THE UNIVERSITY OF NEWCASTLE AUSTRALIA

FACULTY OF ENGINEERING AND BUILT ENVIRONMENT

www.newcastle.edu.au

SENG2130 – Week 3

Dr. Joe Ryan

SENG2130 – Systems Analysis and Design

University of Newcastle

Outline for today

- Requirements elicitation challenges
- Activity Diagram

Requirements Elicitation

First step in identifying the Requirements :System identification

- Two questions need to be answered:
  1. How can we identify the purpose of a system?
     - What are the requirements, what are the constraints?
  2. What is inside, what is outside the system?
- These two questions are answered during requirements elicitation and analysis
- Requirements elicitation:
  - Definition of the system in terms understood by the customer and/or user ("Requirements specification")
- Requirement Analysis:
  - Definition of the system in terms understood by the developer (Technical specification, "Analysis model")
- Requirements Process: Consists of the activities Requirements Elicitation and Analysis.

Techniques to elicit Requirements

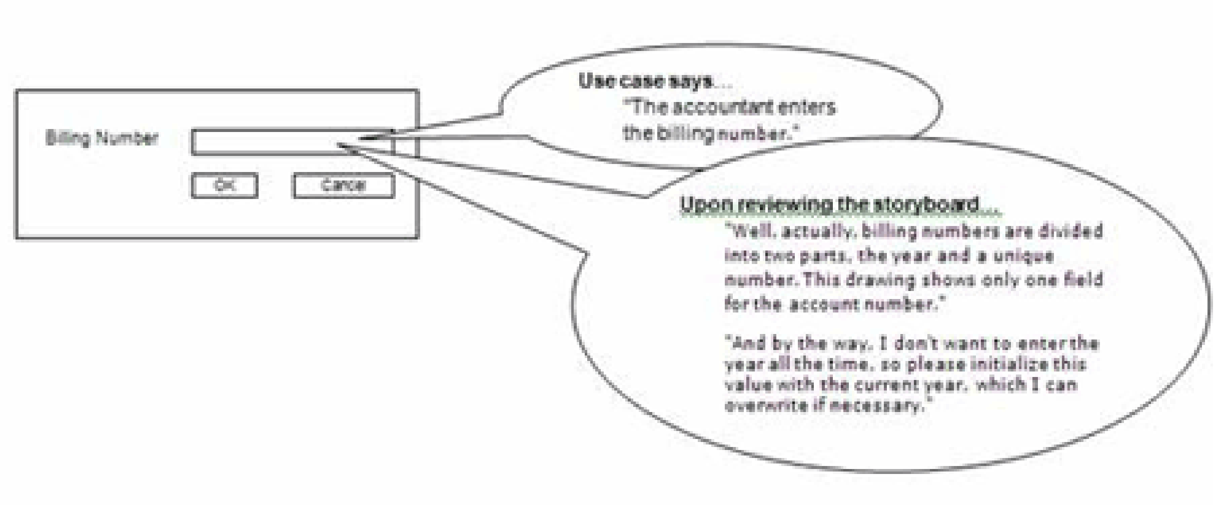
- Bridging the gap between end user and developer:
  - **Questionnaires:** Asking the end user a list of pre-selected questions
  - **Task Analysis:** Observing end users in their operational environment
  - **Scenarios:** Describe the use of the system as a series of interactions between a specific end user and the system
  - **Use cases:** Abstractions that describe a class of scenarios.

Questionnaires / Task Analysis

- Ask yourself or the client the following questions:
  - What are the primary tasks that the system needs to perform?
  - What data will the actor create, store, change, remove or add in the system?
  - What external changes does the system need to know about?
  - What changes or events will the actor of the system need to be informed about?
- However, don't rely on questions *and* questionnaires alone
- Insist on task observation if the system already exists (interface engineering or reengineering)
  - Ask to speak to the end user, not just to the client
  - Expect resistance and try to overcome it.

Scenarios

- A synthetic description of an event or series of actions and events
- A textual description of the usage of a system. The description is written from an end user’s point of view
- A scenario can include text, video, pictures and story boards. It usually also contains details about the work place, social situations and resource constraints.



Use case says…
"The accountant enters the billing number."

Upon reviewing the storyboard…
"Well, actually, billing numbers are divided into two parts, the year and a unique number. This drawing shows only one field for the account number."

"And by the way, I don't want to enter the year all the time, so please initialize this value with the current year, which I can overwrite if necessary."

Scenario example: Warehouse on fire

- Bob, driving down main street in his patrol car notices smoke coming out of a warehouse. His partner, Alice, reports the emergency from her car.
- Alice enters the address of the building into her wearable computer , a brief description of its location (i.e., north west corner), and an emergency level.
- She confirms her input and waits for an acknowledgment;
- John, the dispatcher, is alerted to the emergency by a beep of his workstation. He reviews the information submitted by Alice and acknowledges the report. He allocates a fire unit and sends the estimated arrival time (ETA) to Alice.
- Alice received the acknowledgment and the ETA..

Observations about the Warehouse on Fire Scenario

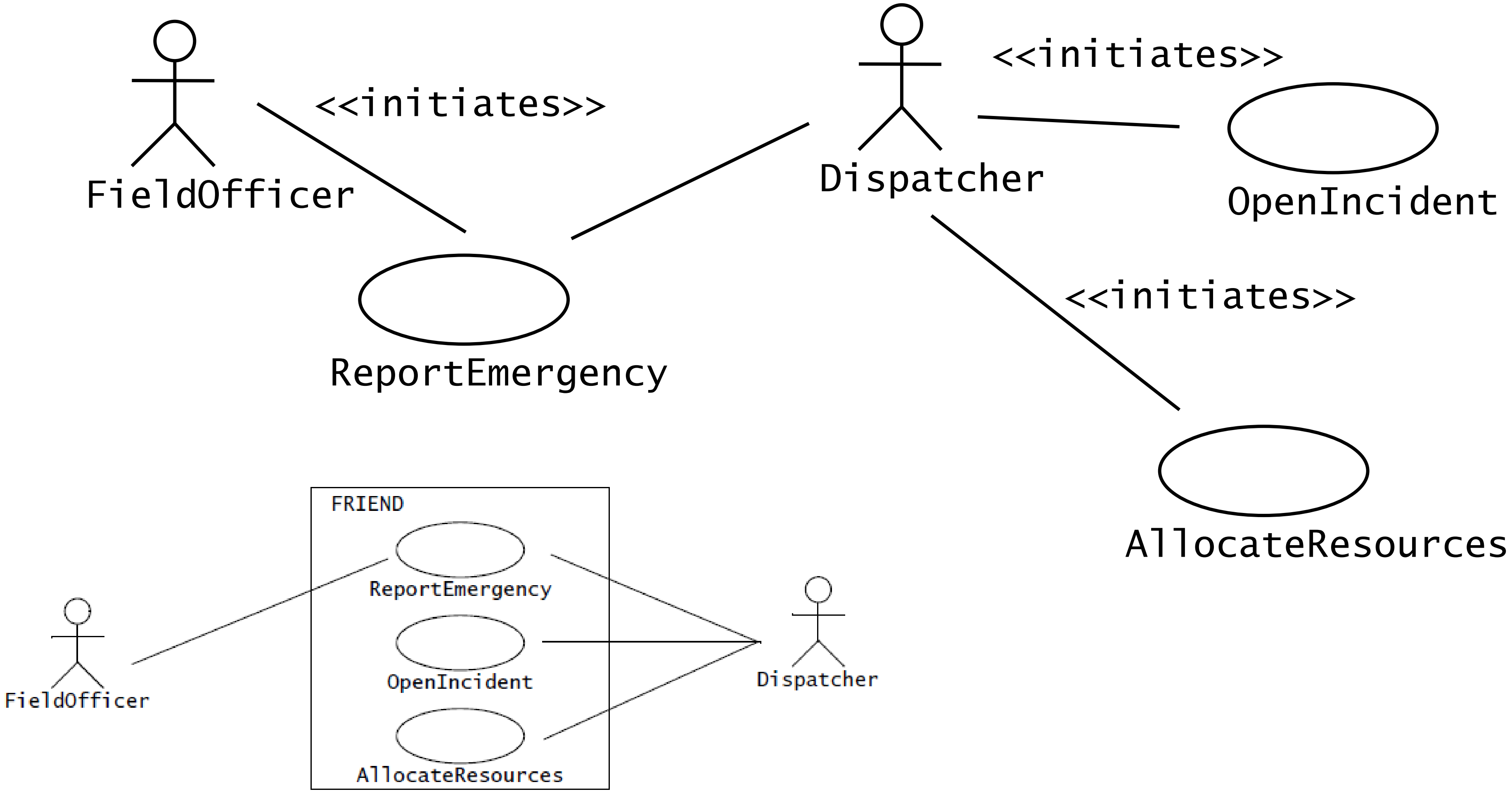
- It is a concrete scenario
  - – It describes a single instance of reporting a fire incident
  - – It does not describe all possible situations in which a fire can be reported

- Participating actors
  - – Bob, Alice and John.

After the scenarios are formulated

- Find all the use cases in the scenario that specify all instances of how to report a fire and model them in a use case model
  - Example: “Report Emergency” in the first paragraph of the scenario is a candidate for a use case
- Then add more detail to each of these use cases by describing:
  1. Name of the use case
  2. Participating actors
  3. Describe the entry condition
  4. Describe the flow of events
  5. Describe the exit condition
  6. Describe exceptions
  7. Describe quality requirements (nonfunctional requirements).

Use Case Model for Incident Management



How to find Use Cases

- Select a narrow vertical slice of the system (i.e. one scenario)
  - Discuss it in detail with the user to understand the user’s preferred style of interaction
- Select a horizontal slice (i.e. many scenarios) to define the scope of the system.
  - Discuss the scope with the user
- Use illustrative prototypes (mock-ups) as visual support
- Find out what the user does
  - Questionnaires
  - Task observation

Use Case Example: ReportEmergency

1. Use case name: ReportEmergency

2. Participating Actors:

   Field Officer, Dispatcher

3. Entry Condition:

   The FieldOfficer is logged into the FRIEND System

4. Flow of Events: **on next slide**

5. Exit Condition:

   The FieldOfficer has received an acknowledgement and the selected response OR The FieldOfficer has received an explanation indicating why the transaction could not be processed

6. Exceptions:

   – The FieldOfficer is notified immediately if the connection between terminal and central is lost

7. Quality Requirements:

   – The FieldOfficer's report is acknowledged within 30 seconds.

THE UNIVERSITY OF NEWCASTLE AUSTRALIA

Use Case Example: ReportEmergency (ctd)

4. Flow of Events:

1. The **FieldOfficer** activates the “Report Emergency” function of her terminal. The system responds by presenting a form to the officer.
2. The FieldOfficer fills the form, by selecting the emergency level, type, location, and brief description of the situation. The FieldOfficer also describes a response to the emergency situation. Once the form is completed, the FieldOfficer submits the form, and the **Dispatcher** is notified.
3. The Dispatcher creates an Incident in the database by invoking the OpenIncident use case. He selects a response and acknowledges the report.
4. The FieldOfficer receives the acknowledgment and the selected response.

Another Example: Allocate a Resource

- Glossary Entries:
  - ***Field Supervisor:*** This is the official at the emergency site
  - ***Resource Allocator:*** The Resource Allocator is responsible for the commitment and release of Resources managed by the FRIEND system
  - ***Dispatcher:*** A Dispatcher enters, updates, and removes Emergency Incidents, Actions, and Requests in the system. The Dispatcher also closes Emergency Incidents
  - ***Field Officer:*** Reports accidents from the Field

Allocate a Resource (cont’d)

1. Use case name: AllocateResources
2. Participating Actors:

   Field Officer, Dispatcher, Resource Allocator, Field Supervisor
3. Entry Condition:

   The Resource Allocator has selected an available resource
4. Flow of Events:

   1. The Resource Allocator selects an Emergency Incident
   2. The Resource is committed to the Emergency Incident
5. Exit Condition:

   The use case terminates when the resource is committed

   The selected Resource is unavailable to other Requests.
6. Special Requirements:

   The Field Supervisor is responsible for managing Resources.

Order of steps when formulating use cases

- First step: Name the use case
  - Use case name: ReportEmergency
- Second step: Find the actors
  - Generalize the concrete names from the scenario to participating actors
  - Participating Actors:
    - Field Officer
    - Dispatcher
- Third step: Concentrate on the flow of events
  - Use informal natural language

Another Use Case Example

Flow of Events

- The Bank Customer specifies a Account and provides credentials to the Bank proving that he is authorized to access the Bank Account
- The Bank Customer specifies the amount of money he wishes to withdraw
- The Bank checks if the amount is consistent with the rules of the Bank and the state of the Bank Customer’s account. If that is the case, the Bank Customer receives the money in cash.

Use Case Attributes

Use Case Name **Withdraw Money Using ATM**

Participating Actor: Bank Customer

Entry condition:

- Bank Customer has opened a Bank Account with the Bank ***and***
  Bank Customer has received an ATM Card and PIN

Exit condition:

- Bank Customer has the requested cash ***or***
  Bank Customer receives an explanation from the ATM about why the cash could not be dispensed.

Flow of Events: A Request-Response Interaction between Actor and System

Actor steps	System steps
1.The Bank Customer inserts the card into the ATM	
	2.The ATM requests the input of a four-digit PIN
3. The Bank Customer types in PIN	
	4. If several accounts are recorded on the card, the ATM offers a choice of the account numbers for selection by the Bank Customer
5. The Bank Customer selects an account	
	6.If only one account is recorded on the card or after the selection, the ATM requests the amount to be withdrawn
7. The Bank Customer inputs an amount	
	8.The ATM outputs the money and a receipt and stops the interaction.

Use Case Exceptions

Actor steps

1. The Bank Customer inputs her card into the ATM.**[Invalid card]**

3. The Bank Customer types in PIN. **[Invalid PIN]**

5. The Bank Customer selects an account .

7. The Bank Customer inputs an amount. **[Amount over limit]**

[Invalid card]

The ATM outputs the card and stops the interaction.

[Invalid PIN]

The ATM announces the failure and offers a 2nd try as well as canceling the whole use case. After 3 failures, it announces the possible retention of the card. After the 4th failure it keeps the card and stops the interaction.

[Amount over limit]

The ATM announces the failure and the available limit and offers a second try as well as canceling the whole use case.

Guidelines for Formulation of Use Cases (1)

- Name
  - Use a verb phrase to name the use case
  - The name should indicate what the user is trying to accomplish
  - Examples:
    - “Request Meeting”, “Schedule Meeting”, “Propose Alternate Date”
- Length
  - A use case description should not exceed 1-2 pages. If longer, use include relationships
  - A use case should describe a complete set of interactions.

Guidelines for Formulation of Use Cases (2)

Flow of events:

- Use the active voice. Steps should start either with “The Actor” or “The System …”
- The causal relationship between the steps should be clear
- All flow of events should be described (not only the main flow of event)
- The boundaries of the system should be clear. Components external to the system should be described as such
- Define important terms in the glossary.

Event Flow: Use Indentation to show the Interaction between Actor and System

1.The Bank Customer inserts the card into the ATM

    2.The ATM requests the input of a four-digit PIN

3. The Bank Customer types in PIN

    4. If several accounts are recorded on the card, the ATM offers a choice of the account numbers for selection by the Bank Customer

5. The Bank Customer selects an account

    6.If only one account is recorded on the card or after the selection, the ATM requests the amount to be withdrawn

7. The Bank Customer inputs an amount

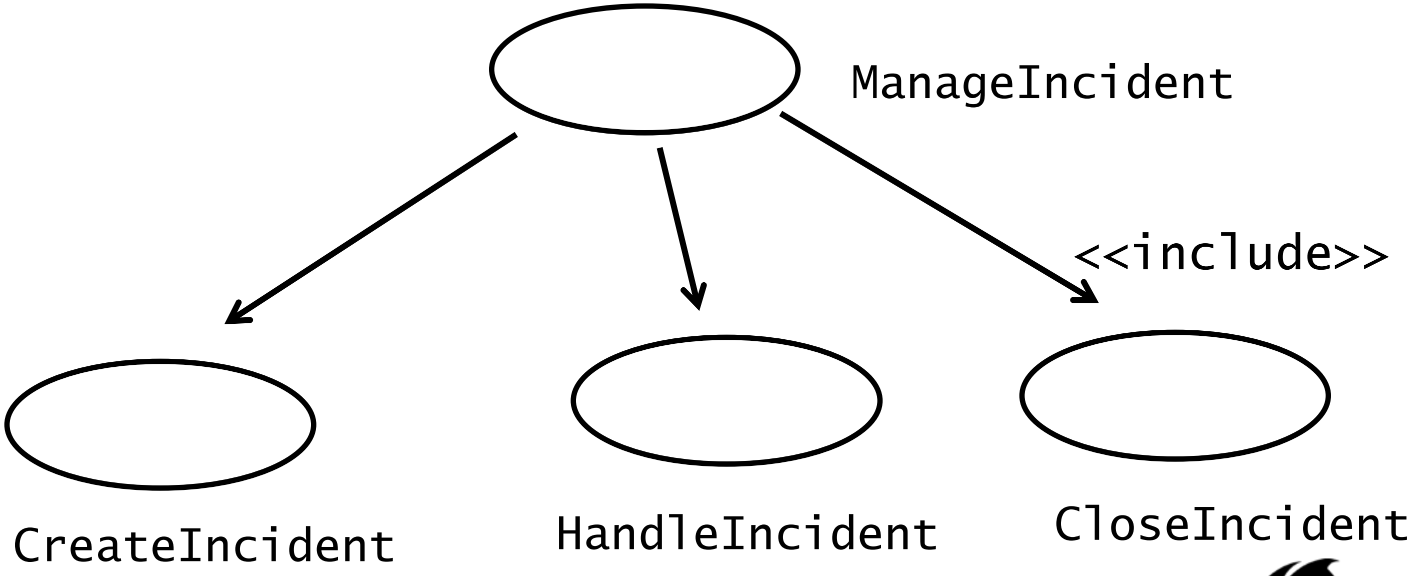
    8.The ATM outputs the money and a receipt and stops the interaction.

Use Case Associations

- Dependencies between use cases are represented with use case associations
- Associations are used to reduce complexity
  - Decompose a long use case into shorter ones
  - Separate alternate flows of events
  - Refine abstract use cases
- Types of use case associations
  - Includes
  - Extends
  - Generalization

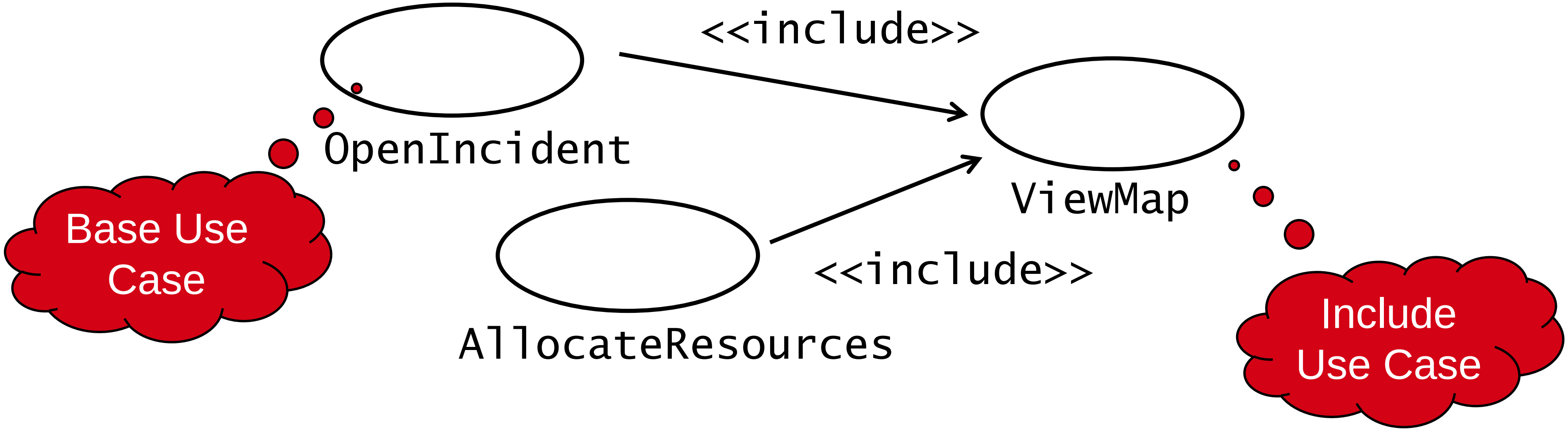
<<include>>: Functional Decomposition

- Problem:
  - A function in the original problem statement is too complex
- Solution:
  - Describe the function as the aggregation of a set of simpler functions. The associated use case is decomposed into shorter use cases



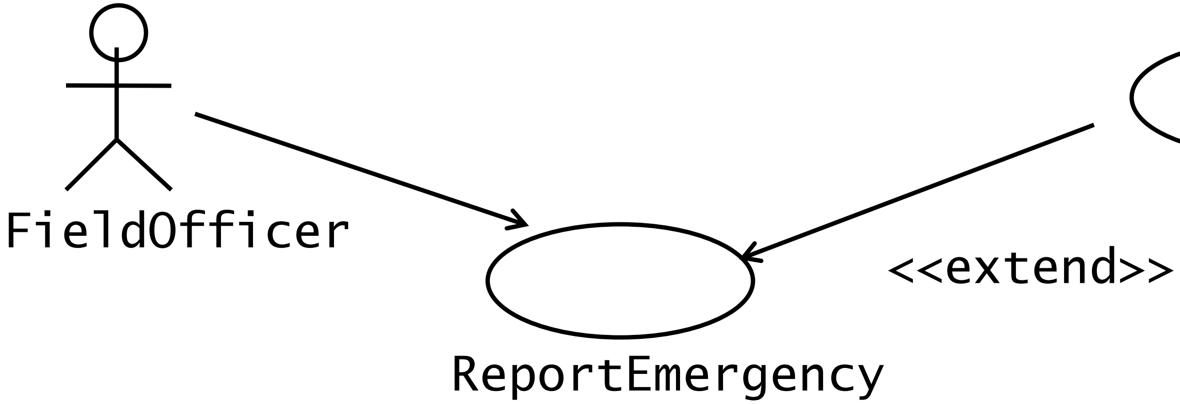
<<include>>: Reuse of Existing Functionality

- Problem: There are overlaps among use cases. How can we *reuse* flows of events instead of duplicating them?
- Solution: The *includes association* from use case A to use case B indicates that an instance of use case A performs all the behavior described in use case B ("A delegates to B")
- Example: Use case "ViewMap" describes behavior that can be used by use case "OpenIncident" ("ViewMap" is factored out)



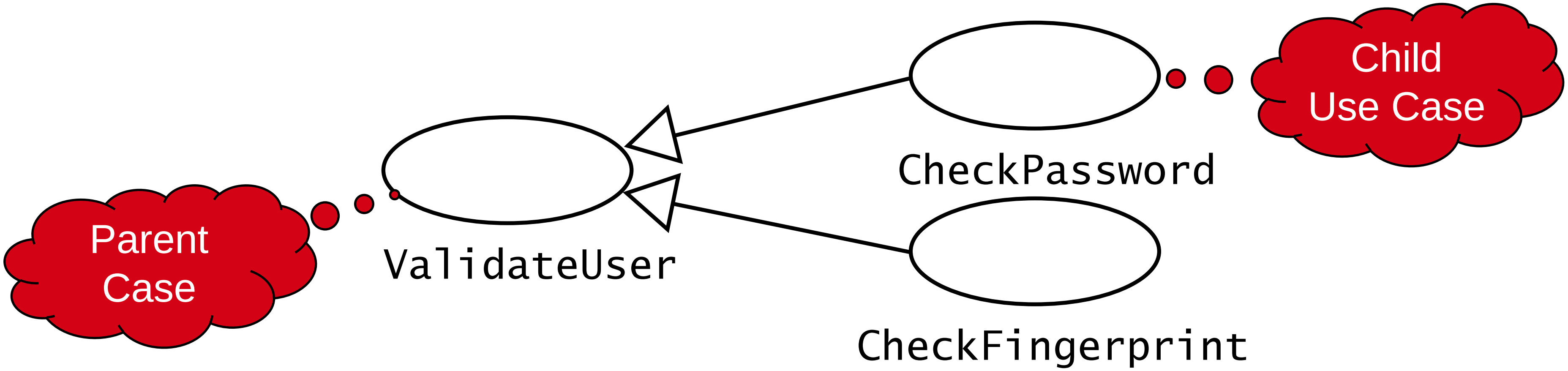
<<extend>> Association for Use Cases

- Problem: The functionality in the original problem statement needs to be extended.
- Solution: An *extend association* from use case A to use case B
- Example: “ReportEmergency” is complete by itself, but can be extended by use case “Help” for a scenario in which the user requires help



Generalization in Use Cases

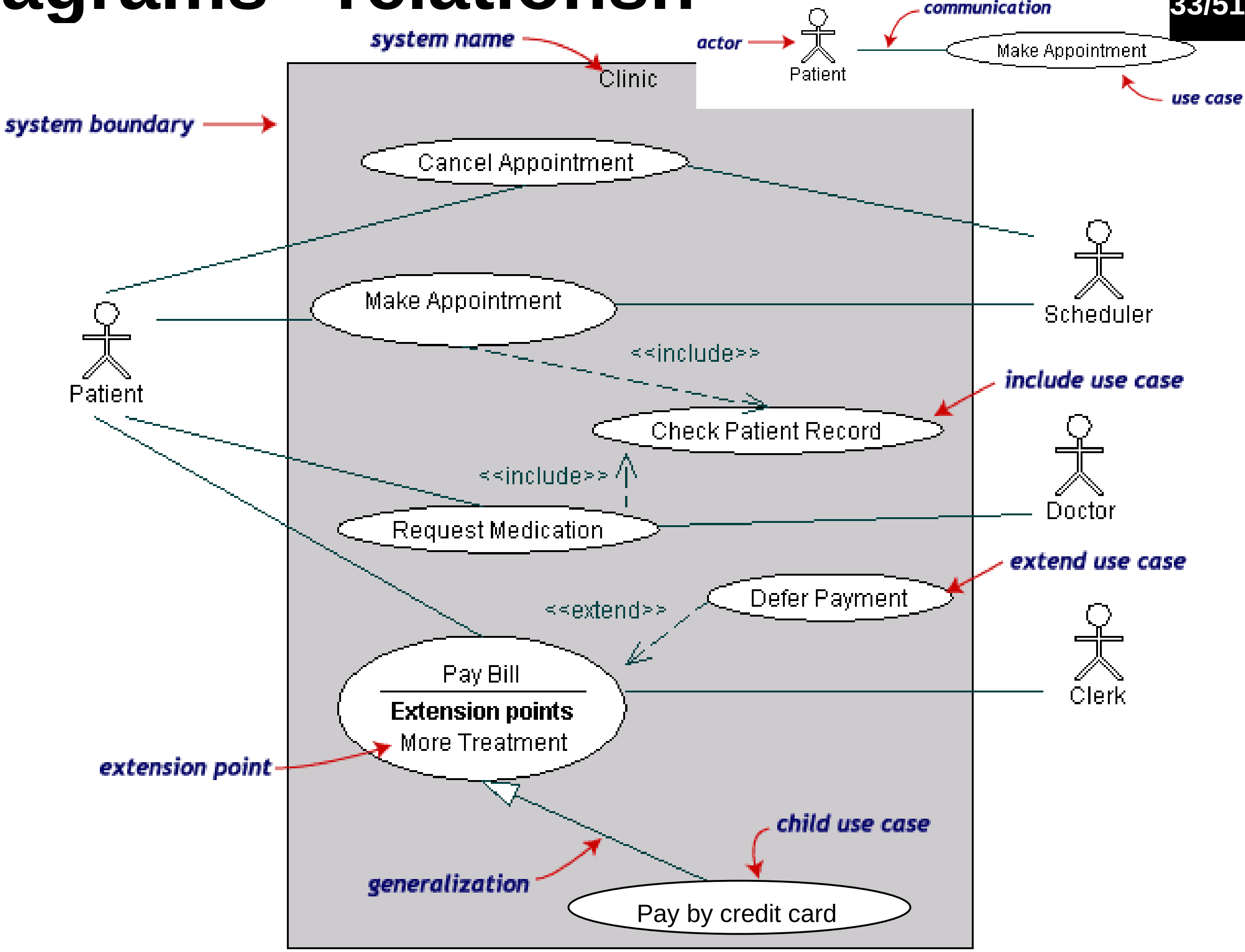
- Problem: We want to factor out common (but not identical) behavior.
- Solution: The child use cases inherit the behavior and meaning of the parent use case and add or override some behavior.
- Example: “ValidateUser” is responsible for verifying the identity of the user. The customer might require two realizations: “CheckPassword” and “CheckFingerprint”



review

Use case diagrams - relationshin

- Communication
  - Actors and use cases exchange information
- Include (or use)
  - A Use case calls the services of a common subroutine, where the common subroutine itself becomes an additional use case
- Extend
  - A use case can extend another by adding events. A typical application of extend relationships is the specification of exceptional behaviour
- Generalization
  - A use case specializes a more general one by adding more detail



Activity Diagram

review

Use case description – Fully developed description

Use Case Name	Buy a Beverage	
Brief Description	The vending machine delivers a beverage after a customer selects and pays for it	
Actors	customer	
Related Use cases	Include or extend	
Entry condition	the waiting state in which it displays the message “Enter coins”	
Exit condition	the item is dispensed.	
Flow of Events	Actors	System
	1. Customer inserts coins into the machine. 2. Customer pushes a button.	1.1 Displays the total value of money entered 1.2 Machine lights up the buttons for the items that can be purchased for the money inserted. 2.1 Dispenses the corresponding item and make changes
Exception condition	2.1 If the customer presses a button for an item that costs more than the money inserted, a. the message “You must insert $nn.nn more for that item” is displayed, where nn.nn is the amount of additional money needed, and b. the machine continues to accept coins or a selection 2.2 If the customer has inserted enough money to buy the item but the machine cannot make the correct change a. the message “Cannot make correct change” is displayed, and b. the machine continues to accept coins or a selection. 2.3 If the customer presses the cancel button before an item has been selected, a. the customer’s money is returned and b. the machine resets to the waiting state.	

Use case description – Act

Use case name: Buy a beverage

Entry condition

the waiting state in which it displays the message “Enter coins”

Customer

1. Customer inserts coins into the machine.

System

1.1 Displays the total value of money entered

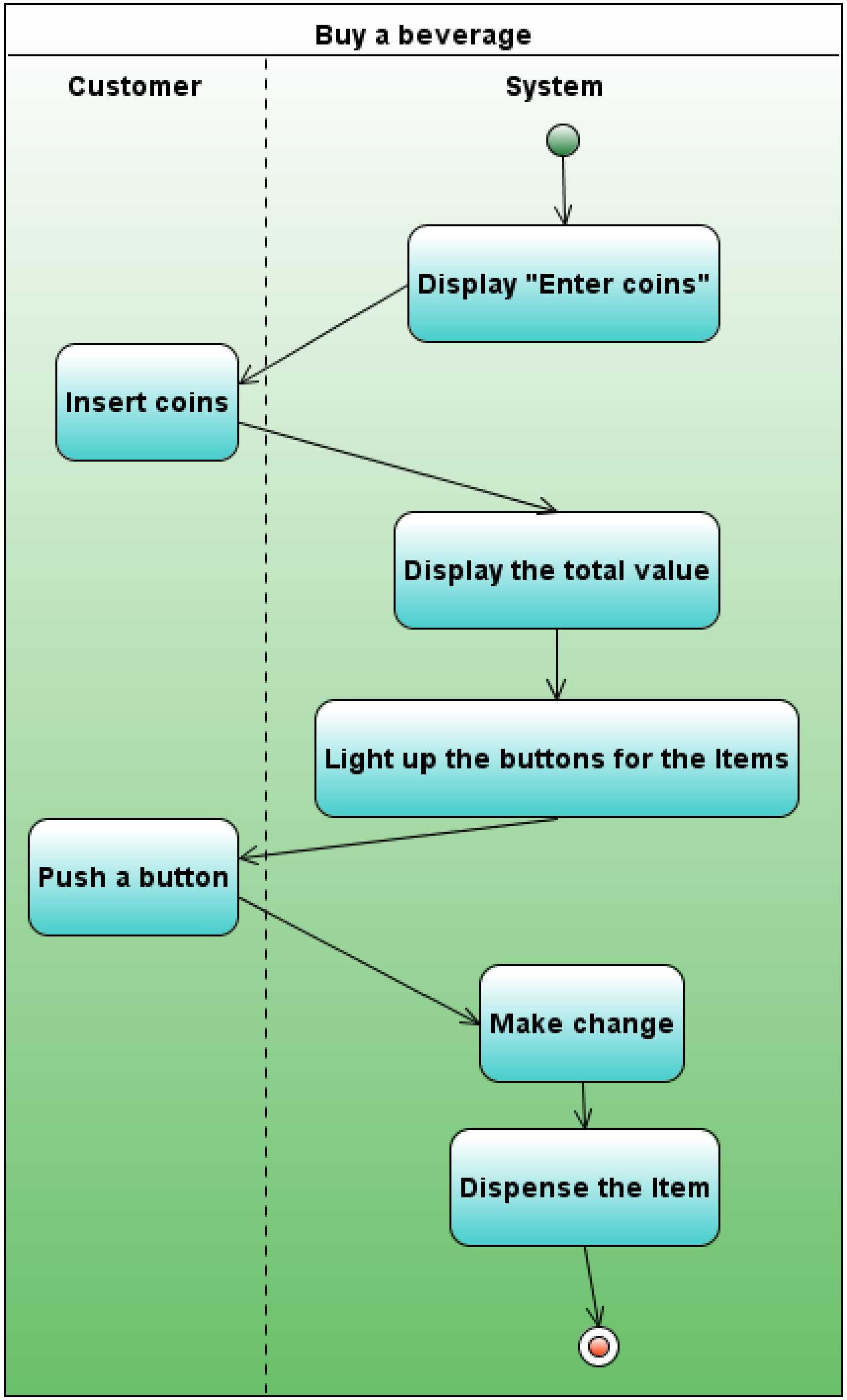
1.2 Machine lights up the buttons for the items that can be purchased for the money inserted

Customer

2. Customer pushes a button.

System

2.1 Dispenses the corresponding item and make changes



Use case description – Activity diagram

Exception condition

2.1 If the customer presses a button for an item that costs more than the money inserted,

a. the message “You must insert $*nn.nn* more for that item” is displayed, where *nn.nn* is the amount of additional money needed, and

b. the machine continues to accept coins or a selection

2.2 If the customer has inserted enough money to buy the item but the machine cannot make the correct change

a. the message “Cannot make correct change” is displayed, and

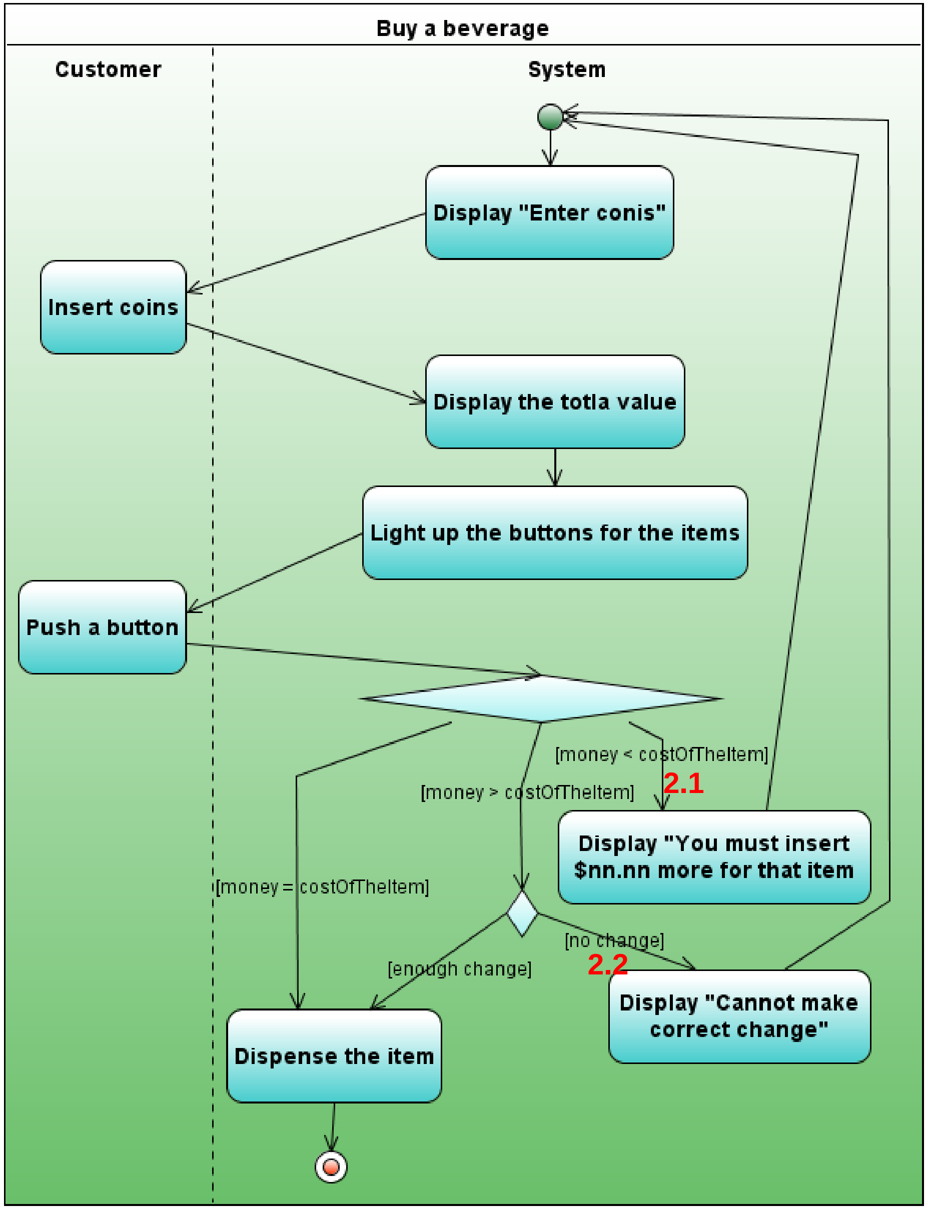
b. the machine continues to accept coins or a selection.

2.3 If the customer presses the cancel button before an item has been selected,

a. the customer’s money is returned and

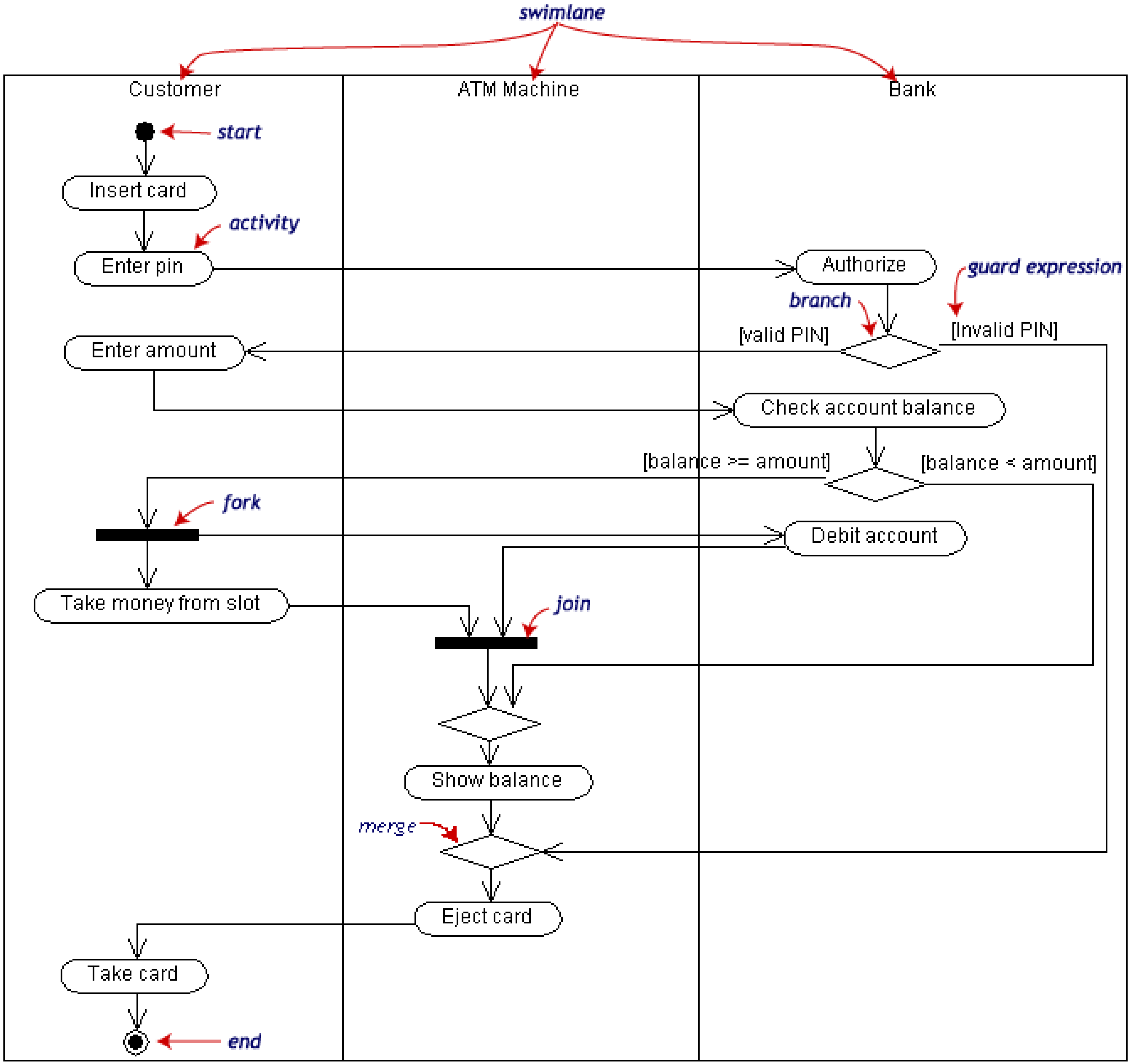
b. the machine resets to the waiting state.

DIY:Add this to the Diagram



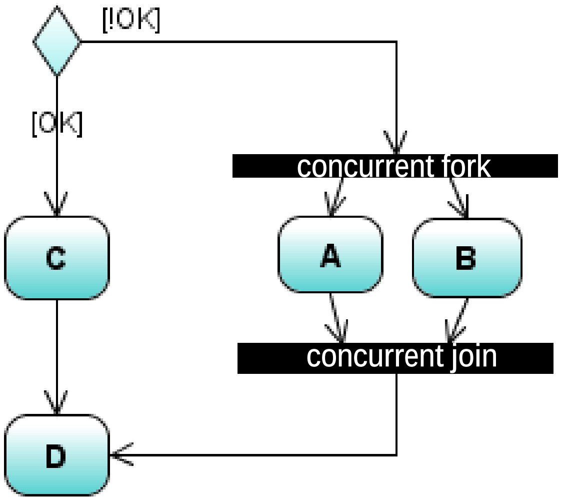
Activity diagrams

- Activity diagrams are used for business process modelling, for modelling the logic captured by a single use case
- It is essentially a flowchart
  - Showing flow of control from activity to activity
  - Records the dependencies between activities, such as which things can happen in parallel and what must be finished before something else can start


swimlane
Customer
ATM Machine
Bank
start
Insert card
activity
Enter pin
Authorize
guard expression
branch
[valid PIN]
[Invalid PIN]
Enter amount
Check account balance
[balance >= amount]
[balance < amount]
fork
Debit account
Take money from slot
join
Show balance
merge
Eject card
Take card
end

Activity diagrams

Which of the following pseudo code implements the activity diagram?



```
If (OK) then
    do C
else
    do A and B //either order
do D
```

```
do C
do A and B
do D
```

```
if (OK) then
    do C
else
    do A OR B //not both
do D
```

```
if (OK) then
    do C
else
    do A
    do B
do D
```

Activity diagram guidelines

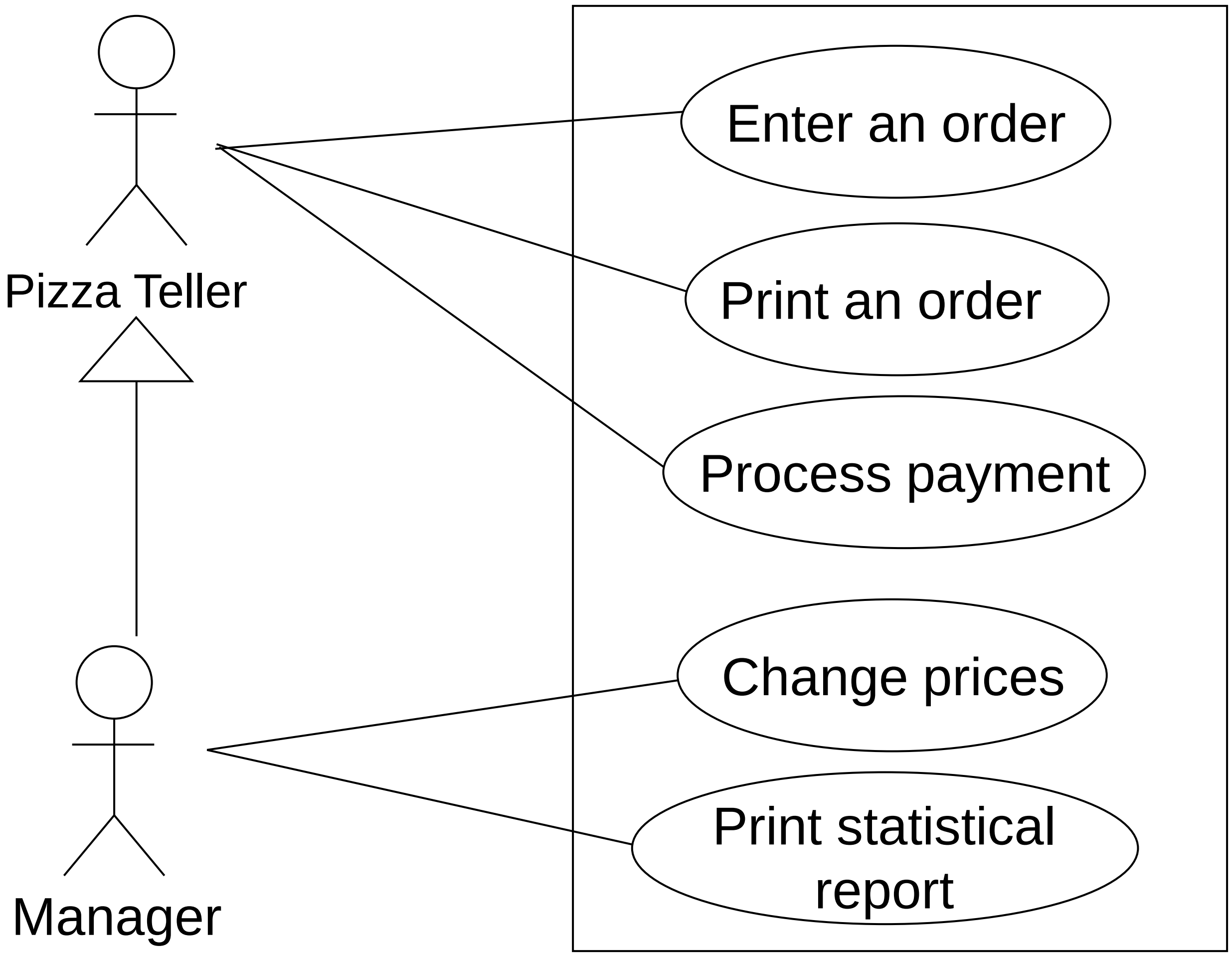
1. Identify activities (steps) of a process
2. Identify who/what performs activities (process steps)
3. Identify order of activities
4. Identify decision points (if-then)
5. Determine if step is parallel
6. Draw swimlanes
7. Draw the start point of the process in the swimlane of the first activity
8. Draw the oval of the first activity
9. Draw an arrow to the location of the second activity
10. Draw subsequent activities, while inserting decision points and synchronization/loop bars where appropriate
11. Draw the end point after the last activity.

The Pizza Shop

Consider a pizza shop. This store caters for many activities, the major one being “pizza eating”. The choice of pizzas is simple: Meat Lovers, Vegetarian or Supreme. The pizzas come in large, medium and small. The base is thick or thin. Customers come to the desk and talk to the Pizza Tellers to make a selection and the order is taken. This includes the customer's name so they can be identified with the order. The customer goes to a table and waits for the order to be made up. The customer is called back to the desk (by name) when the order is ready. The customer pays the bill. Every so often the manager of the store changes the prices for each type of pizza.

- Build the **use case diagram** for the order system.

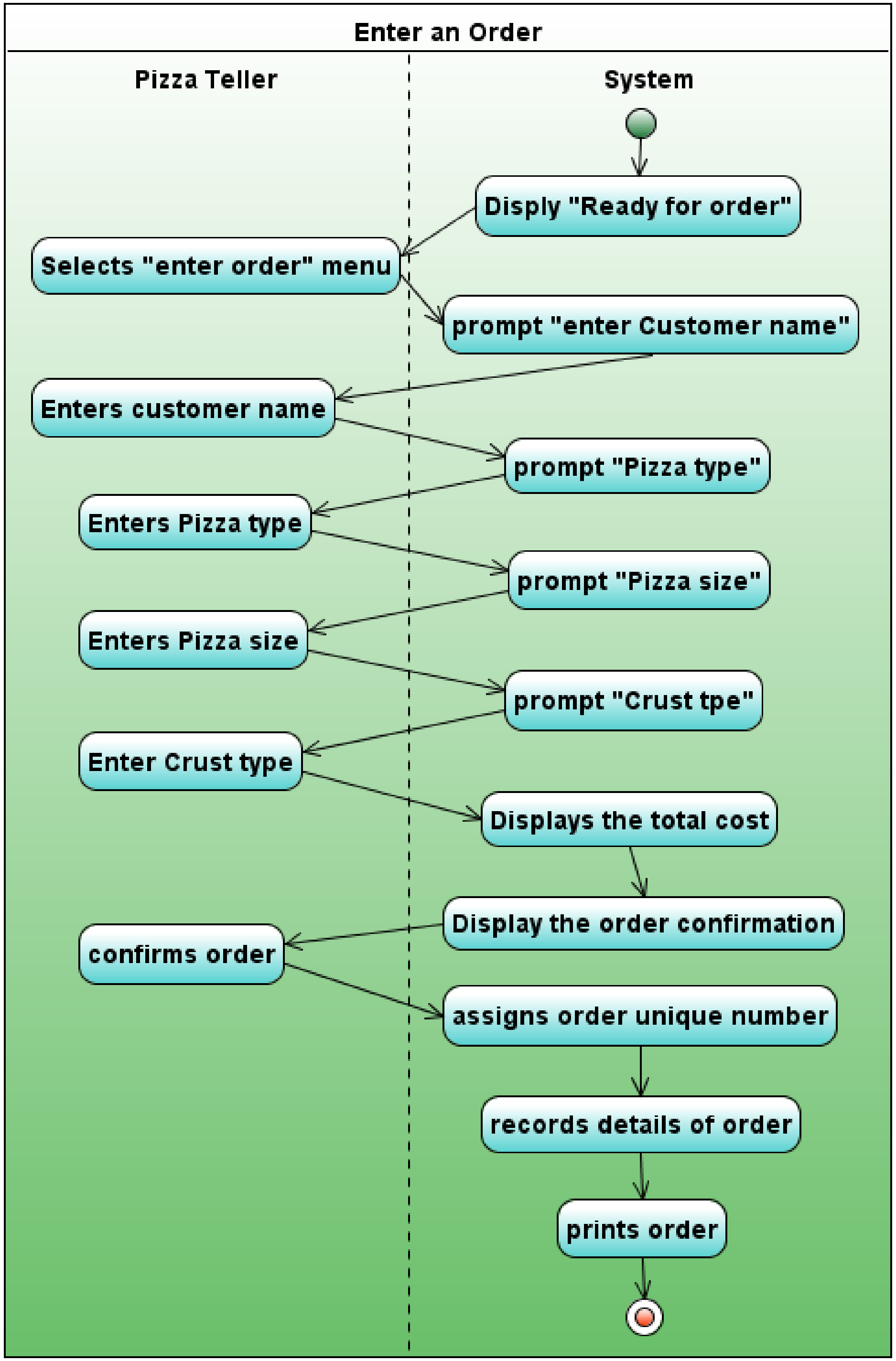
The Pizza Shop use case diagram



The Pizza Shop use case – Enter an Order

Use Case Name	Enter an Order	
Brief Description		
Actors		
Related Use cases		
Entry condition		
Exit condition		
Flow of Events	Actors	System
	1.	1.1
	2.	2.1
	3.	3.1
	4.	4.1
	5.	5.1
	6.	6.1
		6.2
		6.3
Exception condition		

The Pizza S Order



The Pizza Shop:

- Still in the pizza shop

- The total amount should be subject to GST. More than four pizzas ordered get a 20% reduction. Any coupons for special offers submitted by the customer should be subtracted.

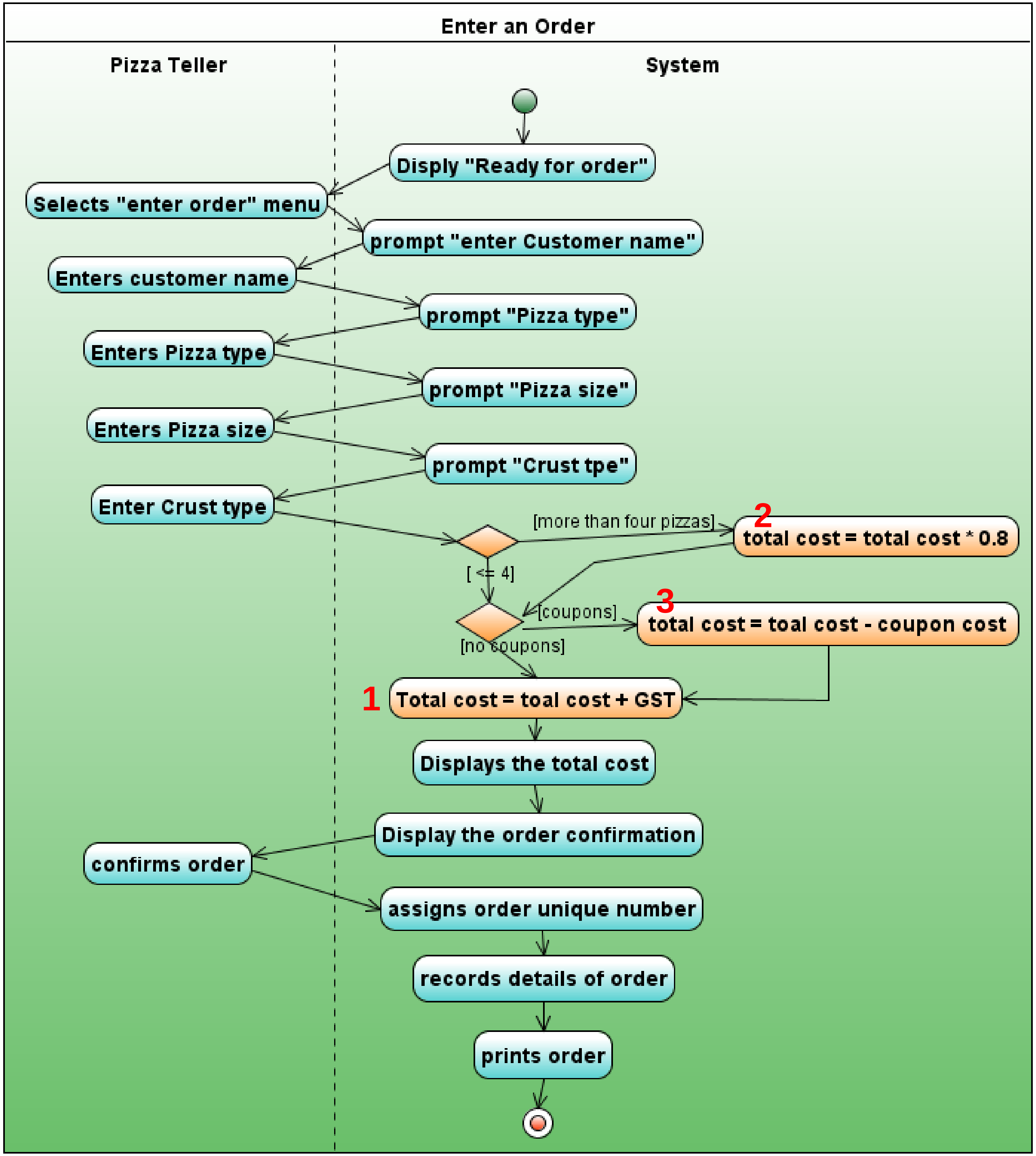
- Prepare an **activity diagram** for computing the bill.

The Pizza Shop:

Use Case Name	Enter an Order	
Brief Description	Pizza Teller enters new order into the system	
Actors	Pizza Teller (or Manager)	
Related Use cases	Include or extend	
Entry condition	The system is waiting for user input.	
Exit condition	The system complete the order.	
Flow of Events	Actors	System
	1. Customer comes to pizza shop to get a pizza. Pizza Teller selects enter order Screen.	1.1 prompt “worker” to enter details of Customer name
	2. Pizza Teller enters Customer name	2.1 prompts for Pizza type
	3. Pizza Teller enters Pizza type	3.1 prompts for Pizza size
	4. Pizza Teller enters Pizza size	4.1 prompts for Crust type
	5. Pizza Teller enters Crust type	5.1 total = total + GST 5.2 displays order for confirmation with pricing
	6. Pizza Teller confirms order	6.1 assigns order unique number 6.2 records details of order 6.3 prints order
Exception condition	5.1 if more than four pizzas ordered get a 20% reduction. 5.2 Any coupons for special offers submitted by the customer should be subtracted.	

The Pizza Sh

1. The total amount should be subject to GST.
2. More than four pizzas ordered get a 20% reduction.
3. Any coupons for special offers submitted by the customer should be subtracted.



Summary

- Requirements Elicitation
  - Questionnaires
  - Task Analysis
  - Scenarios: a synthetic description of the system as a series of interactions between a specific end user and the system
  - Use cases
    - Association/includes/extends/generalization

Summary

- Activity diagrams
  - They are used for business process modelling, for modelling the logic captured by a single use case.
  - Draw swimlanes
  - Draw the start point of the process in the swimlane of the first activity
  - Draw the oval of the first activity
  - Draw an arrow to the location of the second activity
  - Draw subsequent activities, while inserting decision points and synchronization/loop bars where appropriate
  - Draw the end point after the last activity.

Next

- Class and object diagrams