Chapter 3
Use Case

Dr. Supakit Nootyaskool
Faculty of Information Technology
King Mongkut's Institute of Technology Ladkrabang

Reference

[1] Satzinger, Jackson, Burd, "System Analysis and Design in a Changing World, 2012

[2] Alan Dennis, Barbara Haley Wixom David Tegarden, "System Analysis and Design with UML 2nd", 2005

[3] http://metagear.de/articles/uml-introduction/index.html

Learning objective (1)

- Explain why identifying use case is the key to defining function requirement
- Describe the two technique for identifying use cases
- Apply the user goal technique to identify use case
- Apply the even decomposition technique to identify use case
- Apply the CRUD technique to validate and refine the list of use cases

Learning objective (2)

- Describe the notation and purpose for the use case diagram
- Draw use case diagram by actor and by subsystem

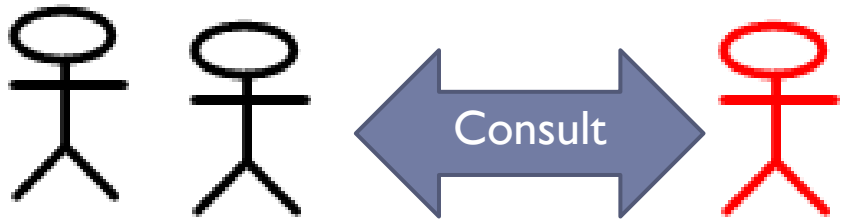
Outline

- User goals technique
- Event decomposition technique
- CRUD
- Use case diagram
- ?

Opening case:
Waiters on call meal-delivery system

The restaurant by Sue and Tom

- Start 2008
- Two restaurant and one driver
- Manual order managing
- Business rapidly expanded, so they think use computer to mange ordering.
- They consult Sam

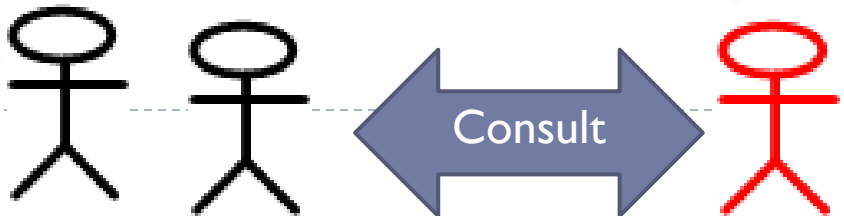


Opening case:
Waiters on call meal-delivery system

When a customer order

1. I need to record it information to right restaurant.
2. I need to know which driver to ask to pickup the order. So the driver call in and tell men when they free.
3. Perhaps this could be include a smart phone or Ipad app.
4. Some time customer want to change the order; so I need to my hands on the original order. And notify the restaurant to make change

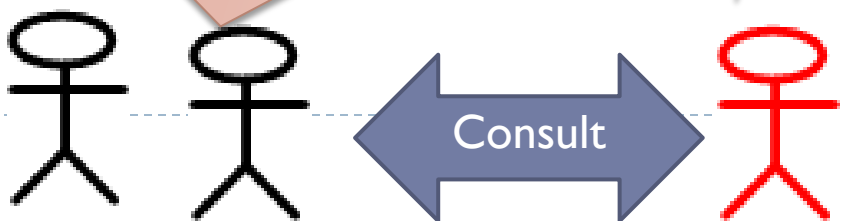
What sort of events happen when you are running your business that make you want to reach for a computer?



Opening case:
Waiters on call meal-delivery system

1. The driver get a copy of the bill directly from the restaurant when they pickup the meal.
2. The driver collect that amount plus a service charge.
3. We need to create deposit slip for the bank for the day's total receipts
4. At the end of each week, we calculate what we owe each restaurant at the agreed-t wholesale price and send each statement and check.

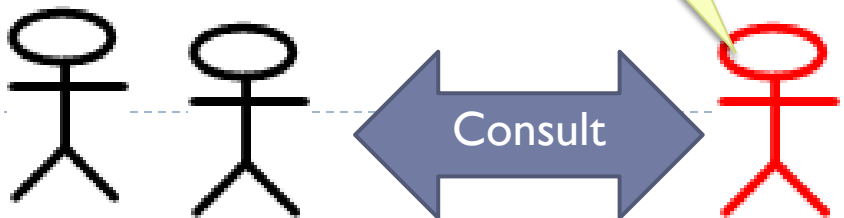
How do you handle the money?



Opening case:
Waiters on call meal-delivery system

Draw a diagram and tell,

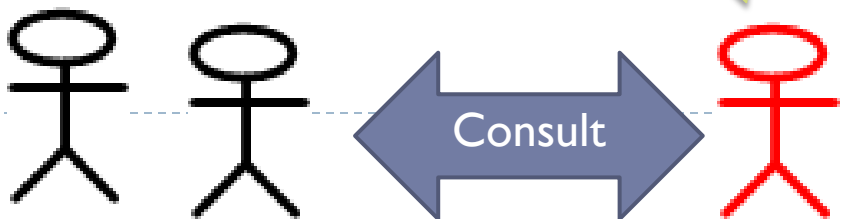
1) A customer call in to place and order, so you need to record an order
2) A customer call back to change an order, you need to update an order
3) A driver is finished with a delivery, so you need to record delivery completion.
4) A driver reports for work, so you need to sign in the driver
5) A deriver submits the day's receipts, so you need to reconcile driver receipts.



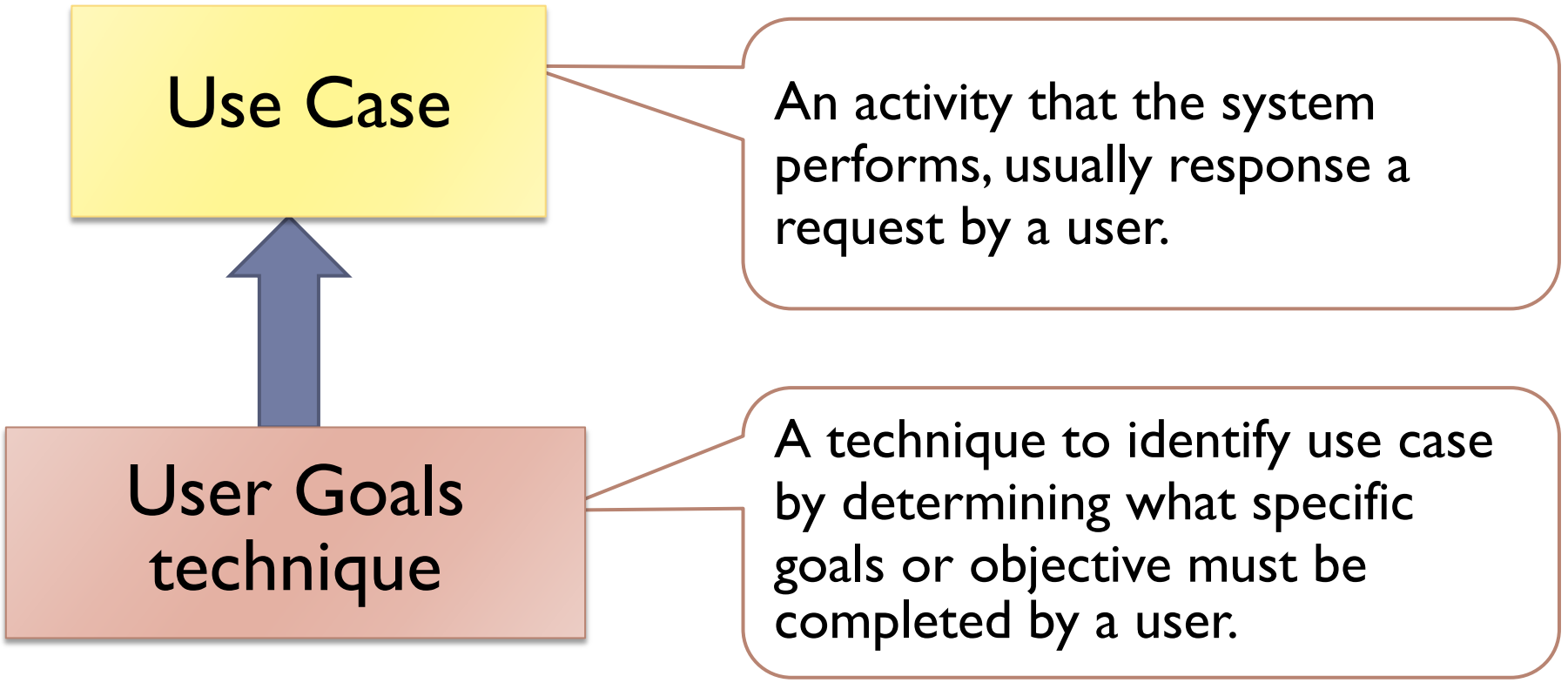
Opening case:
Waiters on call meal-delivery system

The system need to procdure information at speicific point in time;

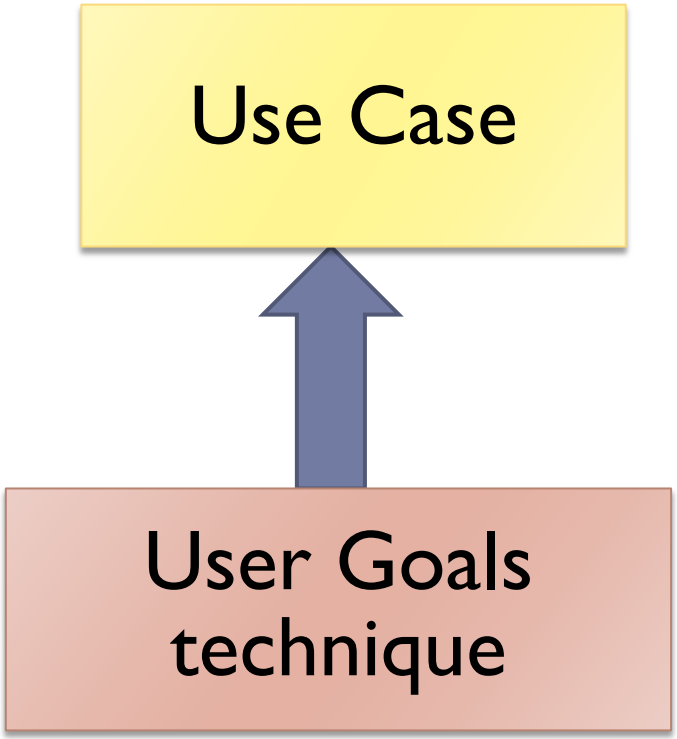
1. Produce a end-of-day deposit slip
2. Produce end of week restaurant payments.
3. Produce weekly sales reports.
4. Produce monthly financial reports.



1. Use cases and User Goals



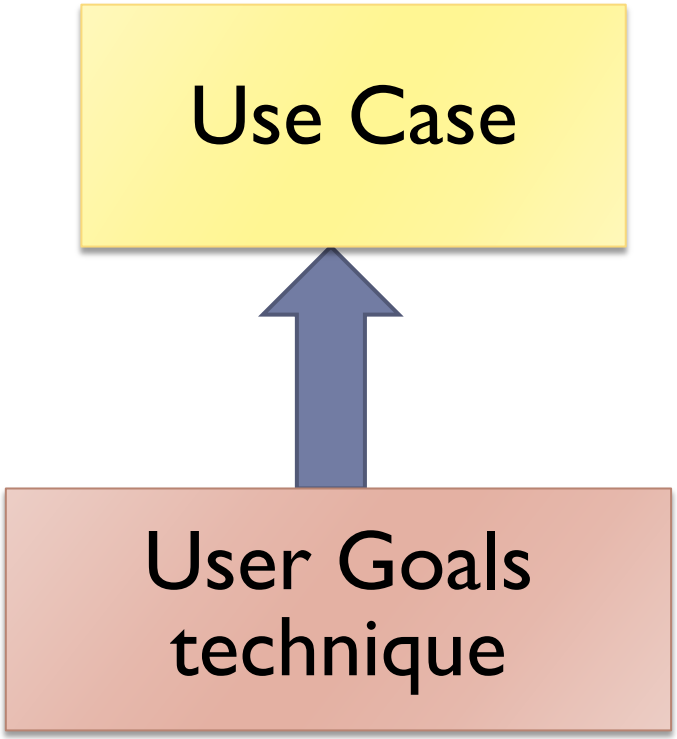
1. Use cases and User Goals



Technique for identifying use case includes these steps:

1) Identify all the potential user for the new system.
2) Classify the potential users in term of their functional rule. {Shipping, Marketing, Sales}
3) Future classify potential users by organizational level. {Operational, Management, Executive}

1. Use cases and User Goals



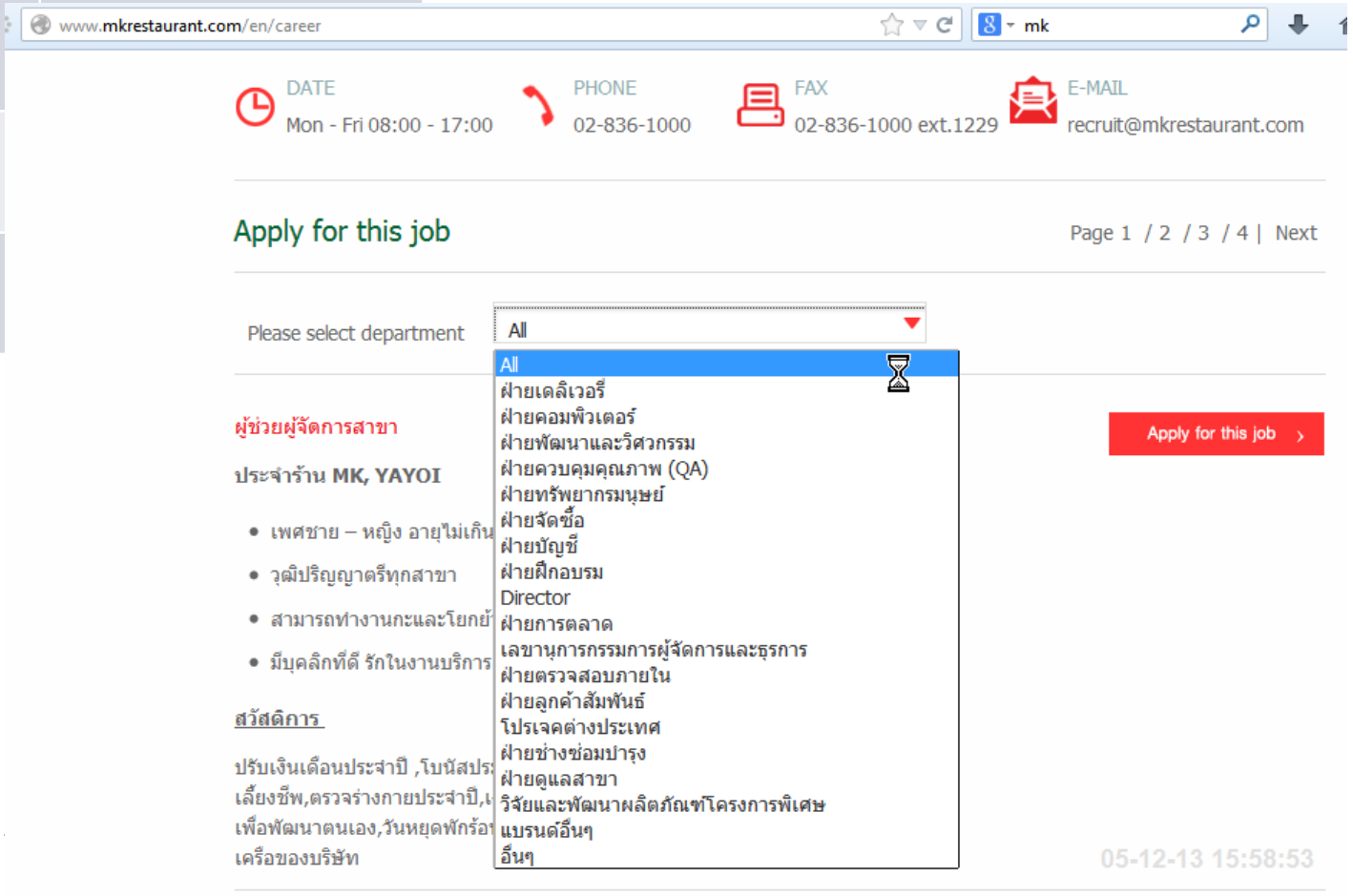
4) Interviewing user to get imagination function they think would add value.
5) Create list of preliminary use case organized by type of user.
6) Look for duplicates with similar use case name and resolve inconsistencies
7) Review list to stackholder

Example RMO

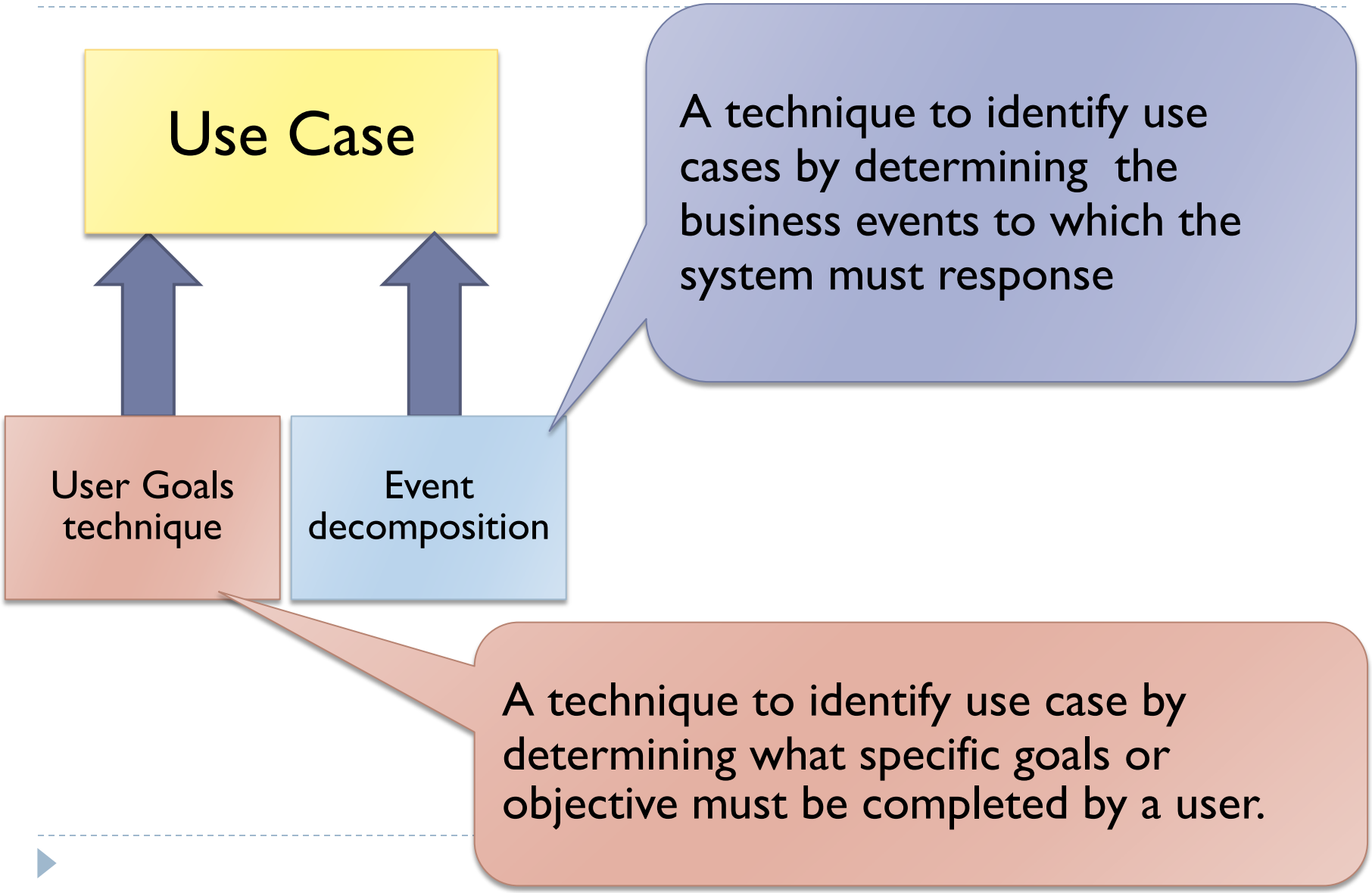
User	User goal and resulting use case
Potential customer	Search for item Fill shopping cart View product rating and comments
Marketing manager	Add/update product information Add/update promotion Produce sales history report
Shipping personnel	Ship items Track shipment Create item return

Question:
Writing user goal of MK restaurant (10Mins)

User	User goal and resulting use case
Waiter	
Cashier	
…	



2. Use cases and Event decomposition

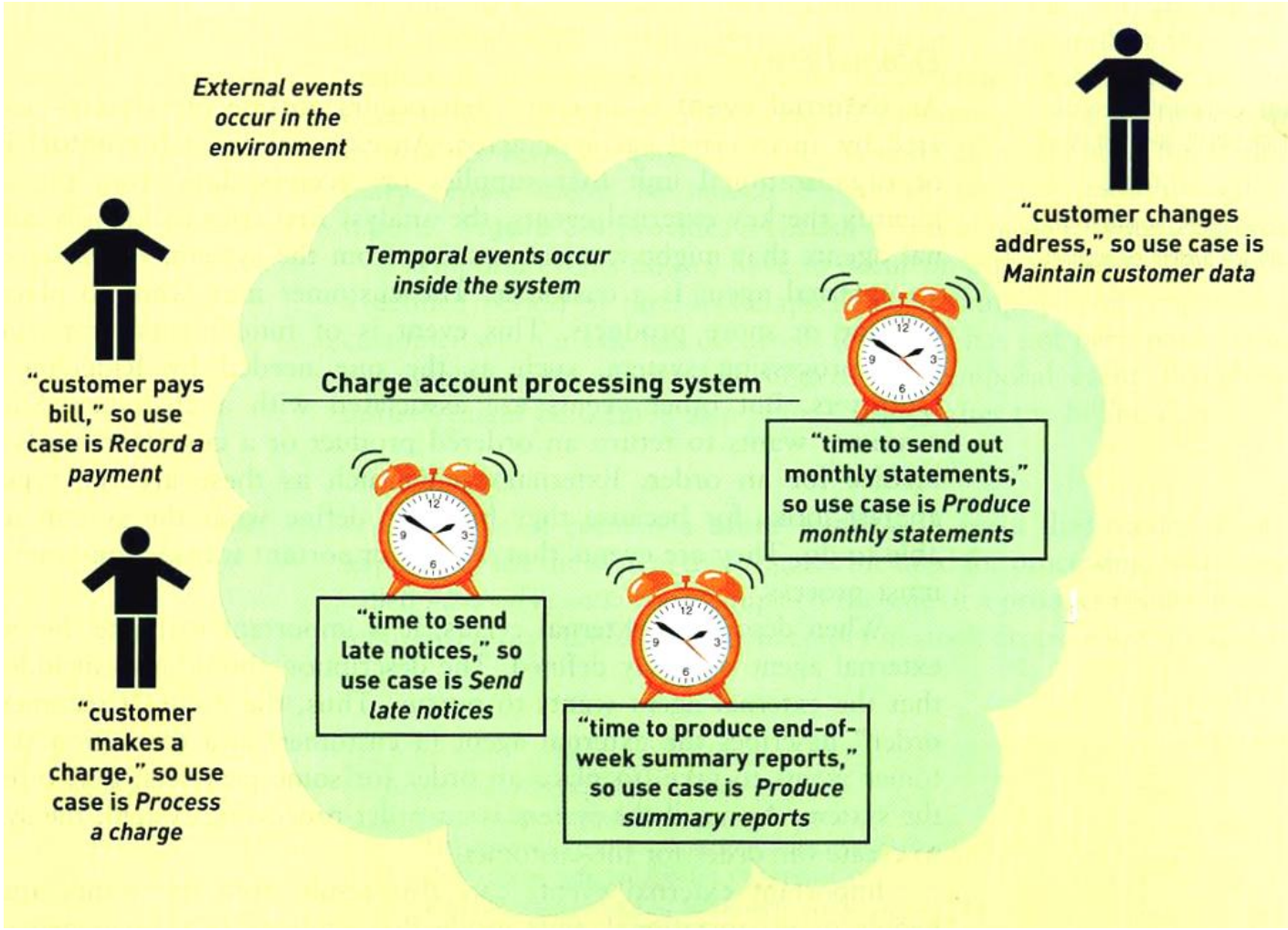


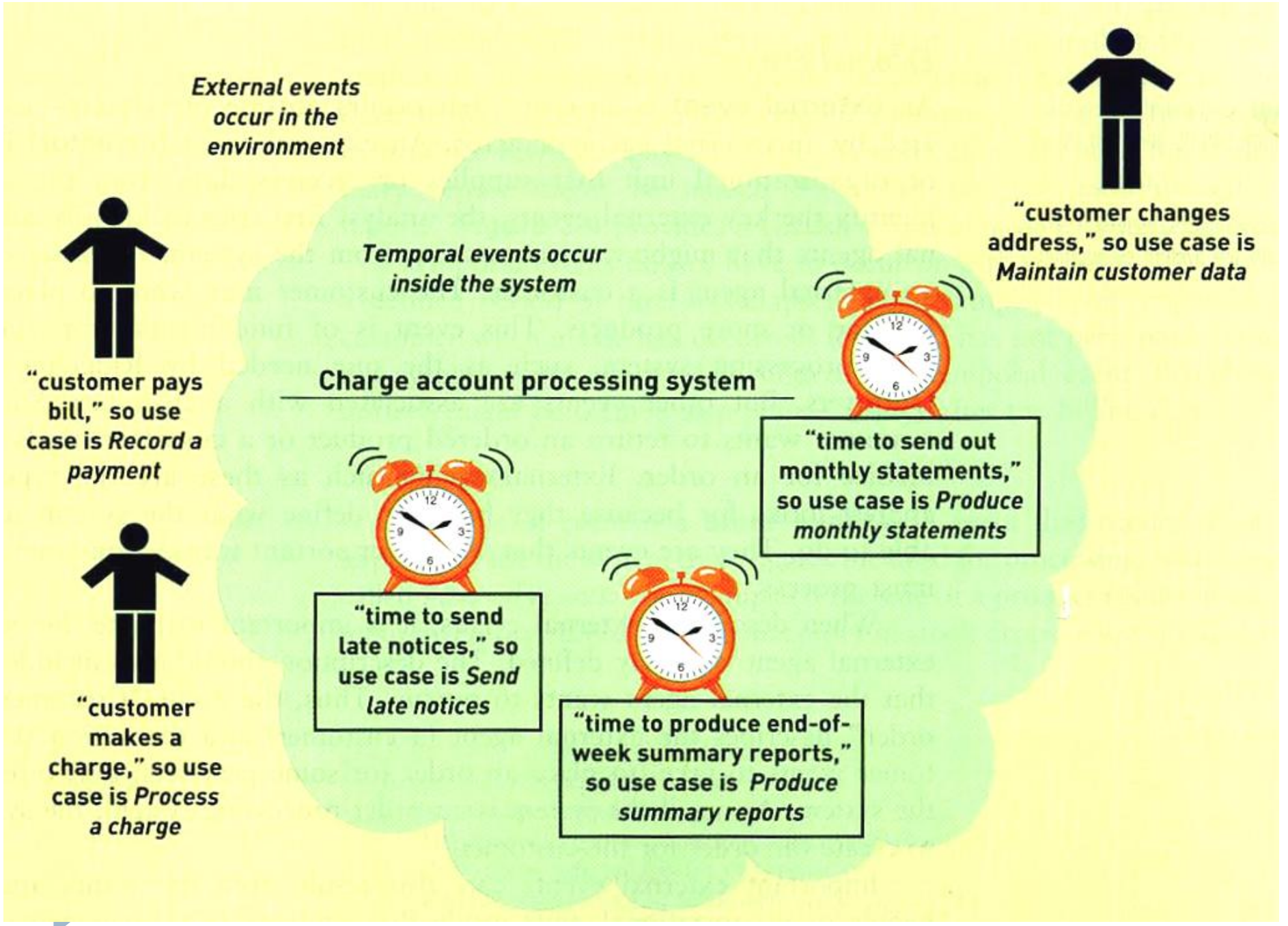
Three events system must response

1. Record a payment
2. Process a change
3. Maintain customer data

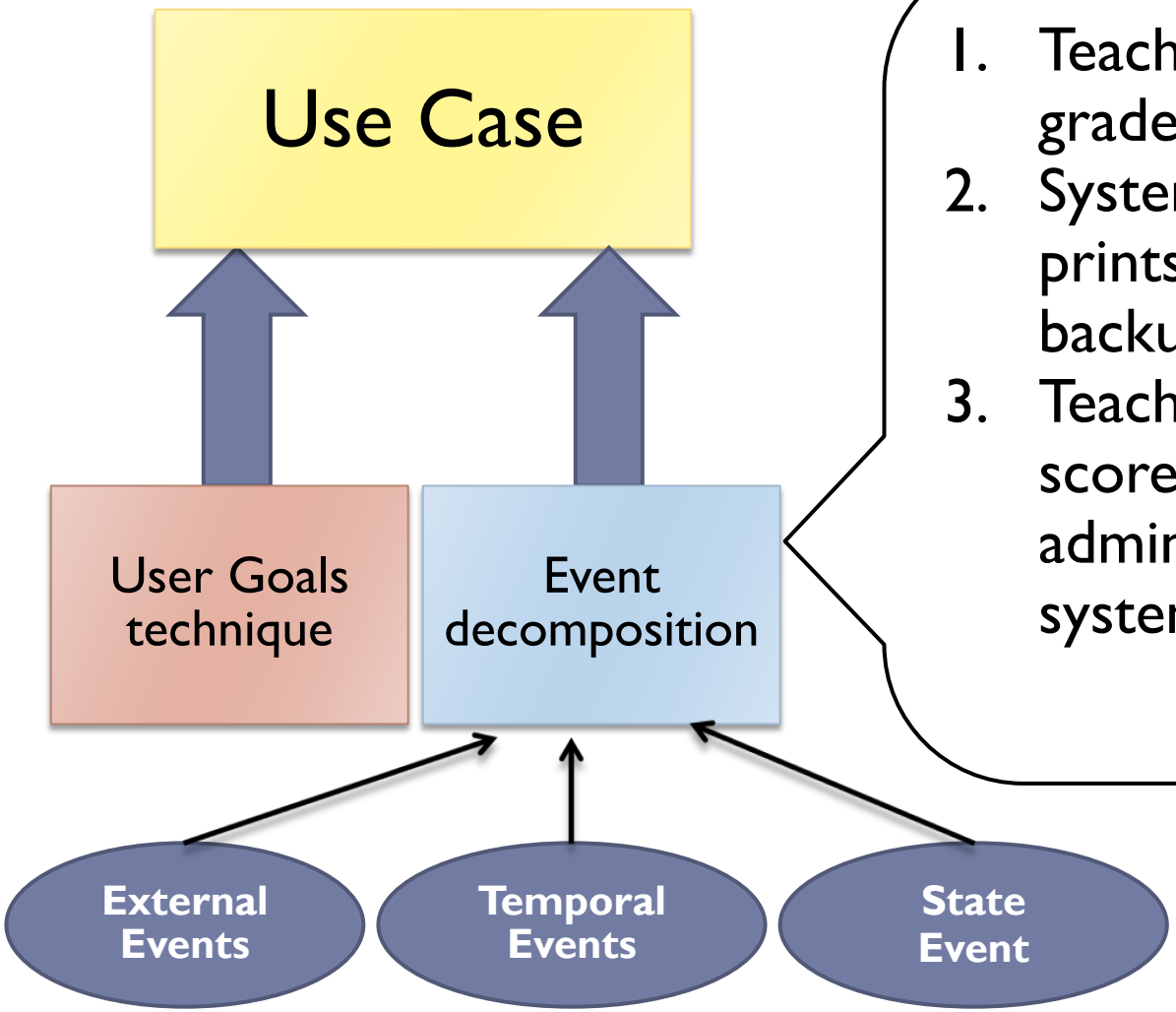
Three trigger events system response

1. Send late notices
2. Produce summary report
3. Produce monthly statements




External events occur in the environment
"customer changes address," so use case is Maintain customer data
Temporal events occur inside the system
"customer pays bill," so use case is Record a payment
Charge account processing system
"time to send out monthly statements," so use case is Produce monthly statements
"time to send late notices," so use case is Send late notices
"time to produce end-of-week summary reports," so use case is Produce summary reports
"customer makes a charge," so use case is Process a charge

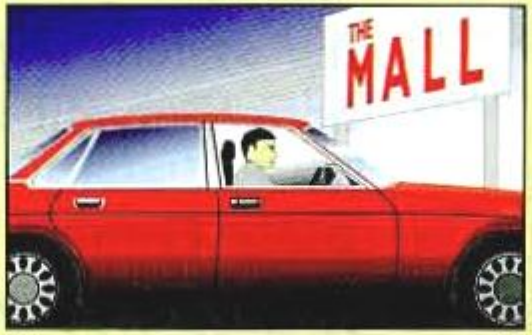
2.1 Type of event



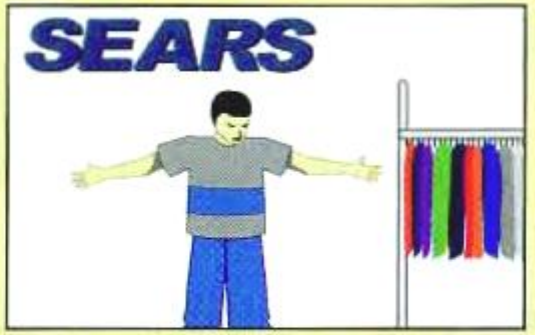
1. Teacher submits student grade.
2. System closes at 16.00 and prints all transaction to backup.
3. Teacher founds a student score incorrectly; so he tell admin to edit data in the system.

Which picture does the system relation?

Customer thinks about getting a new shirt

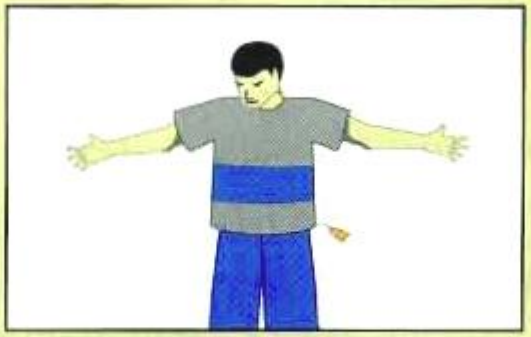


Customer drives to the mall



Customer tries on a shirt at Sears

Customer goes to Walmart



Customer tries on a shirt at Walmart

Customer buys a shirt


CATALOG
Customer requests a catalog
Customer wants to check item availability
Customer places an order
Customer changes or cancels an order
Customer wants to check order status
Account Information
Customer updates account information
RETURNS
Customer returns the item

Don't worry much about these until you are considering design issues

User wants to log on to the system

User wants to change the password

User wants to change preference settings

System crash requires database recovery

Time to back up the database

Time to require the user to change the password

3. CRUD (Create Read/report Update Delete)

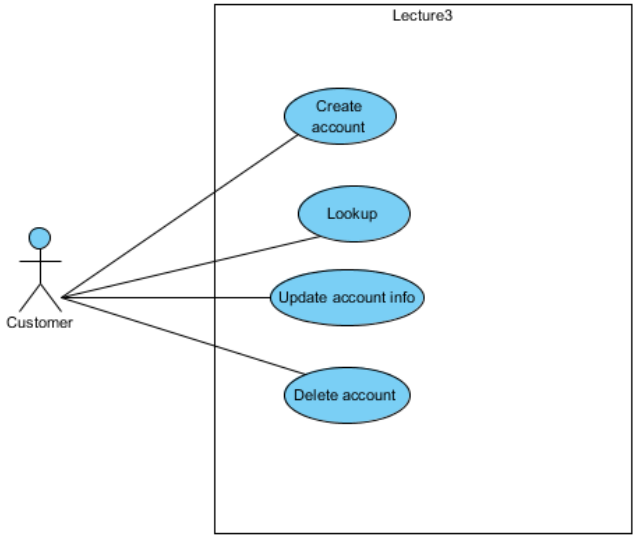
- CRUD used to verify use cases

Data entity/domain class	CRUD	Verified use case
Customer	Create	Create customer account
	Read/report	Look up customer Produce customer usage report
	Update	Process account adjustment Update customer account
	Delete	Update customer account (to archive)

3. CRUD (Create Read/report Update Delete)

- CRUD used to verify use cases

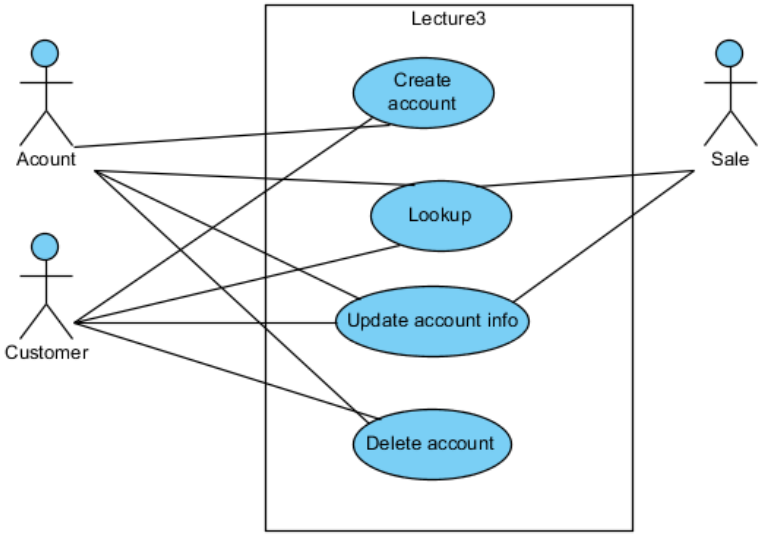
Data entity/domain class	CRUD	Verified use case
Customer	Create	Create customer account
	Read/report	Look up customer Produce customer usage report
	Update	Process account adjustment Update customer account
	Delete	Update customer account (to archive)



3. CRUD (Create Read/report Update Delete)

- CRUD used to verify use cases

Use case vs. entity/domain class	Customer	Account	Sale	Adjustment
Create customer account	C	C		
Look up customer	R	R		
Produce customer usage report	R	R	R	
Process account adjustment	R	U	R	C
Update customer account	UD (archive)	UD (archive)		

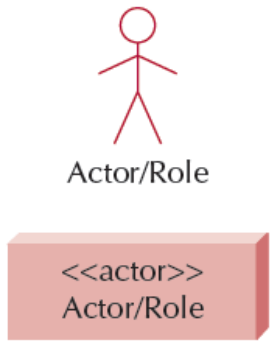


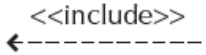
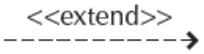
CRUD Matrix for Internet Sale System

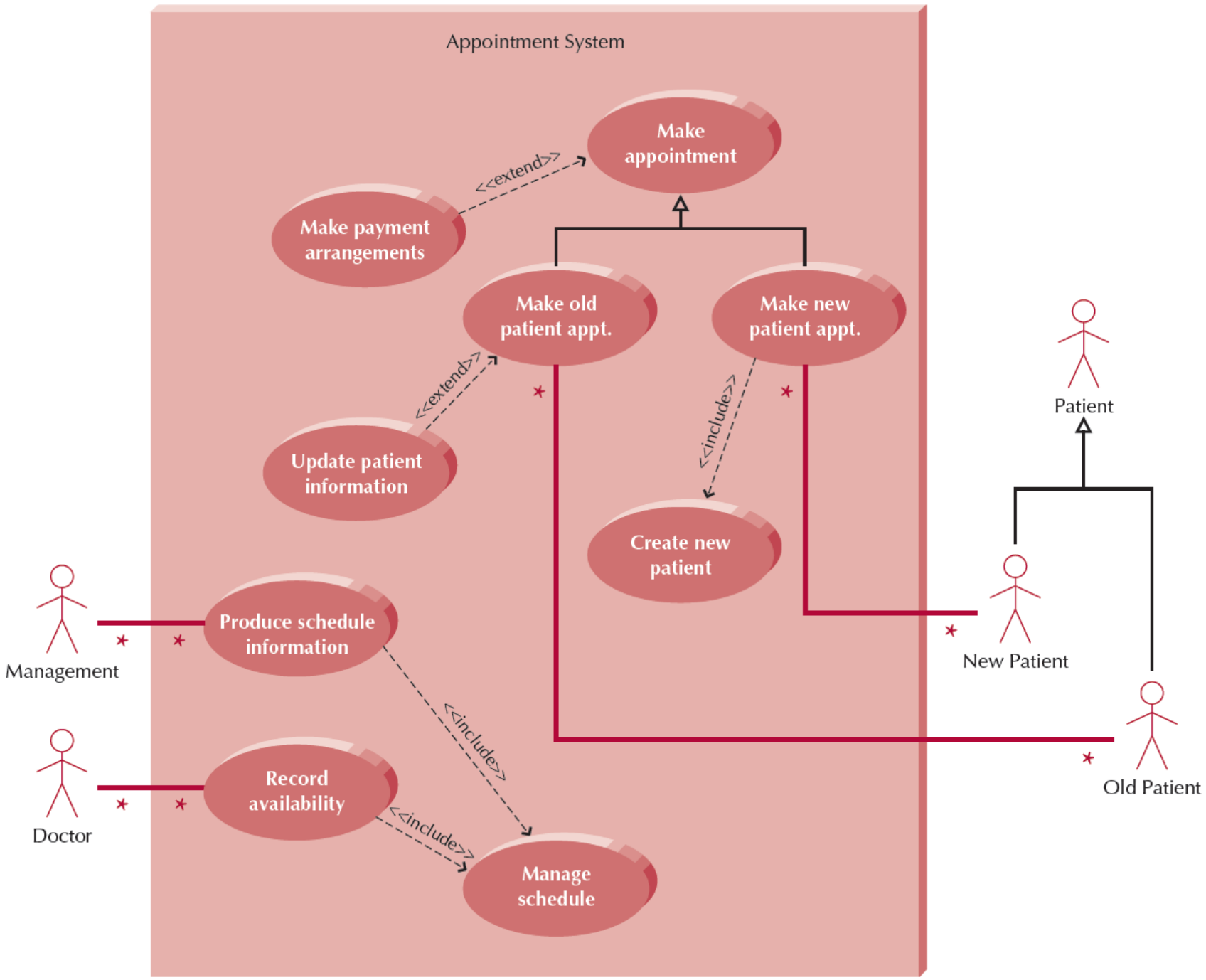
	Customer	SearchReq	CDList	CD	Mkt Info	Review	Artist Info	Sample Clip	Shopping Cart	Order
Customer		R							U	C
SearchReq			CR							
CDList										
CD					R					
Mkt Info						U	U	U		
Review										
Artist Info										
Sample Clip										
Shopping Cart										
Order										

Use case diagram

- The use case diagram is the UML model used to graphically show the use cases and their relationship to user.

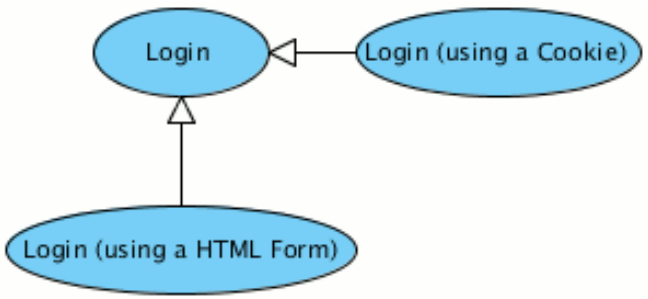
An Actor: ■ Is a person or system that derives benefit from and is external to the subject ■ Is depicted as either a stick figure (default) or if a non-human actor is involved, as a rectangle with <<actor>> in it (alternative) ■ Is labeled with its role ■ Can be associated with other actors using a specialization/superclass association, denoted by an arrow with a hollow arrowhead ■ Are placed outside the subject boundary	
A Use Case: ■ Represents a major piece of system functionality ■ Can extend another use case ■ Can include another use case ■ Is placed inside the system boundary ■ Is labeled with a descriptive verb-noun phrase	
A Subject Boundary: ■ Includes the name of the subject inside or on top ■ Represents the scope of the subject, e.g., a system or an individual business process	

An Association Relationship: ■ Links an actor with the use case(s) with which it interacts	
An Include Relationship: ■ Represents the inclusion of the functionality of one use case within another ■ The arrow is drawn from the base use case to the included use case	
An Extend Relationship: ■ Represents the extension of the use case to include optional behavior ■ The arrow is drawn from the extension use case to the base use case	
A Generalization Relationship: ■ Represents a specialized use case to a more generalized one ■ The arrow is drawn from the specialized use case to the base use case	

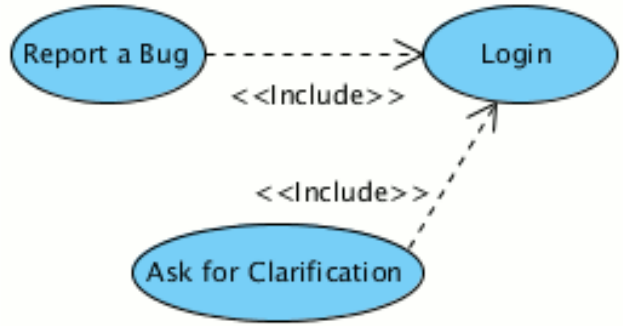

Appointment System
Make appointment
Make payment arrangements
<<extend>>
Make old patient appt.
Make new patient appt.
<<extend>>
<<include>>
Update patient information
Create new patient
Patient
Produce schedule information
Management
New Patient
<<include>>
Record availability
Doctor
<<include>>
Manage schedule
Old Patient

Symbols

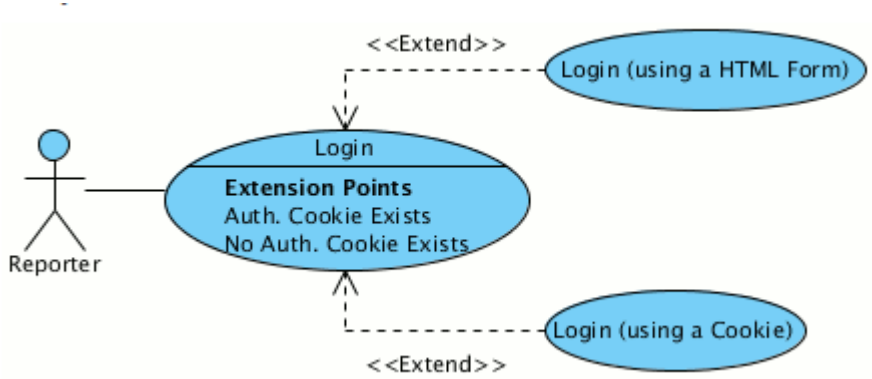
Generalization



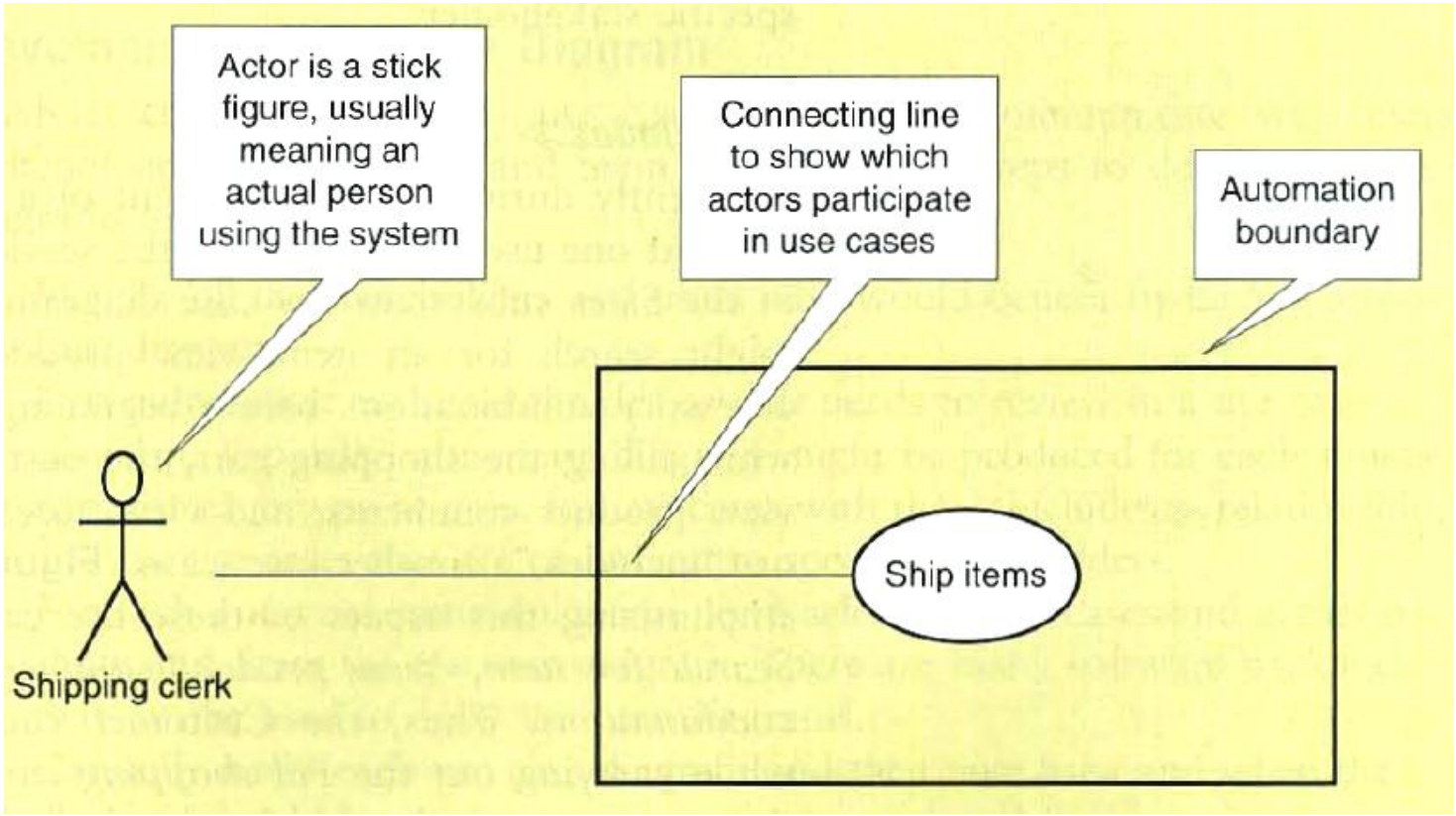
<<include>> relationship



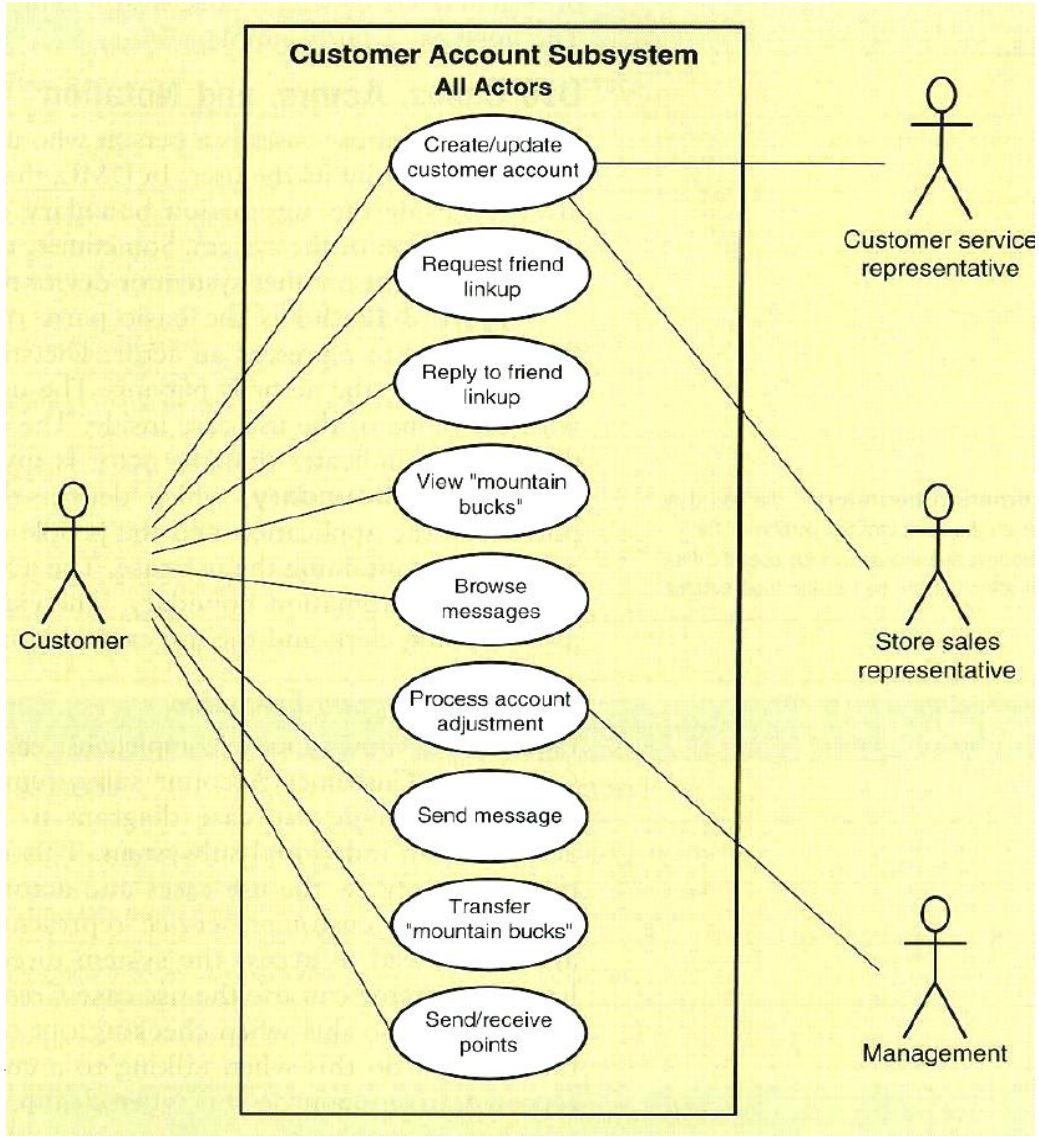
<<extend>> relationship



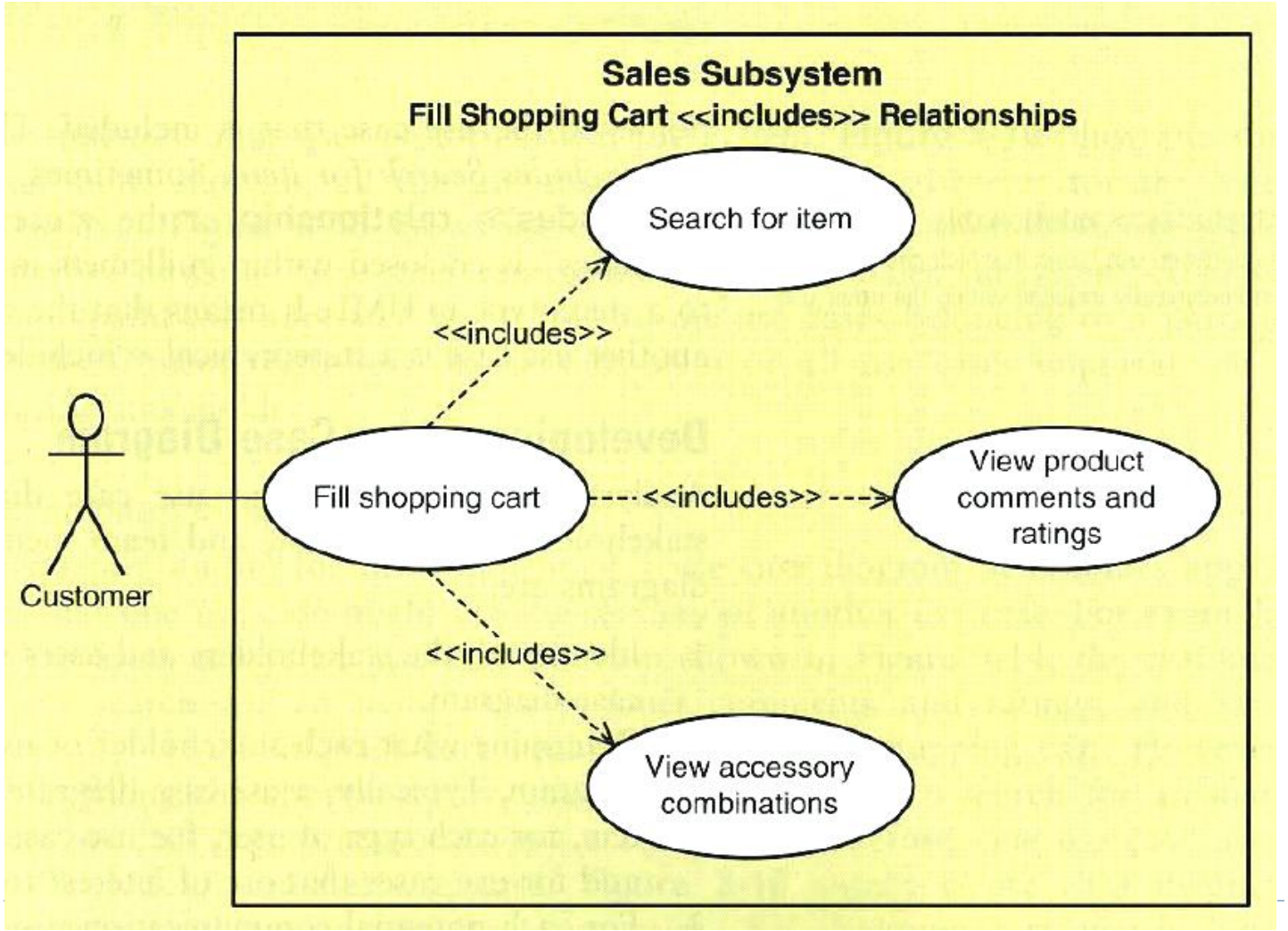
A simple use case with an actor

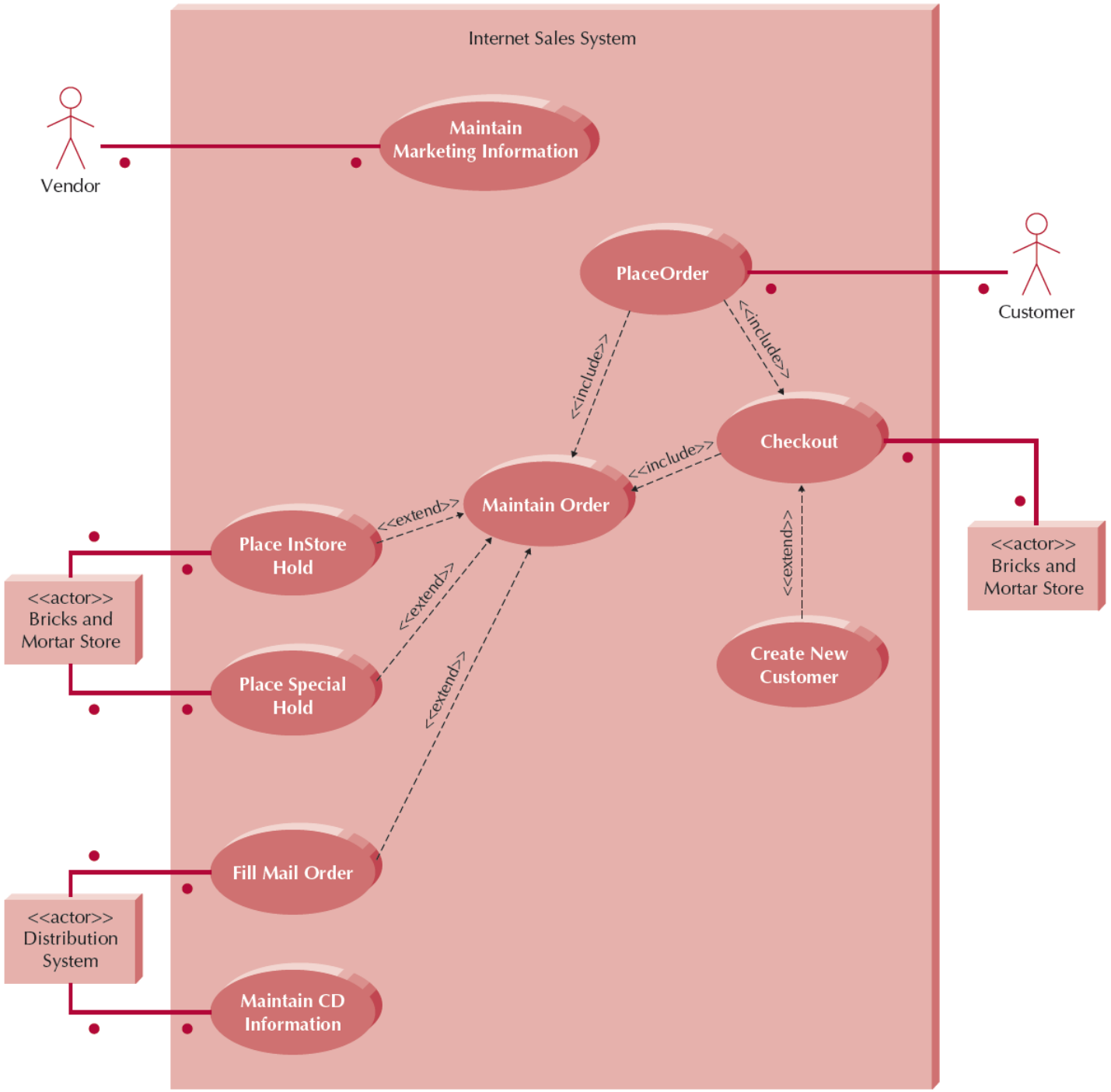


Account subsystem for RMO

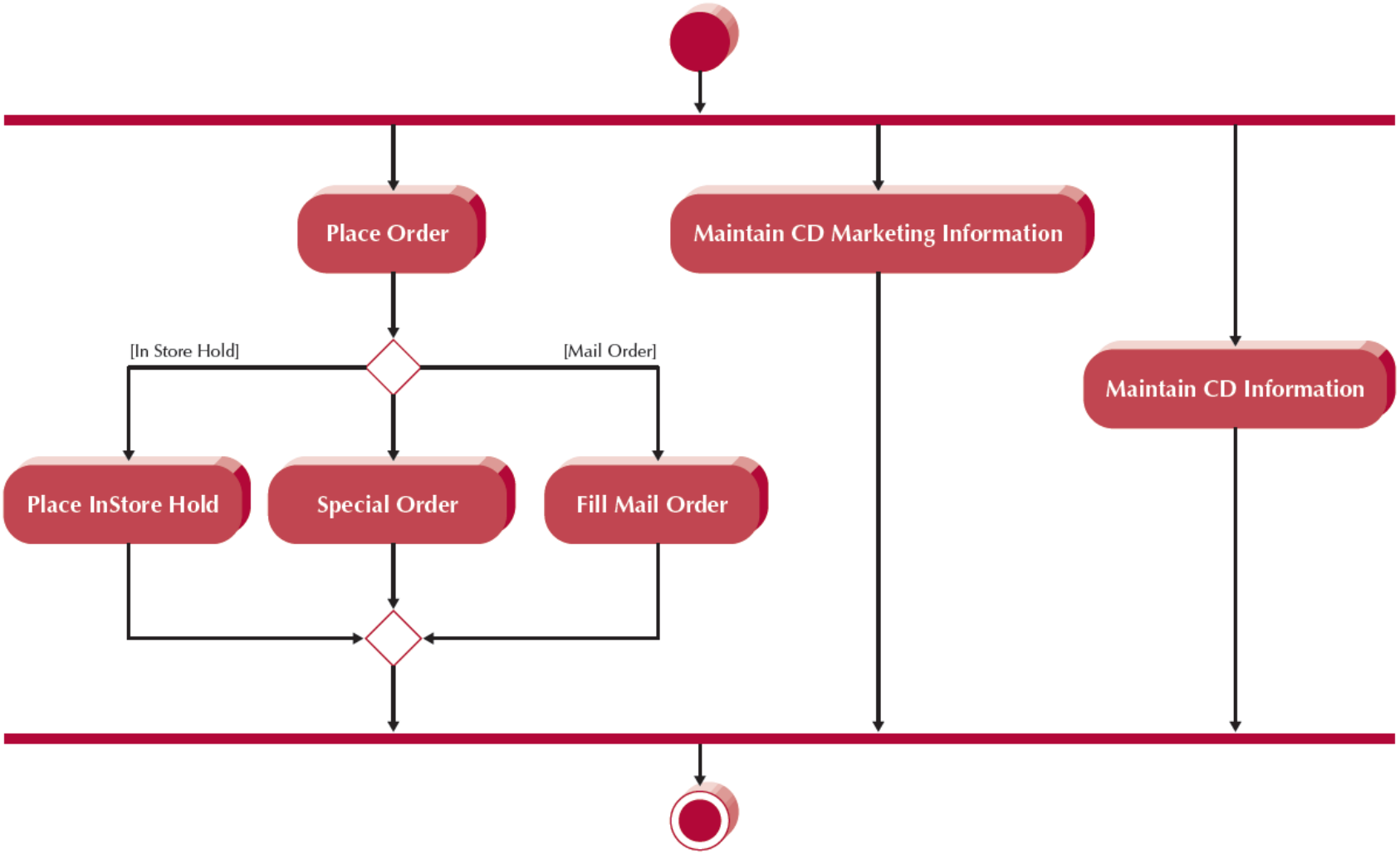


A use case diagram of the Fill shopping

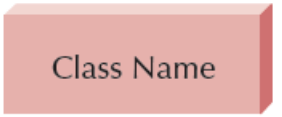



Internet Sales System
Vendor
Maintain Marketing Information
PlaceOrder
Customer
<<include>>
<<include>>
Checkout
<<include>>
Maintain Order
<<extend>>
Place InStore Hold
<<actor>> Bricks and Mortar Store
<<extend>>
<<extend>>
Place Special Hold
<<extend>>
Create New Customer
<<actor>> Bricks and Mortar Store
Fill Mail Order
<<actor>> Distribution System
Maintain CD Information

Activity diagram for the CD selections in Internet Sales System



Syntax for an Activity diagram (1)

An Action: ■ Is a simple, non-decomposable piece of behavior ■ Is labeled by its name	
An Activity: ■ Is used to represent a set of actions ■ Is labeled by its name	
An Object Node: ■ Is used to represent an object that is connected to a set of Object Flows ■ Is labeled by its class name	
A Control Flow: ■ Shows the sequence of execution	
An Object Flow: ■ Shows the flow of an object from one activity (or action) to another activity (or action)	
An Initial Node: ■ Portrays the beginning of a set of actions or activities	
A Final-Activity Node: ■ Is used to stop all control flows and object flows in an activity (or action)	

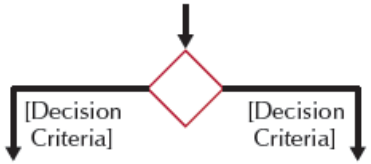
Syntax for an Activity diagram (1)

A Final-Flow Node:

- Is used to stop a specific control flow or object flow

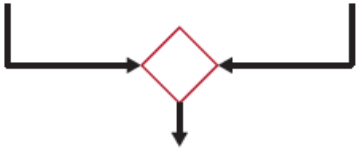
A Decision Node:

- Is used to represent a test condition to ensure that the control flow or object flow only goes down one path
- Is labeled with the decision criteria to continue down the specific path



A Merge Node:

- Is used to bring back together different decision paths that were created using a decision-node



A Fork Node:

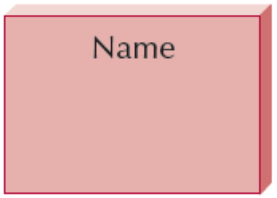
- Is used to split behavior into a set of parallel or concurrent flows of activities (or actions)

A Join Node:

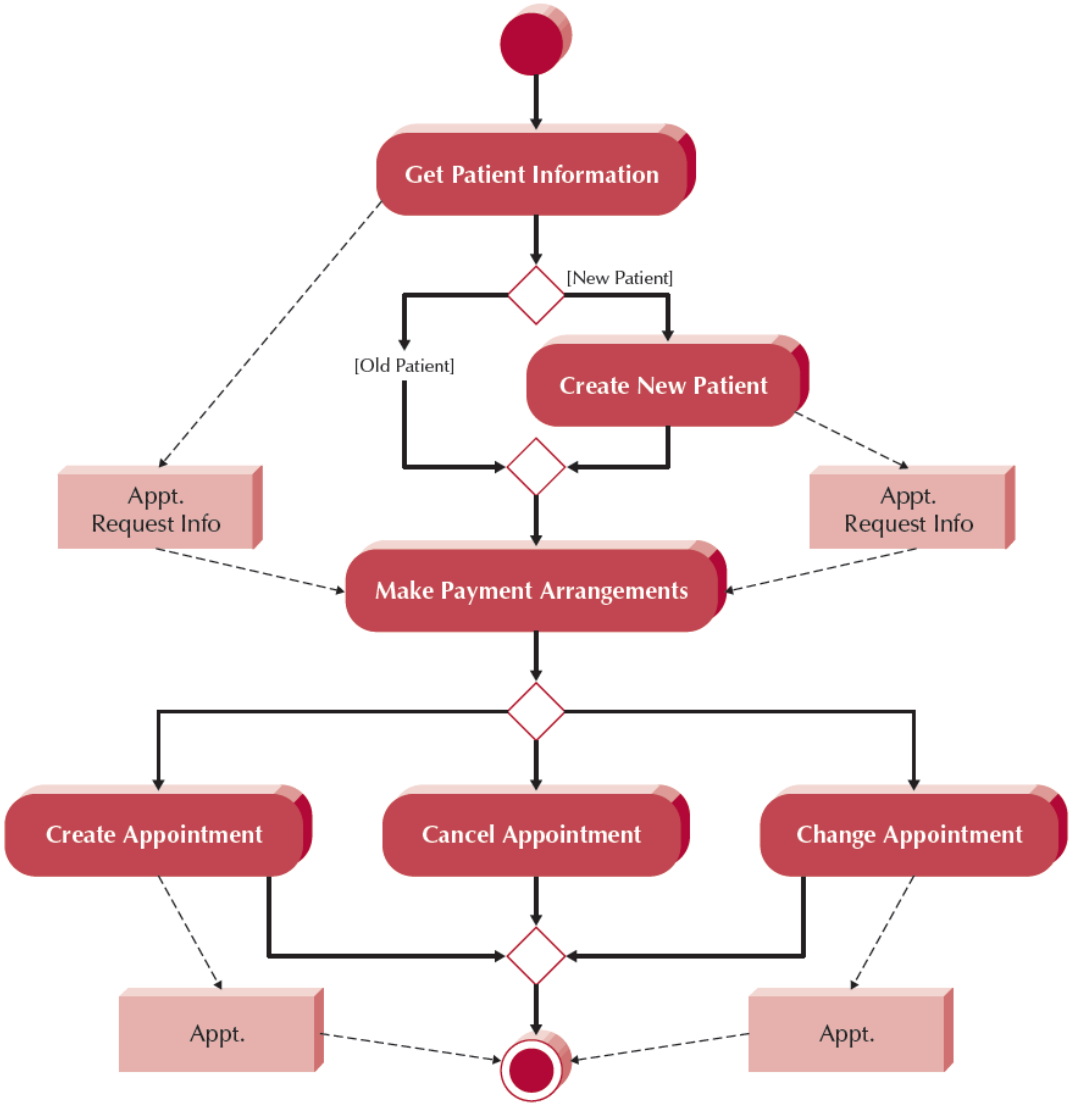
- Is used to bring back together a set of parallel or concurrent flows of activities (or actions)

A Swimlane:

- Is used to break up an activity diagram into rows and columns to assign the individual activities (or actions) to the individuals or objects that are responsible for executing the activity (or action)
- Is labeled with the name of the individual or object responsible

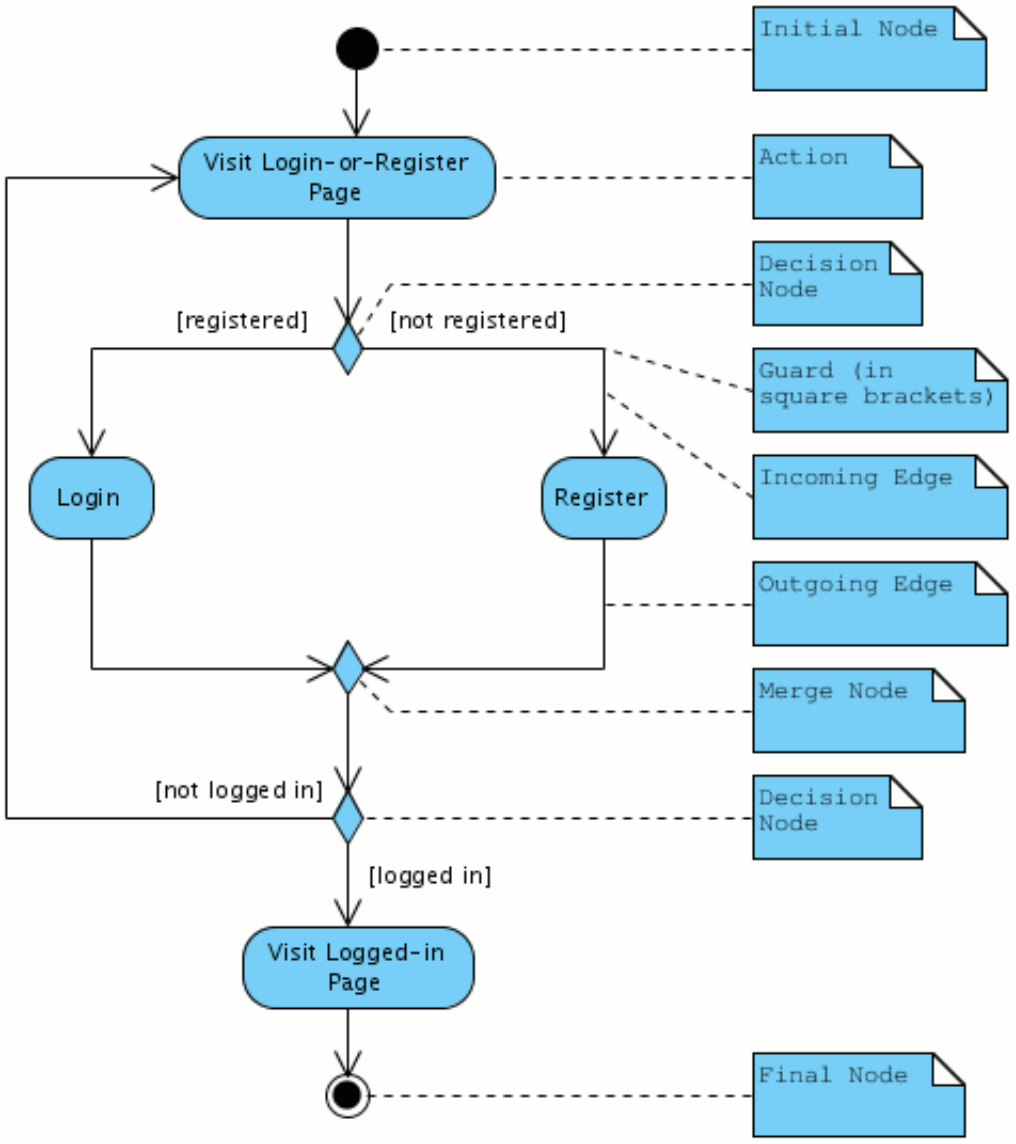


Activity diagram for appointment system

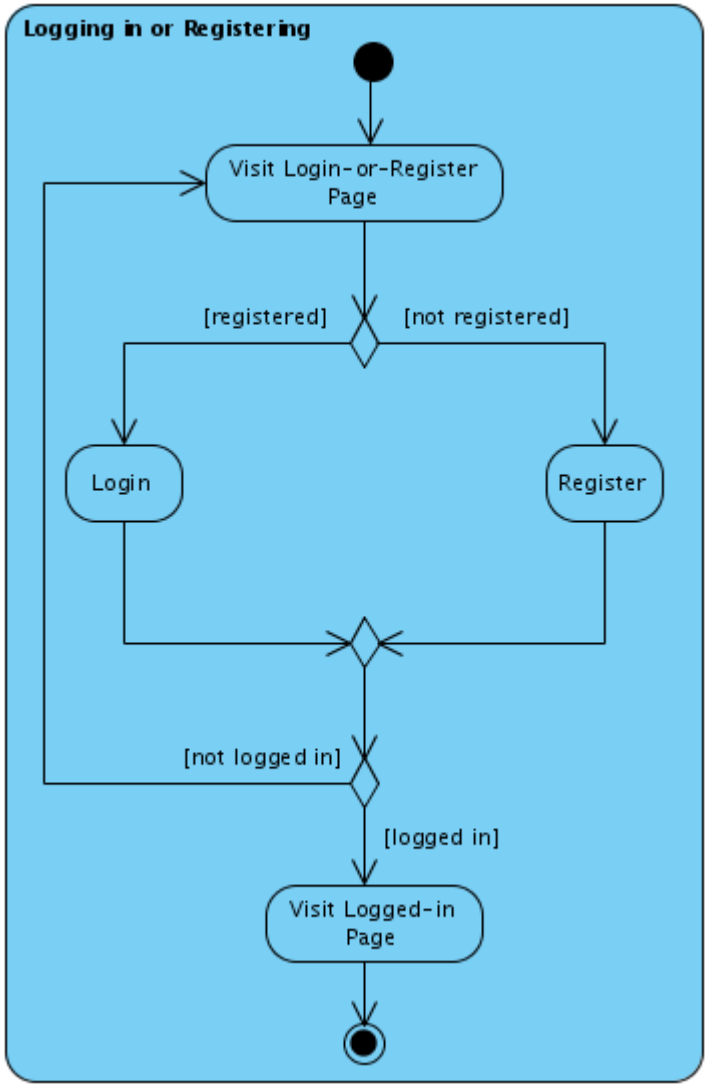


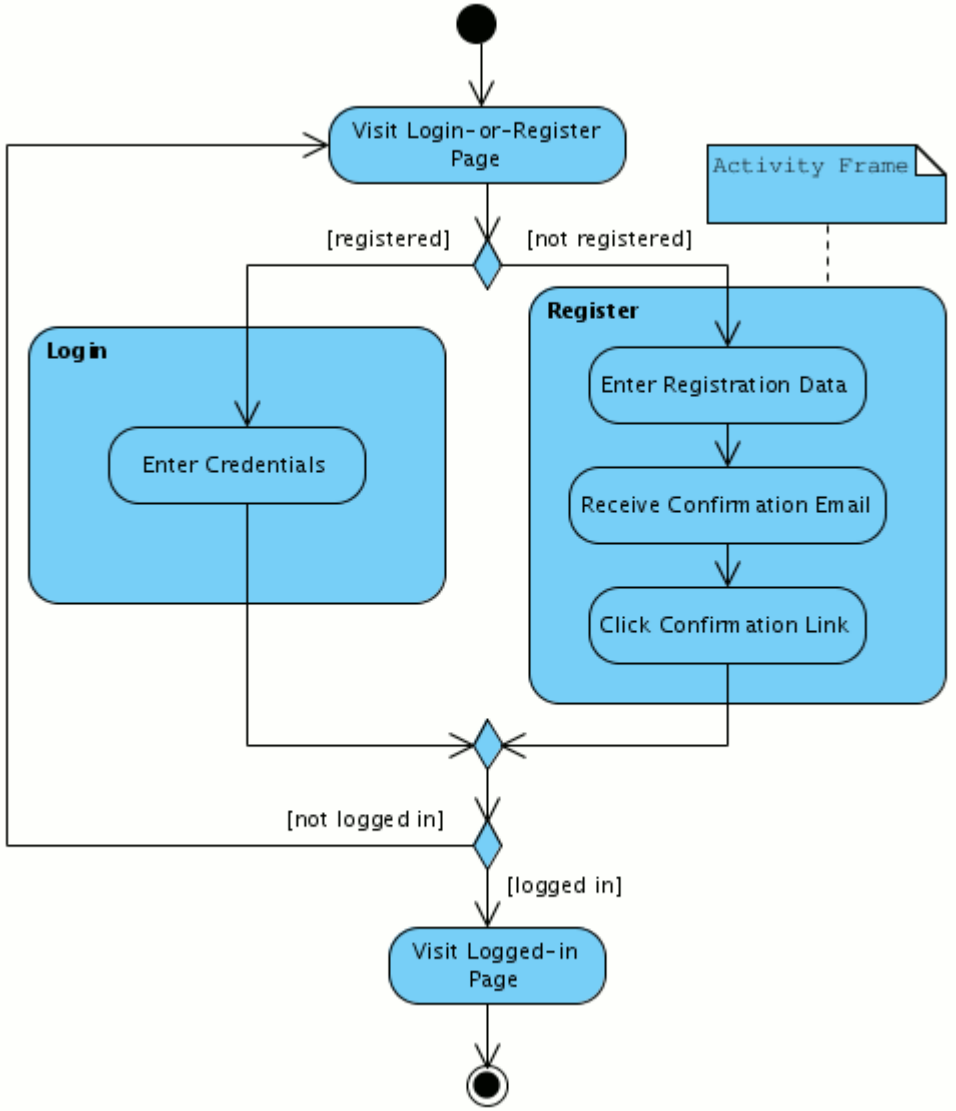
[2] P167

Introduction Activity Diagram

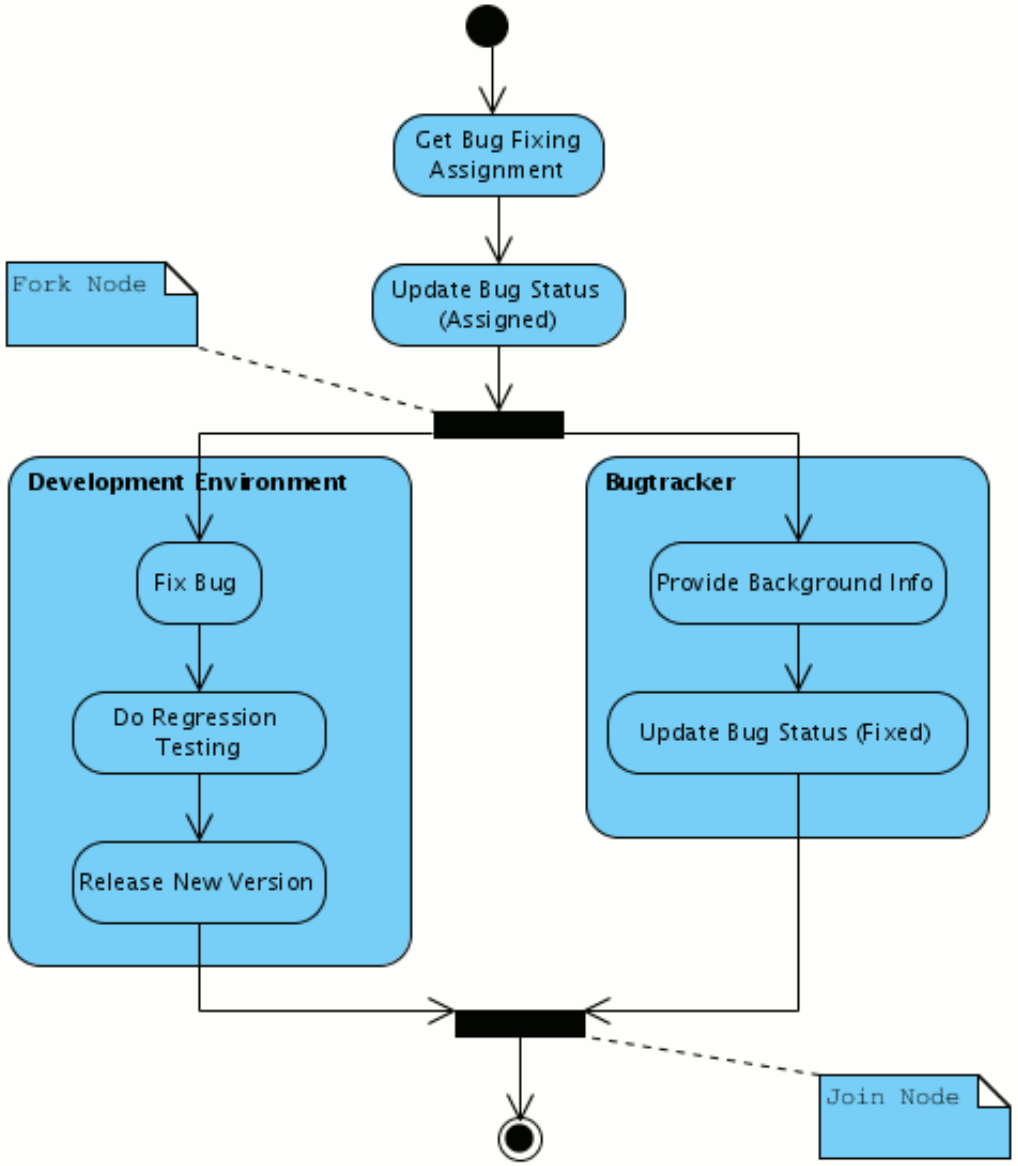


Activity diagram: Action and Activity

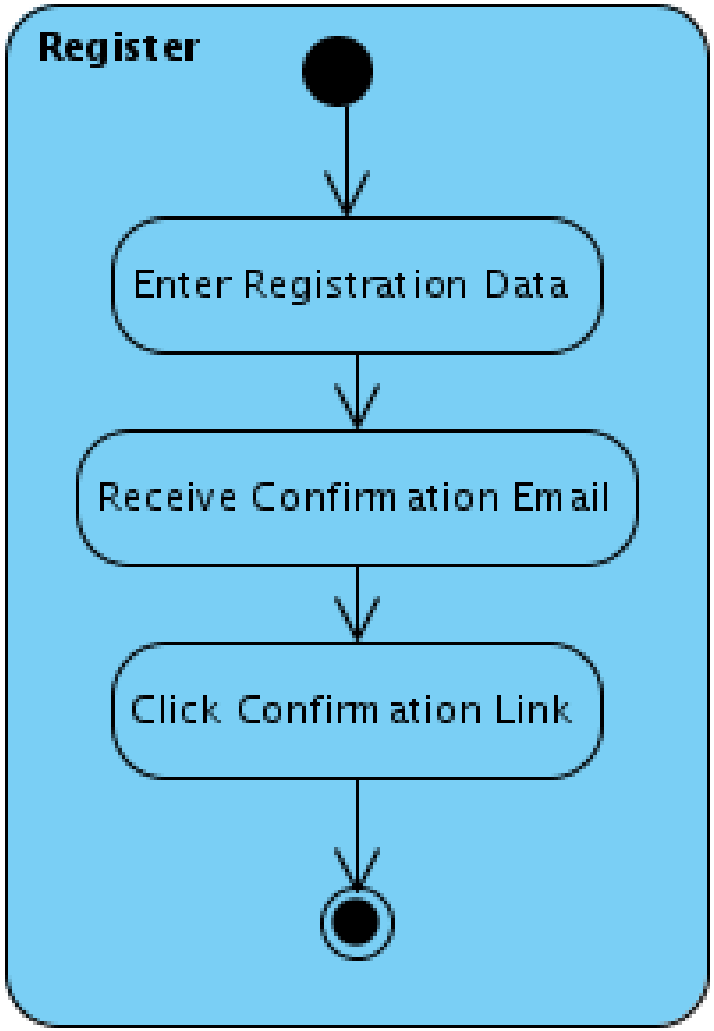


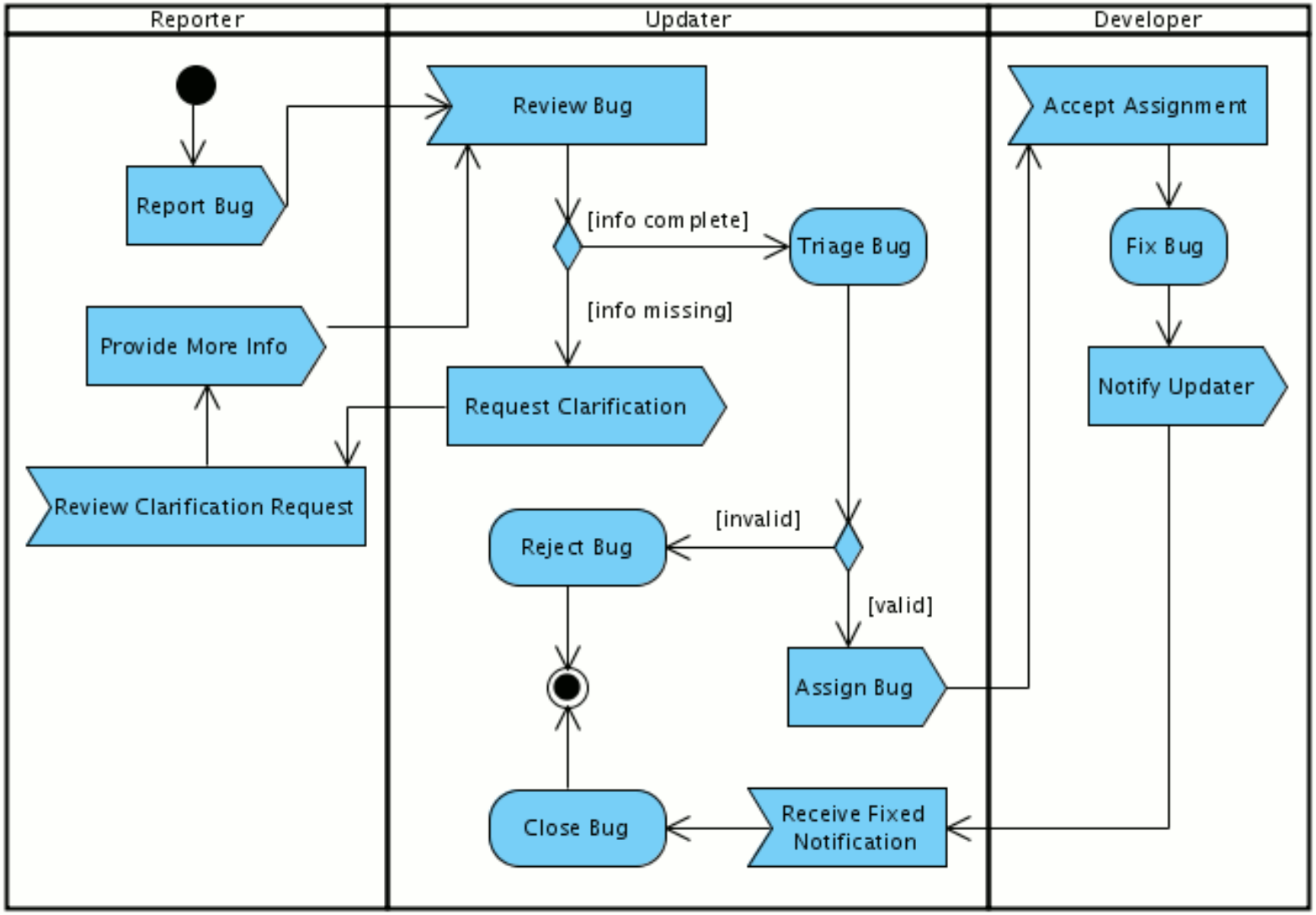


Parallel activity: Fork/Join Node



Calling External Activity




Reporter
Updater
Developer
Report Bug
Review Bug
Accept Assignment
[info complete]
Triage Bug
Fix Bug
Provide More Info
[info missing]
Notify Updater
Request Clarification
Review Clarification Request
[invalid]
Reject Bug
[valid]
Assign Bug
Close Bug
Receive Fixed Notification

Writing effective
Activity diagram and Use case

Get / Understand Requirement

Models to understand Req.

- Activity diagram
- Use case
- …

UML Tools
http://www.visual-paradigm.com