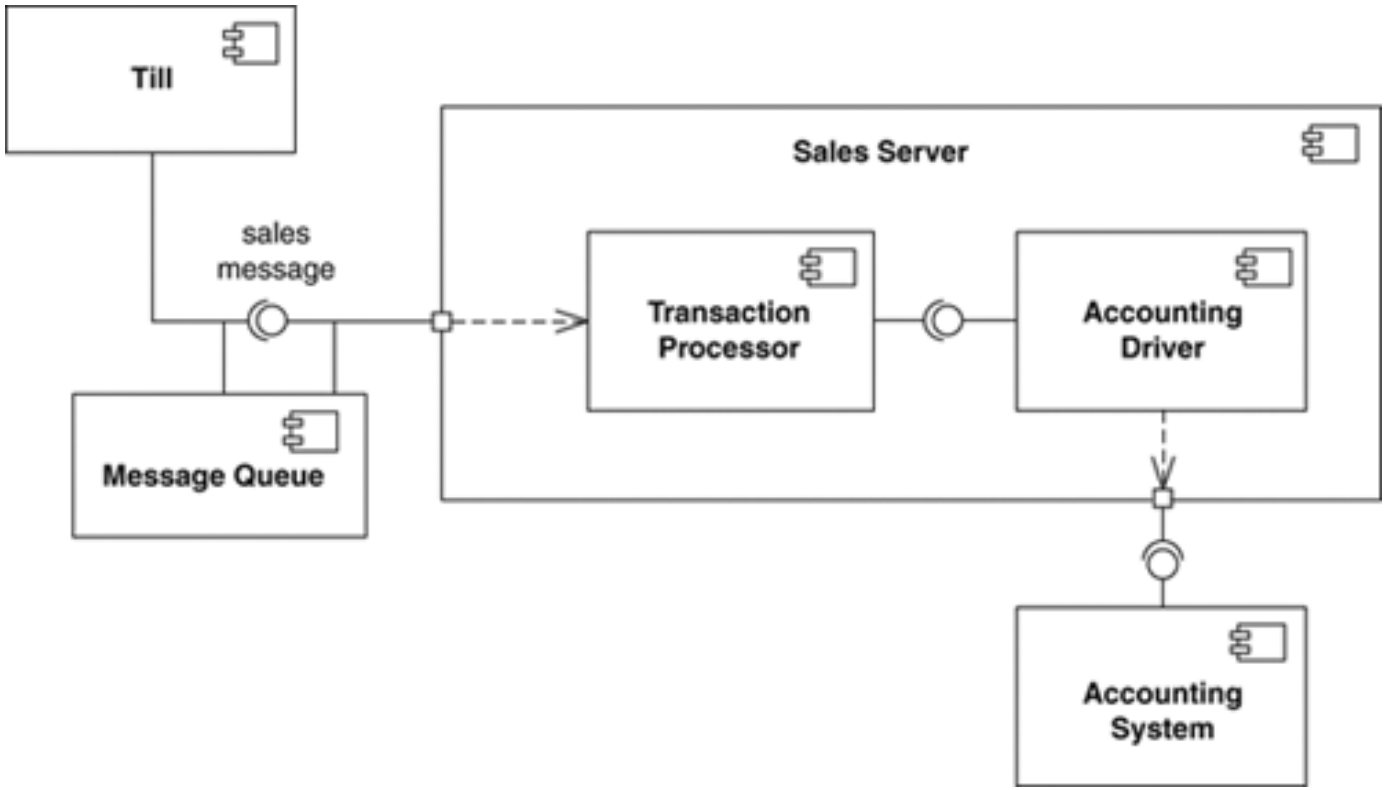


Packages (contd)

- Example showing the contents of My Package



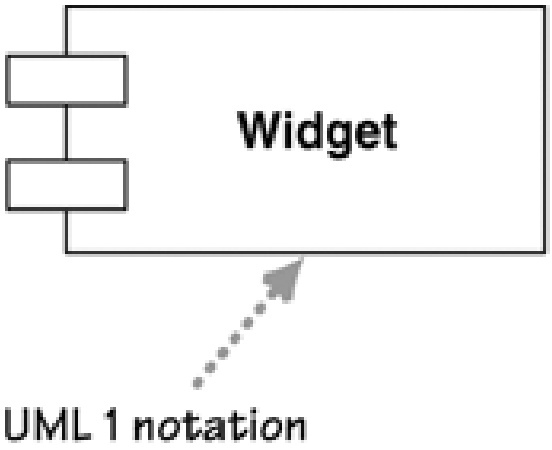
Component Diagram

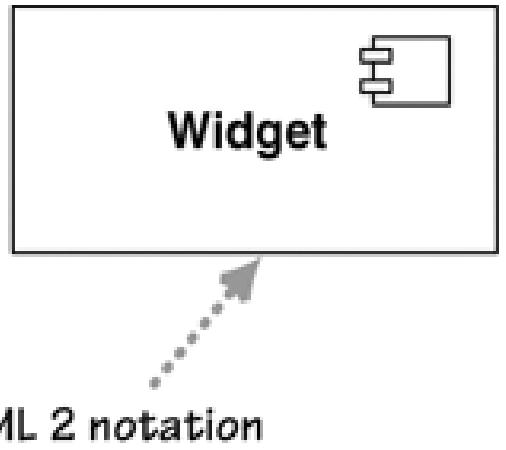


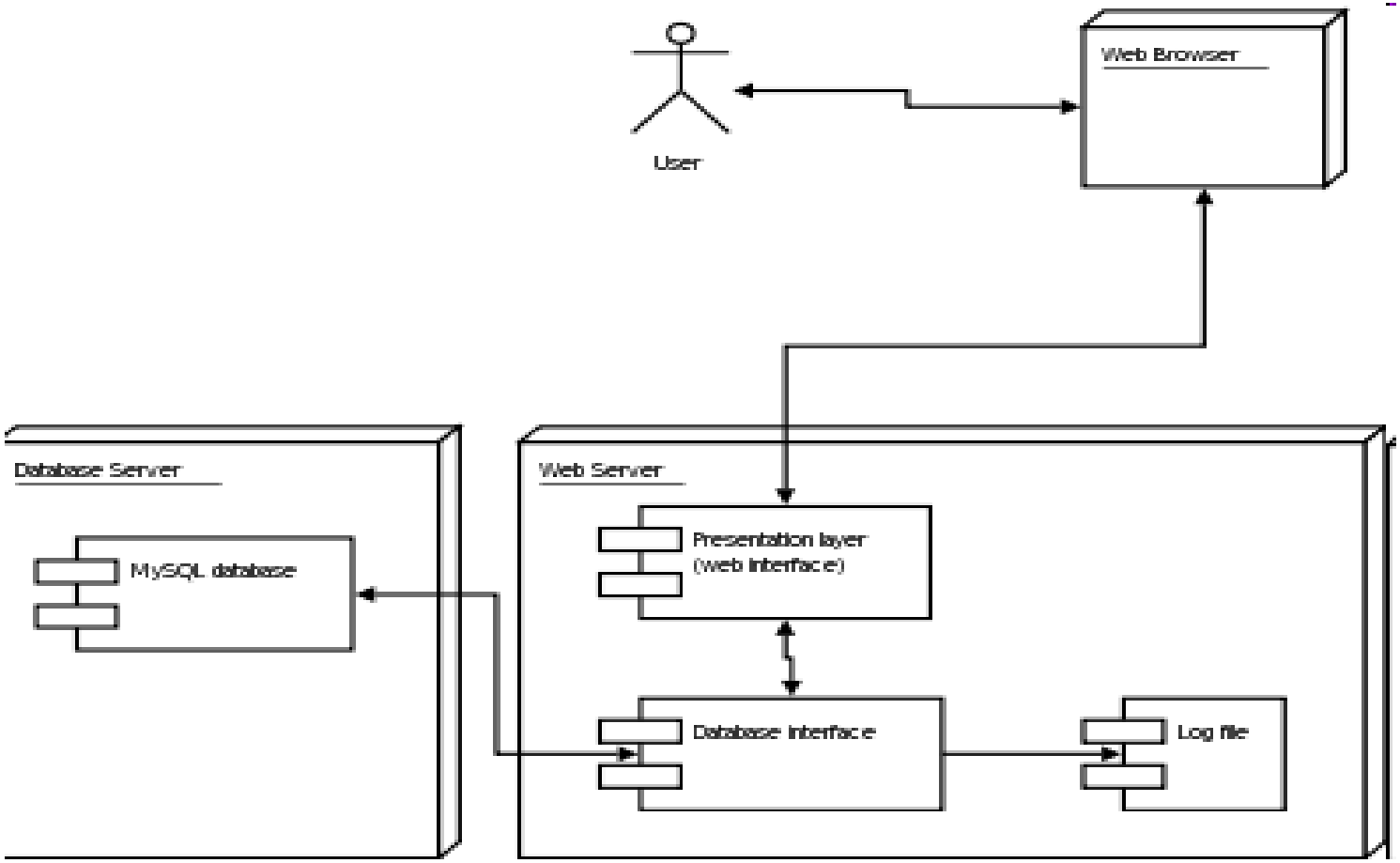
Component Diagram

- There is little difference b/w component diagram and class diagram

Component diagram notation





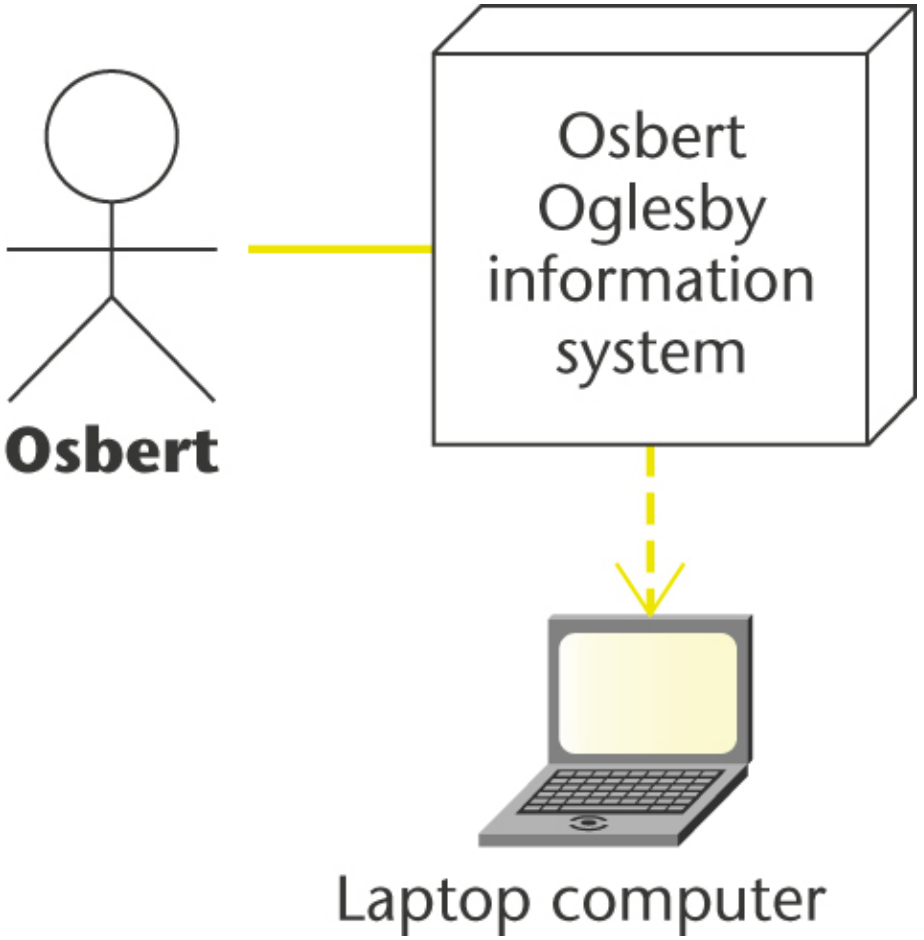

User
Web Browser
Database Server
MySQL database
Web Server
Presentation layer
(web interface)
Database interface
Log file

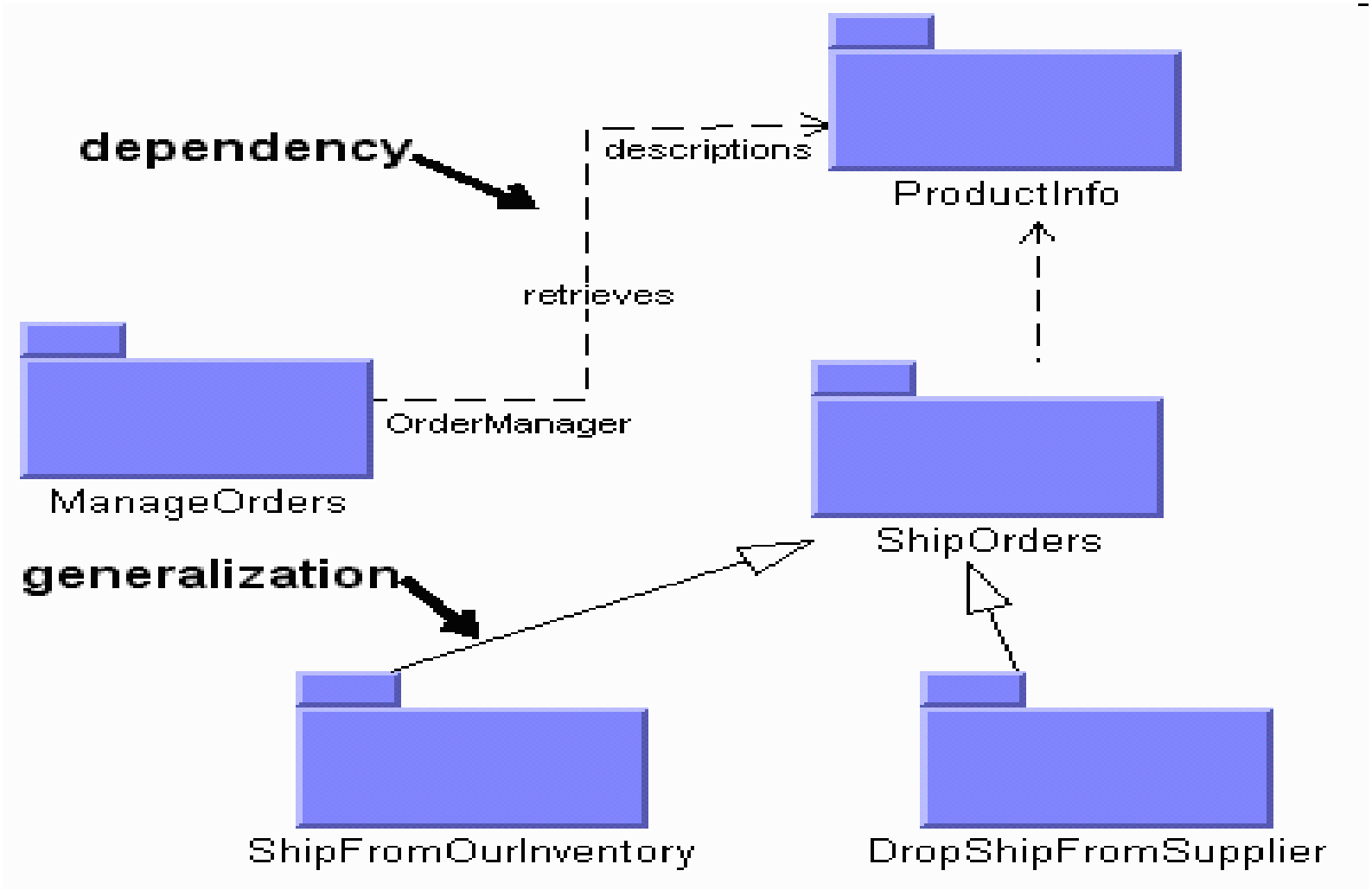
Deployment Diagrams

- A deployment diagram shows on which hardware component each software component is installed (or *deployed*)
- It also shows the communication links between the hardware components

Deployment Diagrams (contd)

- Example:




dependency
descriptions
ProductInfo
retrieves
OrderManager
ManageOrders
ShipOrders
generalization
ShipFromOurInventory
DropShipFromSupplier

Review of UML Diagrams

- Some diagrams that could be confused include:
  - A *use case* models the interaction between actors and the information system
  - A *use-case diagram* is a single diagram that incorporates a number of use cases
  - A *class diagram* is a model of the classes showing the static relationships between them
    - Including association and generalization

Review of UML Diagrams

- A *statechart* shows
  - States (specific values of attributes of objects)
  - Events that cause transitions between states (subject to guards), and
  - Actions taken by objects
- An *interaction diagram* (sequence diagram or collaboration diagram) shows how objects interact as messages are passed between them
- An *activity diagram* shows how events that occur at the same time are coordinated

UML and Iteration

- Using just a subset of UML
- Can be incomplete, but valid
- -> fit to the Unified Process

Exercise

- Draw an activity diagram showing the following blog review process:

1. Author submits content
2. Editor reviews content

3a. If editor approves, the content is published to the site

3b. Otherwise, the author is sent a rejection notification