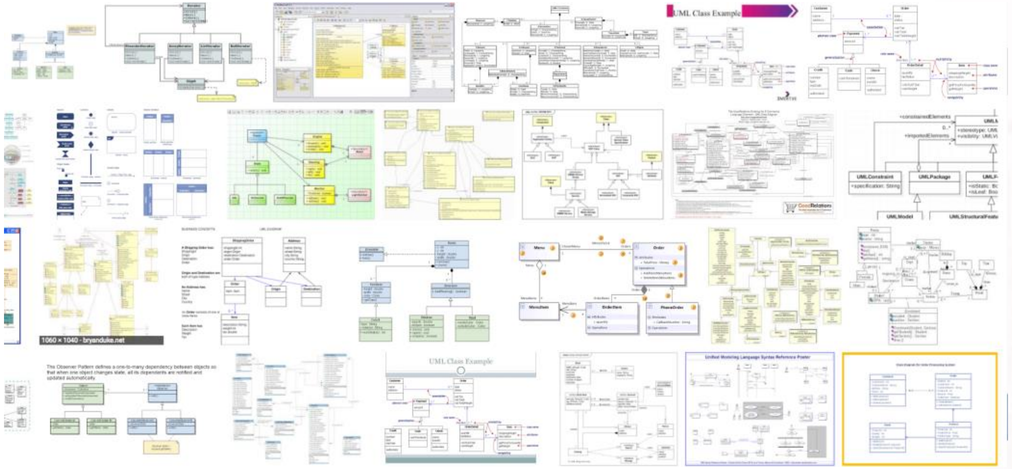
SSW 322: Software Engineering Design VI

Unified Modeling Language (UML)
2020 Spring

Prof. Lu Xiao
lxiao6@stevens.edu
Babbio 513
Office Hour: Monday/Wednesday 2 to 4 pm
Software Engineering
School of Systems and Enterprises

Today's Topics

- Unified Modeling Language (UML)



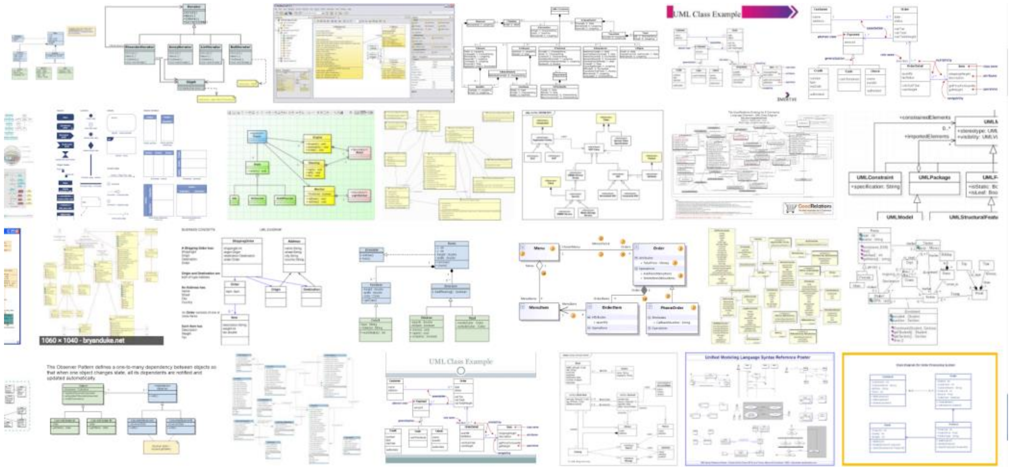
Acknowledgement

UML Distilled by Martin Fowler

http://www.holub.com/goodies/uml
http://creately.com/blog/diagrams/uml-diagram-types-examples/
https://en.wikipedia.org/wiki/Activity_diagram
https://www.tutorialspoint.com/uml/index.htm
https://cloud.smartdraw.com/

What is Unified Modeling Language ?

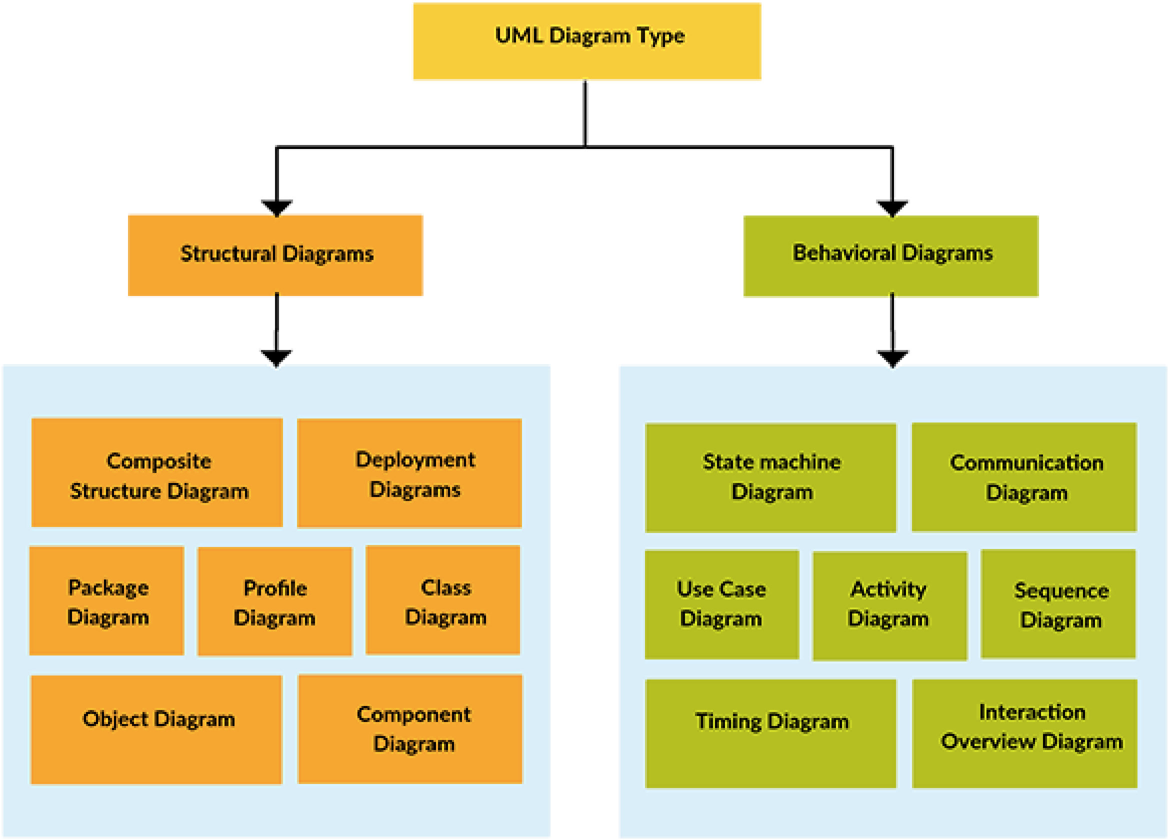
- It is a rich language that can be used to model an ***application structures***, ***behavior*** and even business ***processes***.
- There are 14 UML diagram types to help you model these behavior.



UML Diagram Categories

- Structure diagrams show the things in a system being modeled. In a more technical term, they show different objects in a system.

- Behavioral diagrams shows what should happen in a system. They describe how the objects interact with each other.



UML Diagram Types

- Structural UML Diagrams
  - Class diagram: static structure of a system (backbone of object-oriented method)
  - Package diagram: organize elements of a system into related groups to minimize dependencies between packages
  - Object diagram: the static structure of a system at a particular time.
  - Component diagram: the organization of physical software components, including source code, run-time (binary) code, and executables.
  - Composite structure diagram: the internal part of a class.
  - Deployment diagram: depict the physical resources in a system, including nodes, components, and connections.
- Behavioral UML Diagrams
  - Activity diagram: illustrate the dynamic nature of a system by modeling the flow of control from activity to activity.
  - Sequence diagram: describe interactions among classes in terms of an exchange of messages over time.
  - Use case diagram: model the functionality of a system using actors and use cases.
  - State diagram: describe the dynamic behavior of a system in response to external stimuli.
  - Communication diagram: model the interactions between objects in sequence.
  - Interaction overview diagram: a combination of activity and sequence diagrams.
  - Timing diagram: focuses on processes that take place during a specific period of time.

UML Diagram Types

- Structural UML Diagrams
  - Class diagram: static structure of a system (backbone of object-oriented method)
  - Package diagram: organize elements of a system into related groups to minimize dependencies between packages
  - Object diagram: the static structure of a system at a particular time.
  - Component diagram: the organization of physical software components, including source code, run-time (binary) code, and executables.
  - Composite structure diagram: the internal part of a class.
  - Deployment diagram: depict the physical resources in a system, including nodes, components, and connections.
- Behavioral UML Diagrams
  - Activity diagram: illustrate the dynamic nature of a system by modeling the flow of control from activity to activity.
  - Sequence diagram: describe interactions among classes in terms of an exchange of messages over time.
  - Use case diagram: model the functionality of a system using actors and use cases.
  - State diagram: describe the dynamic behavior of a system in response to external stimuli.
  - Communication diagram: model the interactions between objects in sequence.
  - Interaction overview diagram: a combination of activity and sequence diagrams.
  - Timing diagram: focuses on processes that take place during a specific period of time.

This Class’s Focus

The Use Case Diagram

The Activity Diagram

The Class Diagram

What is a Use Case Diagram?

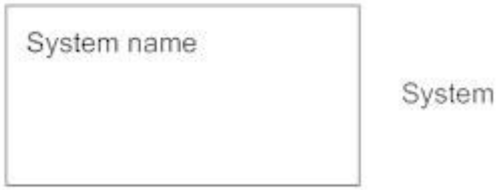
- A dynamic or behavior diagram.
- Model the functionality of a system using ***actors*** and ***use cases***.
  - Use cases are a set of actions, services, and functions that the system needs to perform.
  - The "actors" are people or entities operating under defined roles within the system
- An overview of the ***actors*** involved in a system, different ***functions*** needed by those actors and how these different functions are ***interacted***.

Why a Use Case Diagram is needed?

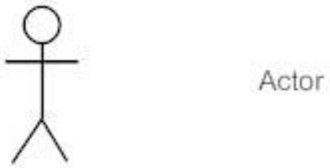
- A great ***starting point*** for any project discussion, because you can easily identify the main actors involved and the main processes of the system.
- Identify internal and external factors that influence the system and should be taken into considerations
- Provide high level analysis from outside the system.
- Specify how the system interacts with actors without worrying about implementation details.

Use Case Diagram Symbols and Notations

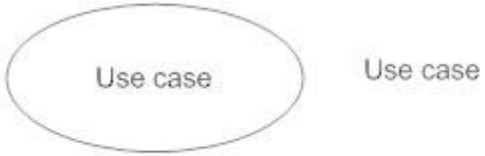
System
Draw your system's boundaries using a rectangle that contains use cases. Place actors outside the system's boundaries.



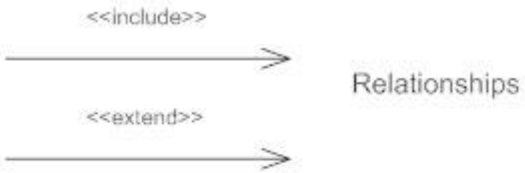
Actors
Actors are the users of a system. When one system is the actor of another system, label the actor system with the actor stereotype.



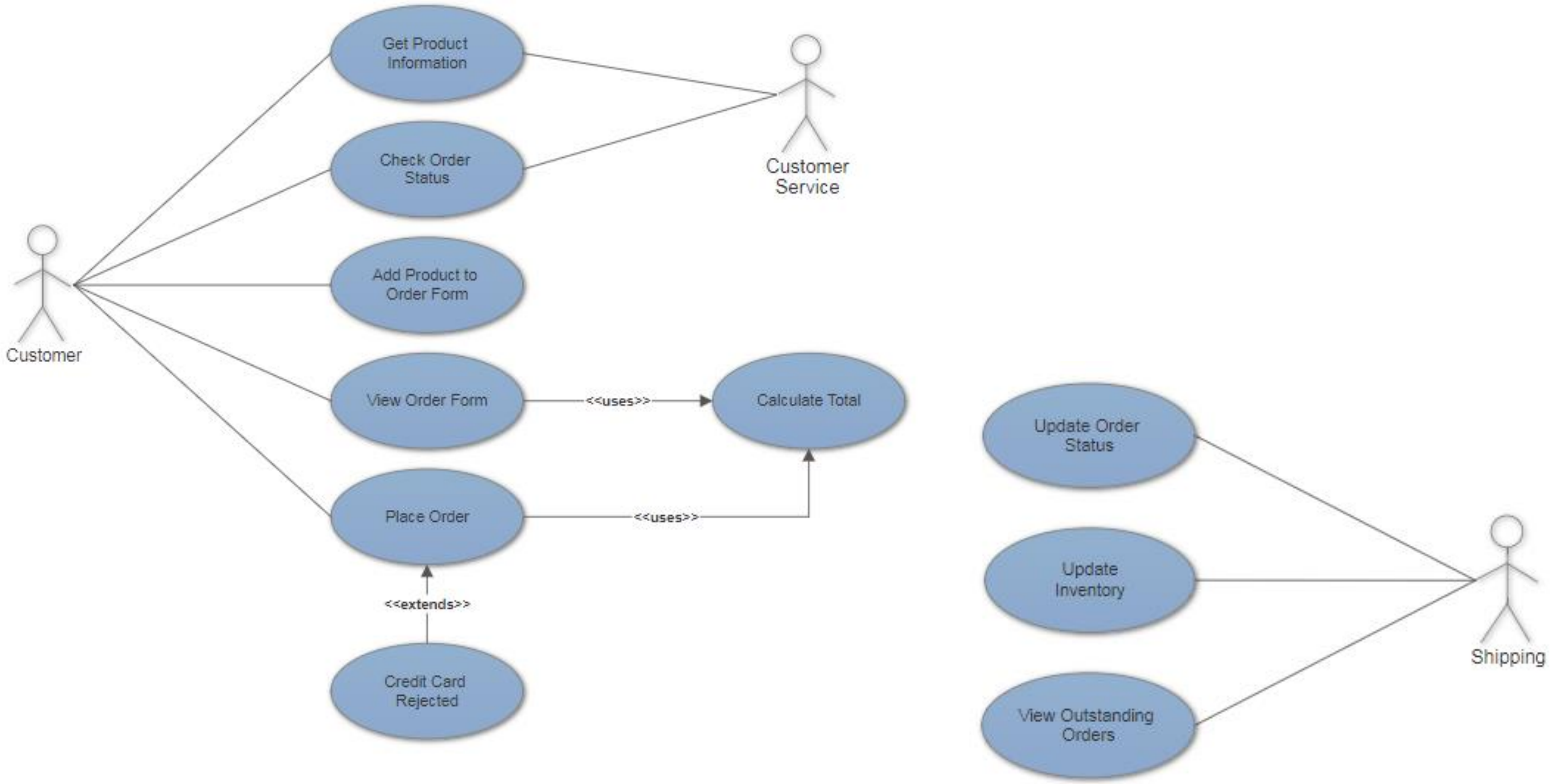
Use Case
Draw use cases using ovals. Label the ovals with verbs that represent the system's functions.



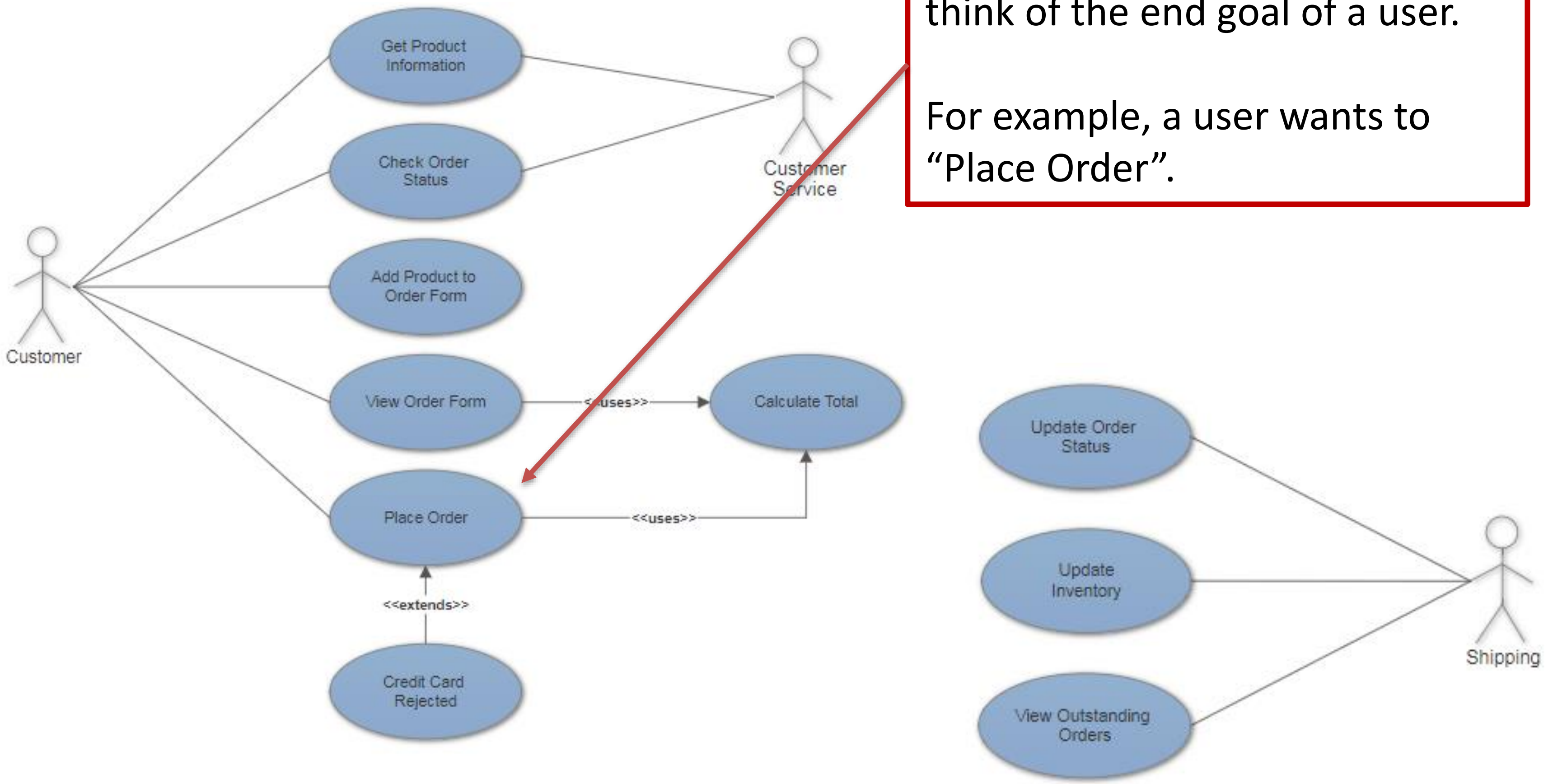
Relationships
Illustrate relationships between an actor and a use case with a simple line. For relationships among use cases, use arrows labeled either "uses" or "extends." A "uses" relationship indicates that one use case is needed by another in order to perform a task. An "extends" relationship indicates alternative options under a certain use case.



Use Case Diagram: Online Shopping



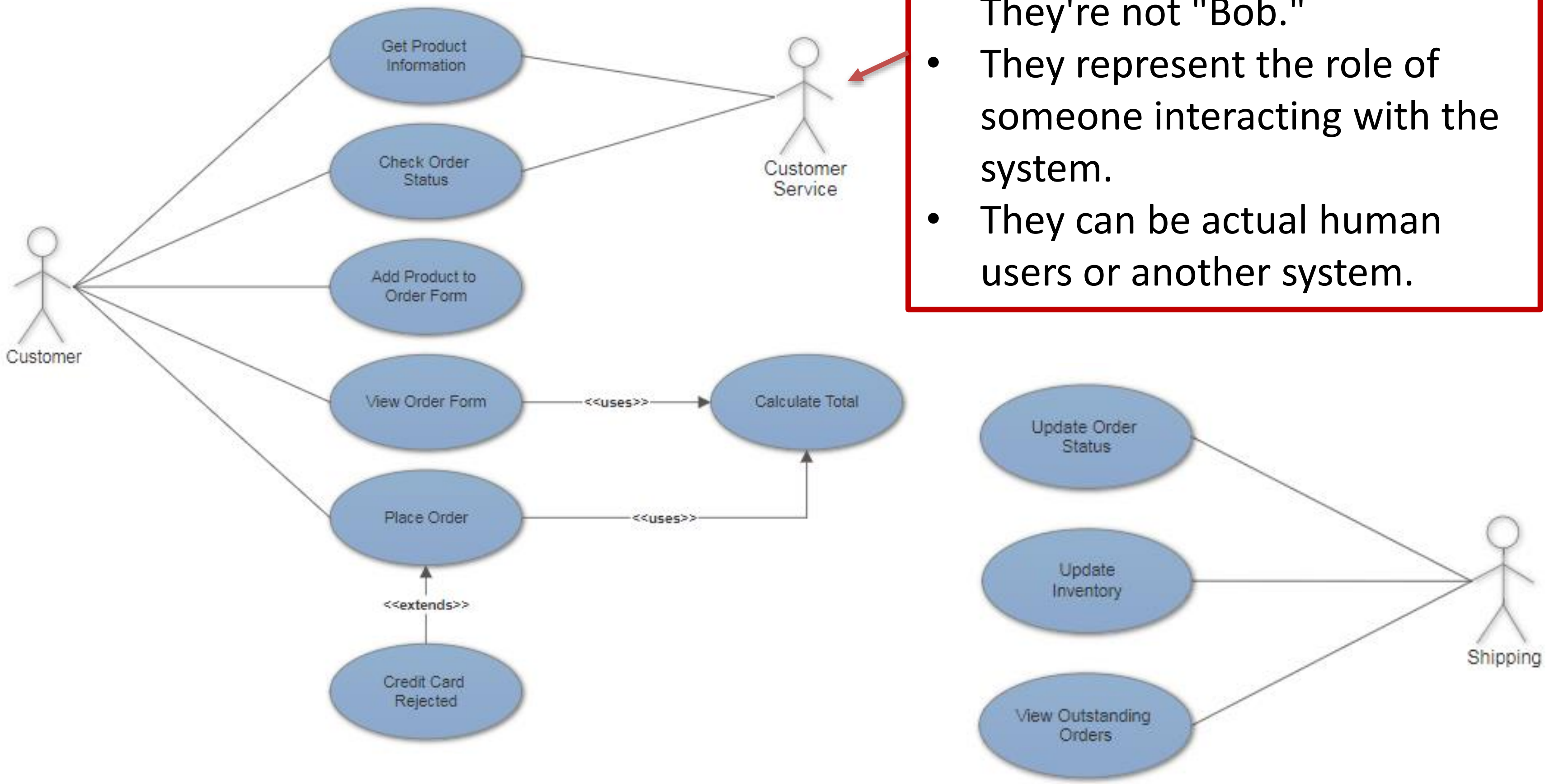
Tips for UML Use Case Diagram



When thinking of use cases, think of the end goal of a user.

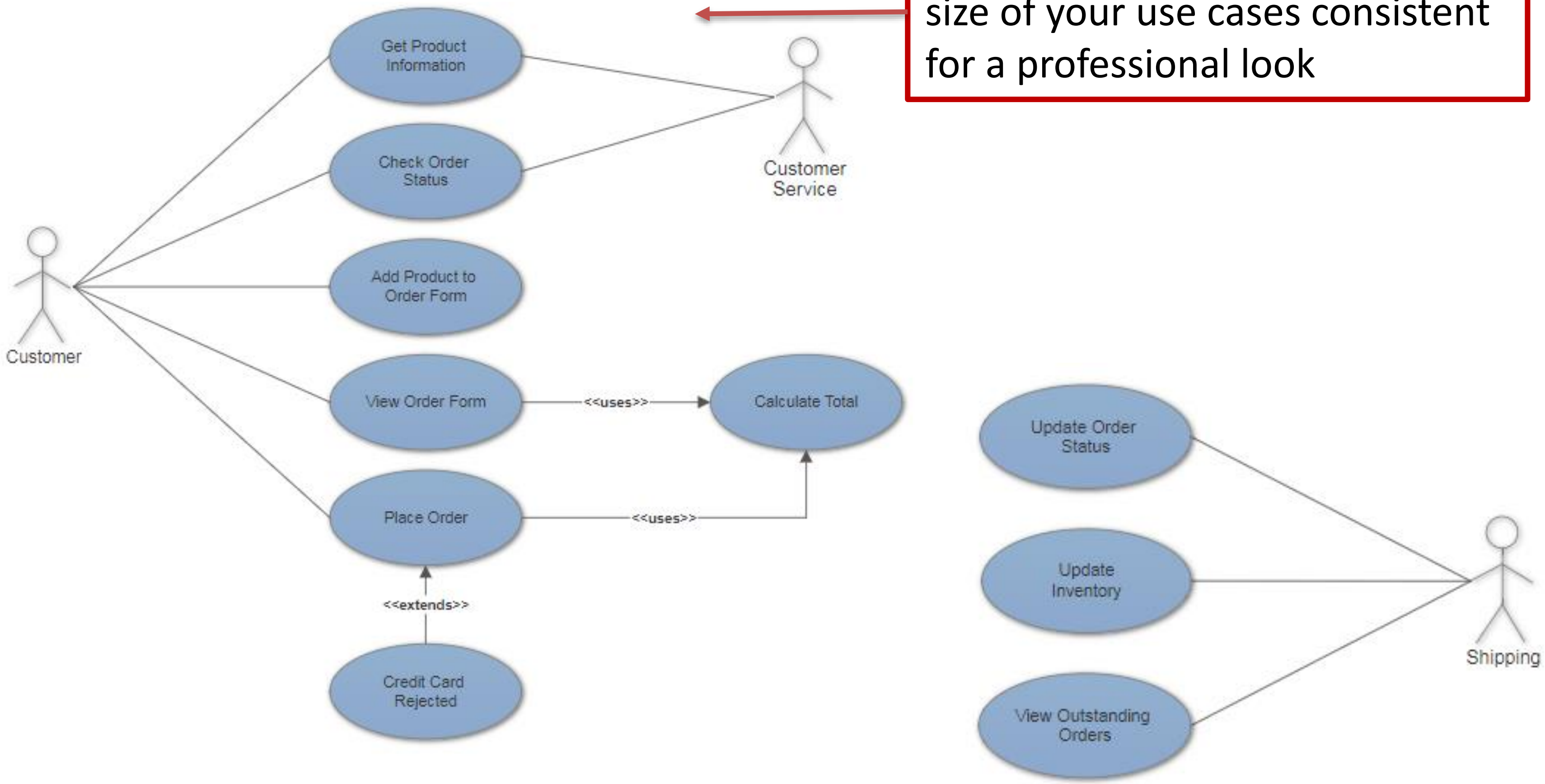
For example, a user wants to "Place Order".

Tips for UML Use Case Diagram



- Actors don't have names. They're not "Bob."
- They represent the role of someone interacting with the system.
- They can be actual human users or another system.

Tips for UML Use Case Diagram



Keep your names short and the size of your use cases consistent for a professional look

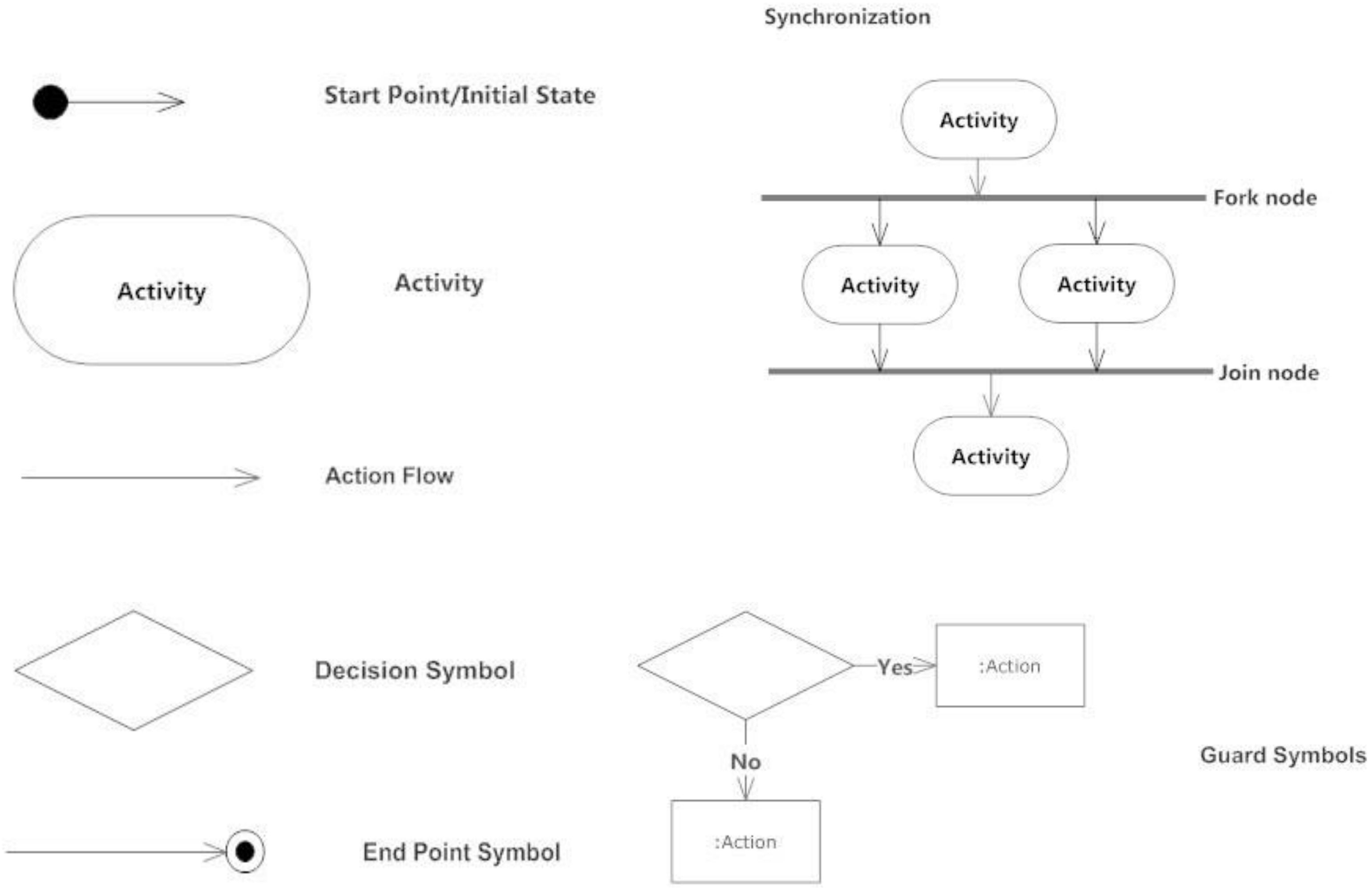
What is an Activity Diagram?

- An activity diagram visually presents a series of ***dynamic*** actions or flow of control in a system.
- Activity diagram is basically a flowchart to represent the flow from one activity to another activity. It shows the steps in a use case diagram.
- Activities modeled can be ***sequential*** and ***concurrent***. In both cases an activity diagram will have a ***beginning*** and an ***end***.

Why an Activity Diagram is needed?

- This diagram is used to model the activities which are business requirements.
- The diagram has more impact on business understanding rather than on implementation details.
- High level understanding of the system's functionalities.

Activity Diagram Symbols and Notations



Synchronization

Start Point/Initial State

Fork node

Activity

Join node

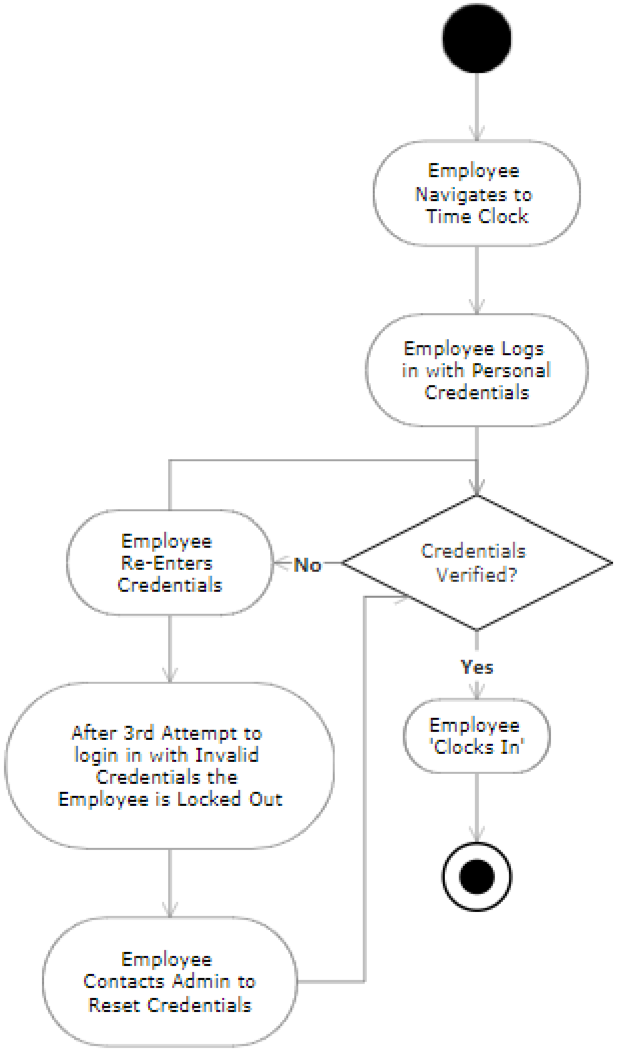
Action Flow

Decision Symbol

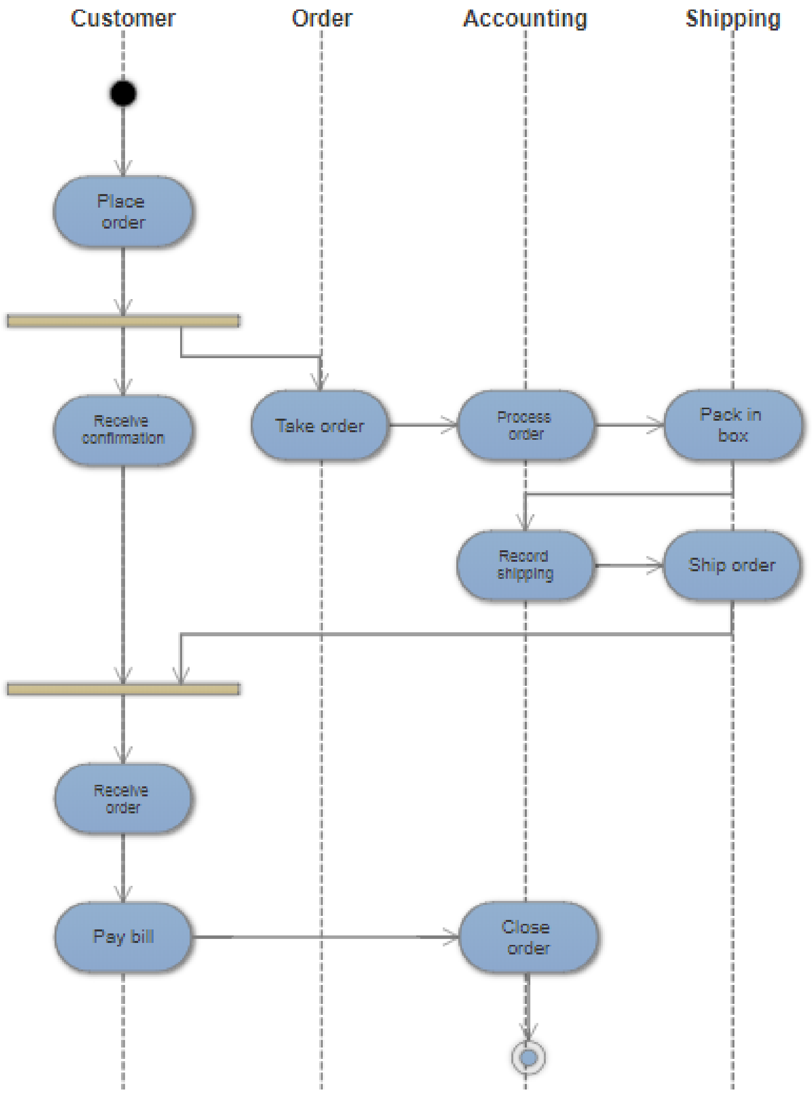
Guard Symbols

End Point Symbol

Activity Diagram: Employee Clock-In


Employee Navigates to Time Clock
Employee Logs in with Personal Credentials
Credentials Verified?
No
Employee Re-Enters Credentials
Yes
Employee 'Clocks In'
After 3rd Attempt to login in with Invalid Credentials the Employee is Locked Out
Employee Contacts Admin to Reset Credentials

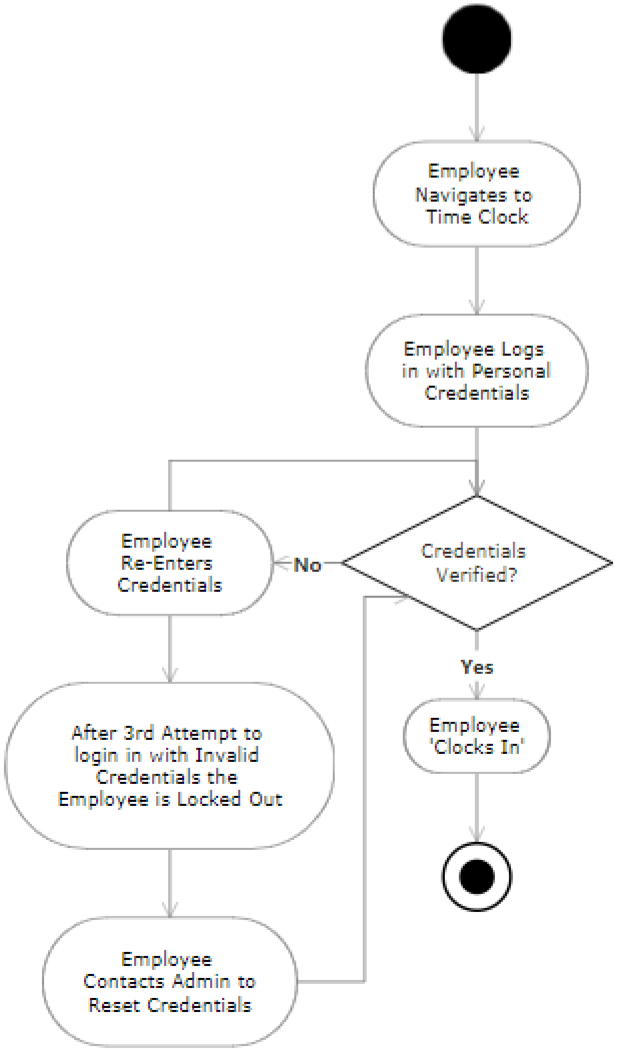
Activity Diagram: Online Order



Swimlanes

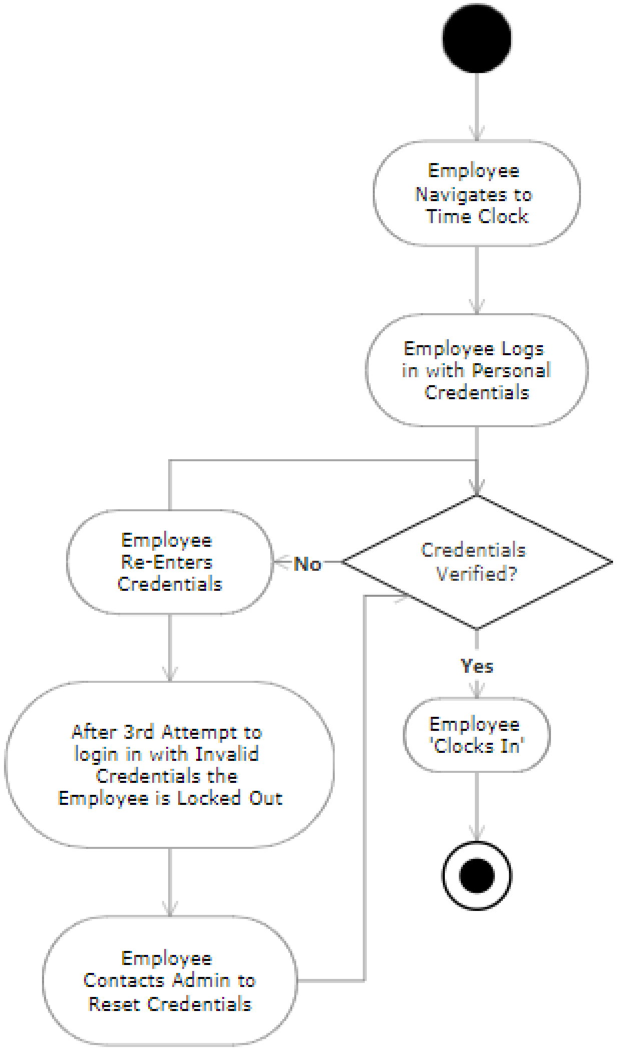
Swimlanes group related activities into one column.

Tips for Activity Diagram


Employee Navigates to Time Clock
Employee Logs in with Personal Credentials
Credentials Verified?
No
Employee Re-Enters Credentials
After 3rd Attempt to login in with Invalid Credentials the Employee is Locked Out
Employee Contacts Admin to Reset Credentials
Yes
Employee 'Clocks In'

- We must have a clear understanding about the elements used in activity diagram.
- The main element of an activity diagram is the activity itself.
- An activity is a function performed by the system.

Tips for Activity Diagram



Employee Navigates to Time Clock
Employee Logs in with Personal Credentials
Credentials Verified?
No
Employee Re-Enters Credentials
After 3rd Attempt to login in with Invalid Credentials the Employee is Locked Out
Employee Contacts Admin to Reset Credentials
Yes
Employee 'Clocks In'

- After identifying the activities, we need to understand how they are associated with constraints and conditions.

- Activities
- Association
- Conditions
- Constraints

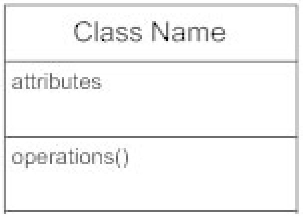
What is a Class Diagram?

- A class diagram models the static structure of a system. It shows relationships between classes, objects, attributes, and operations.
- Class diagram shows a collection of classes, interfaces, associations, collaborations, and constraints. It is also known as a structural diagram.

Why a Class Diagram is needed?

- The class diagrams are widely used in the modeling of object-oriented systems because they are the only UML diagrams, which can be mapped directly with object-oriented languages.
- It is the most popular UML diagram in the coder community.

Class Diagram Symbols and Notations

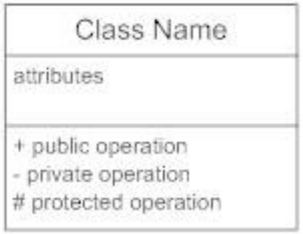


Class Name

attributes

operations()

Class

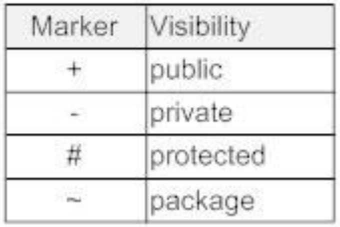


Class Name

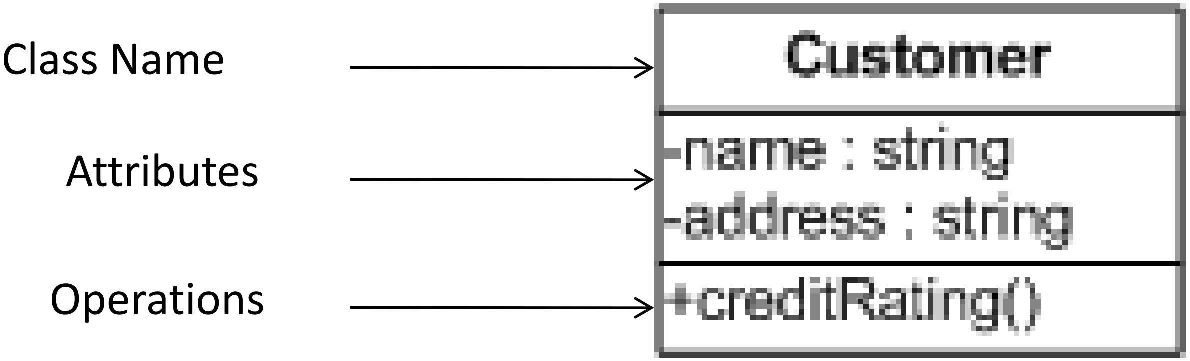
attributes

+ public operation
- private operation
protected operation

Visibility



Marker	Visibility
+	public
-	private
#	protected
~	package



Class Name

Attributes

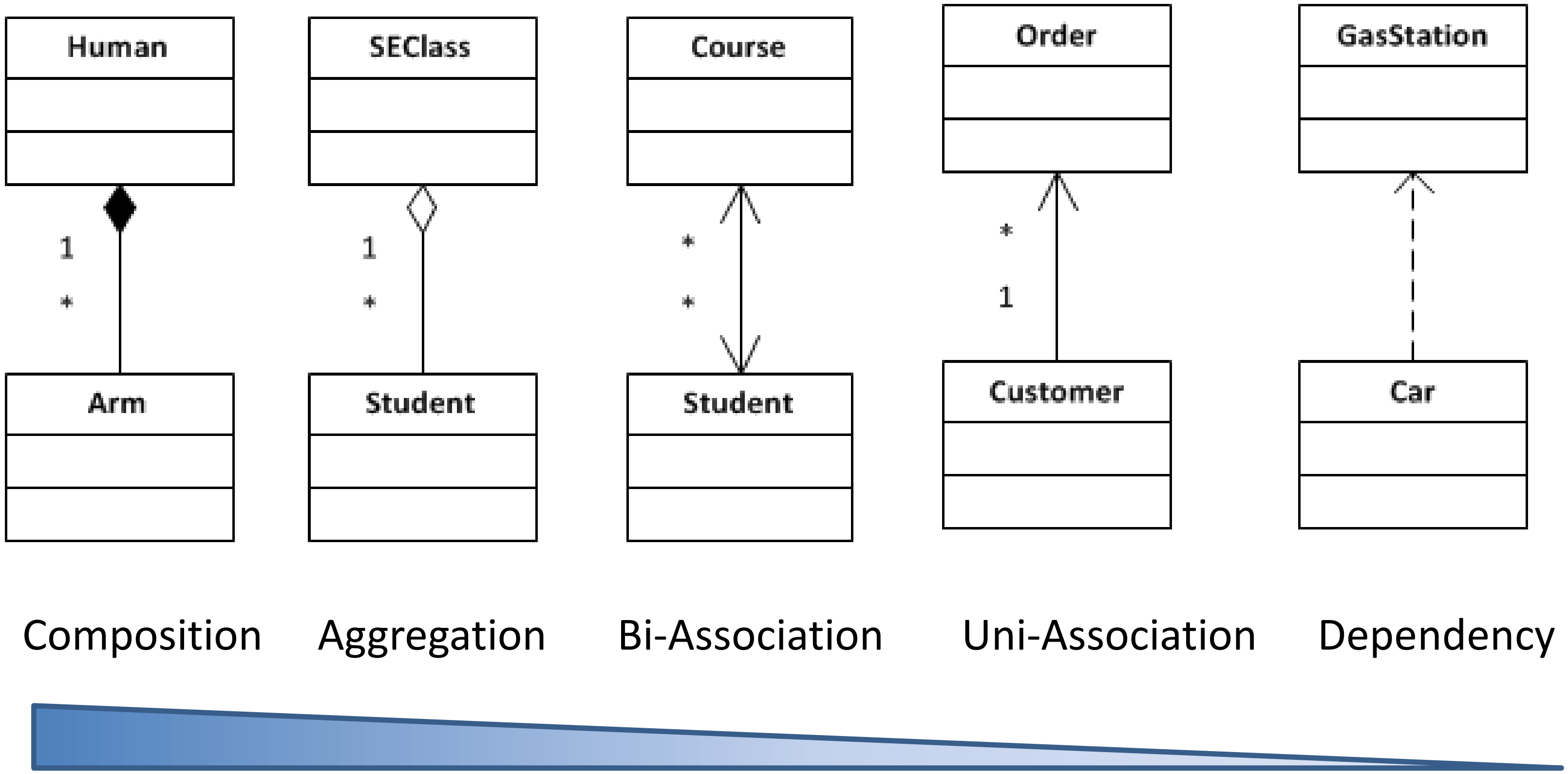
Operations

Customer

-name : string
-address : string

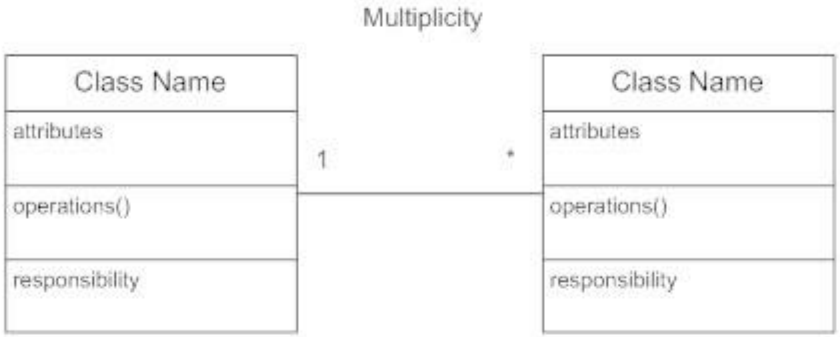
+creditRating()

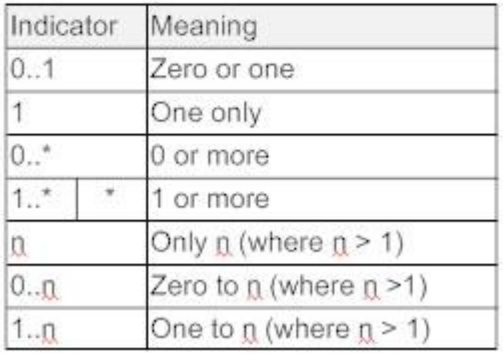
Class Diagram Symbols and Notations



Composition Aggregation Bi-Association Uni-Association Dependency

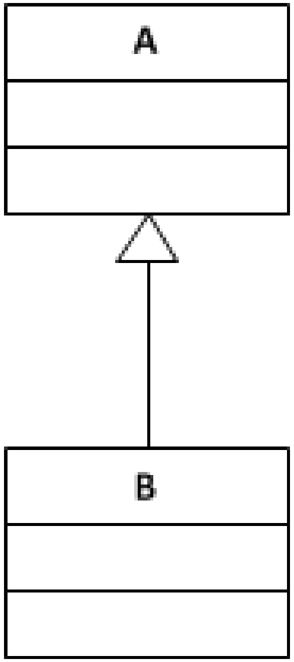
Class Diagram Symbols and Notations



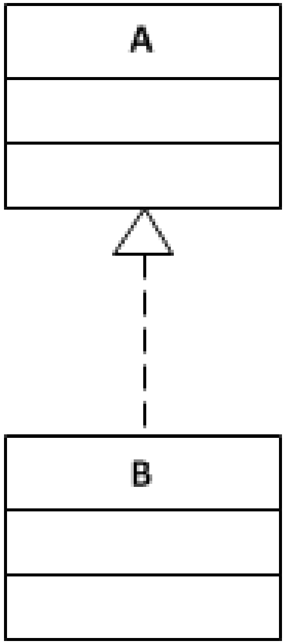


Indicator	Meaning
0..1	Zero or one
1	One only
0..*	0 or more
1..* *	1 or more
n	Only n (where n > 1)
0..n	Zero to n (where n >1)
1..n	One to n (where n > 1)

Class Diagram Symbols and Notations

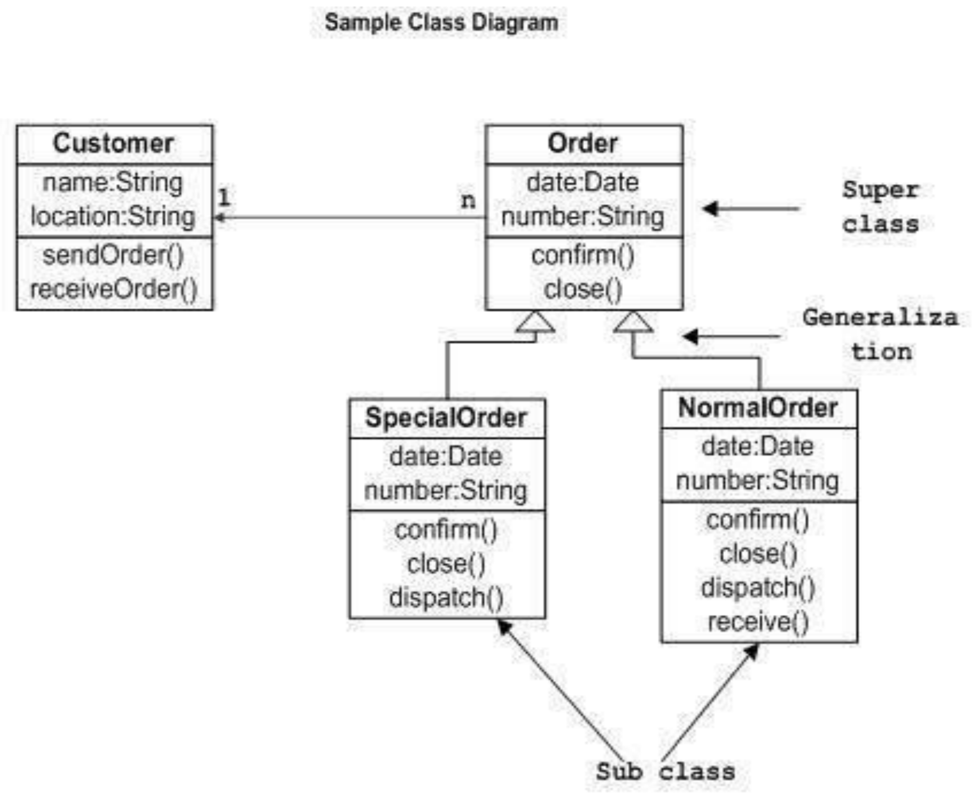


B is derived from A
A generalizes B

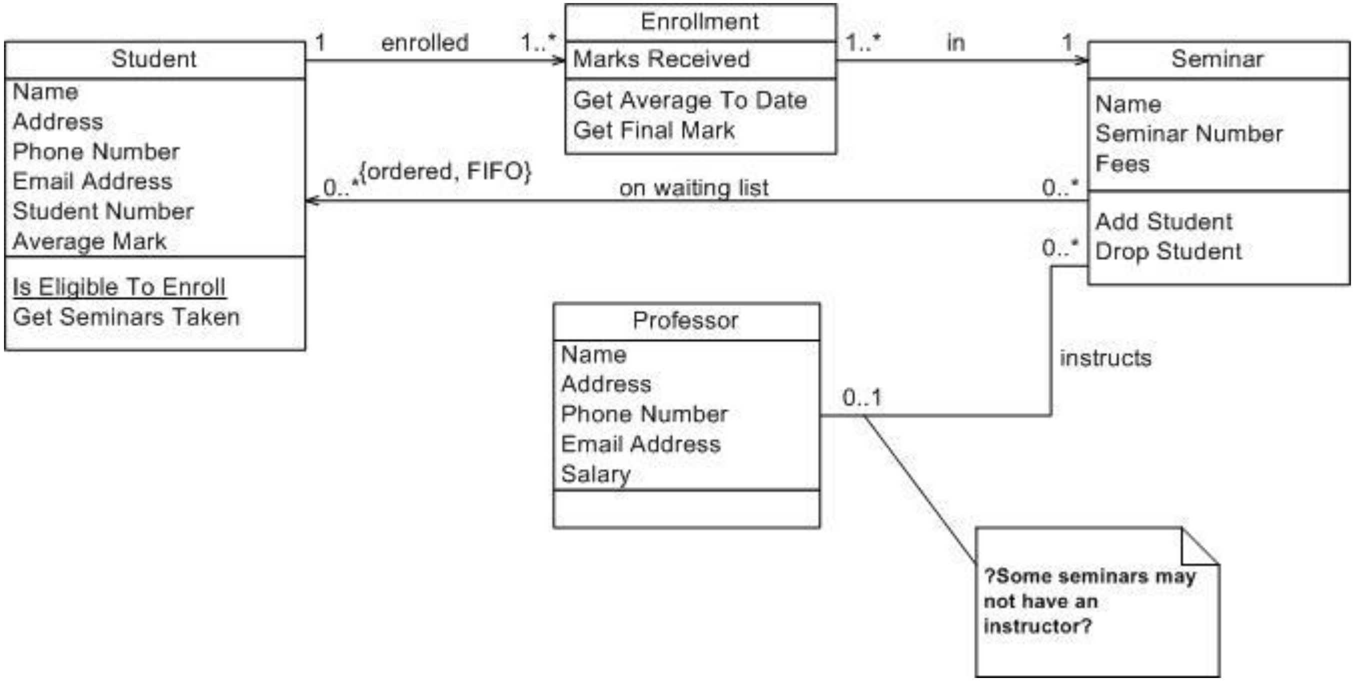


B realizes the
interfaces defined in A

Class Diagram: Order



Class Diagram: Student Seminar



Student

Name
Address
Phone Number
Email Address
Student Number
Average Mark

Is Eligible To Enroll
Get Seminars Taken

1 enrolled 1..*

Enrollment

Marks Received

Get Average To Date
Get Final Mark

1..* in 1

Seminar

Name
Seminar Number
Fees

Add Student
Drop Student

0..* {ordered, FIFO} on waiting list 0..*

0..*

instructs

0..1

Professor

Name
Address
Phone Number
Email Address
Salary

?Some seminars may not have an instructor?

Tips for Class Diagram

- The name of the class diagram should be meaningful to describe the aspect of the system.
- Each element and their relationships should be identified in advance.
- Responsibility (attributes and methods) of each class should be clearly identified
- For each class, minimum number of properties should be specified, as unnecessary properties will make the diagram complicated.

Tools for Creating UML Diagrams

- Proprietary:
  - Rational Rose
  - Visio
  - OmniGraffle.
- Open Source:
  - Dia (http://live.gnome.org/Dia)
  - BOUML (http://bouml.free.fr)
  - ArgoUML (http://argouml.tigris.org)

Thank you