09 Gathering Requirements

Introduction to OOA OOD and UML

2022 Spring

College of Information Science and Engineering

Ritsumeikan University

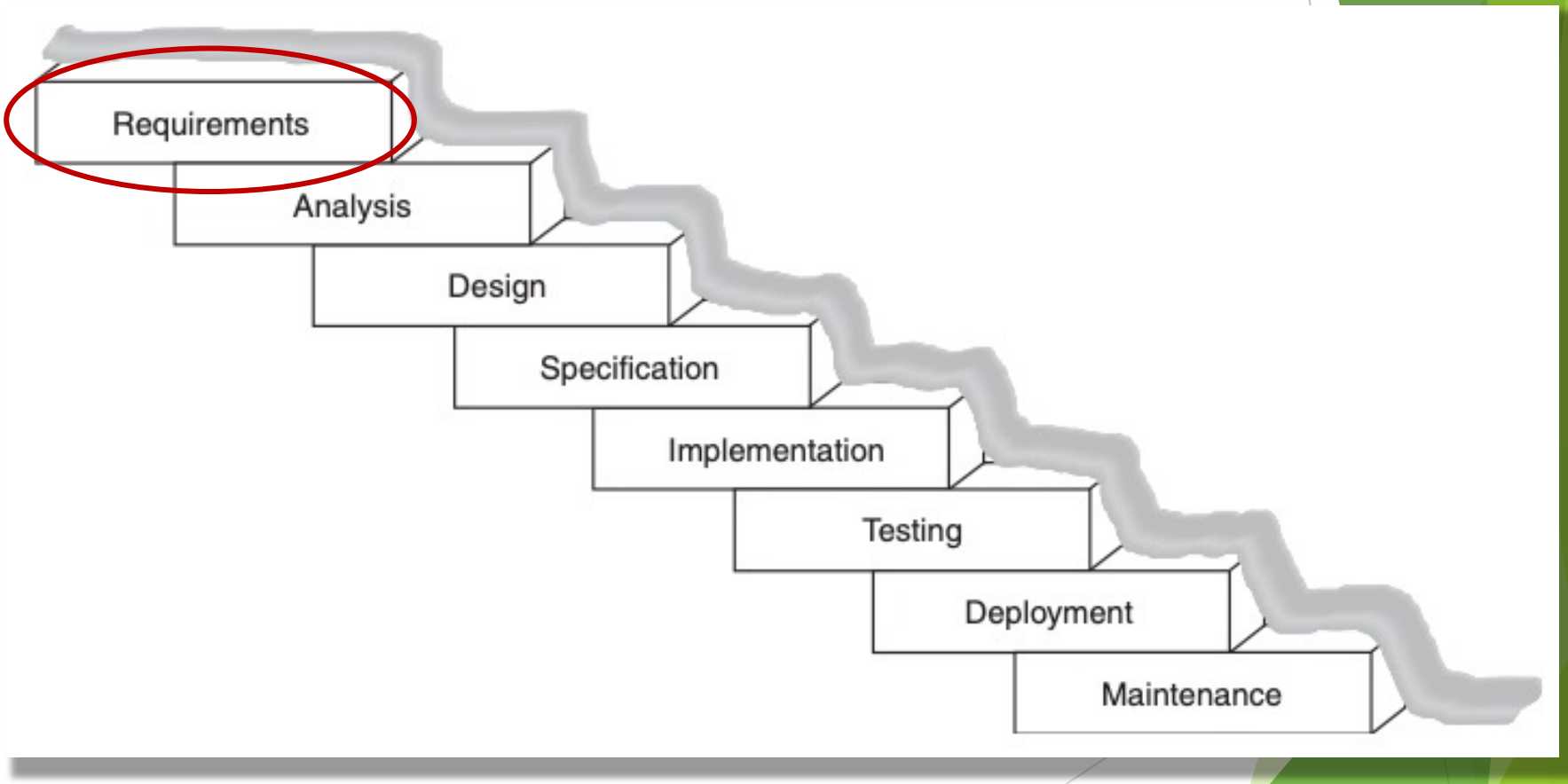
Yu YAN

Outline

- Introduction
- The Birth of A System
- Use Cases
- Business Perspective
- Developer Perspective
- Exercise 09

Where Are We Now?

- Requirement analysis is the first step of any development methodology



Requirements

- Requirement capture is about discovering what we're going to achieve with our new piece of software
- There are two aspects of Requirements:
  - Business modeling: it involves understanding the context in which our software will operate (from the standpoint of customers)
  - System requirements modeling (or functional specification): it means deciding what capabilities the new software will have and writing down those capabilities. All relevant functions must be implemented, and all irrelevant functions must be avoided

Purposes of Requirement Analysis

- Examine the business context:
  - Come up with good reasons to develop the software
  - Investigate the business in which the software will operate ("problem domain")
  - Investigate scenarios of the software usage
- Describe the system requirements:
  - Describe the functionality of the system (software)
  - Detect the system constraints: performance, development cost, resources
  - Non-functional requirements: everything else that needs to be specified (for example, a user interface that can be used easily by novice Internet surfers)

Outline

- Introduction
- The Birth of A System
- Use Cases
- Business Perspective
- Developer Perspective
- Exercise 09

Introduction

- As developers, we must transform the customer's requirements document or mission statement into complete, unambiguous description of the system to be developed, in a standard format that the customer can understand and ratify.
  - Mission statement is a short statement of some new, desirable, business direction.

Case Study -- Mission Statement

Since we automated the tracking of cars at our stores – using bar codes, counter-top terminals and laser readers – we have seen many benefits: the productivity of our rental assistants has increased 20%, cars rarely go missing and our customer base has grown strongly (according to our market research, this is at least partly due to the improved perception of professionalism and efficiency).

The management feels that the Internet offers further exciting opportunities for increasing efficiency and reducing costs. For example, rather than printing catalogs of available cars, we could make the catalog available to every Internet surfer for browsing on-line. For privileged customers, we could provide extra services, such as reservations, at the click of a button. Our target saving in this area is a reduction of 15% in the cost of running each store.

Within two years, using the full power of e-commerce, we aim to offer all of our services via a Web browser, with delivery and pick-up at the customer's home, thus achieving our ultimate goal of the virtual rental company, with minimal running costs relative to walk-in stores.

Nowhere Cars mission statement

The fictitious company's new system is referred to as **Coot**, with the Internet facilities avaliable to customers referred to as **iCoot**

Outline

- Introduction
- The Birth of A System
- Use Cases
- Business Perspective
- Developer Perspective
- Exercise 09

Introduction

- Use cases is invented to define the way in which part of a business or a system is used.
  - Use cases are used to document our understanding of the way a business operates (business requirements modeling). For example: in the car rental business, "Member Reserves Car Model" is a business use case which describes how a member makes a reservation.
  - Use cases are also used to document our understanding of the way to specify what our new software system should be able to do (system requirements modeling). For example: in the car rental business, "Make Reservation" is a system use case (functions of software system)

An Actor

- A use case is written in natural language, broken up into a sequence of steps
- Diagrams can accompany the use case if more explanation is needed
- A use case starts with a participant called an actor
  - The actor then descends into the business, or the system, and eventually returns to the actor
  - The effect of each use case should be of value to the actor
  - The value can mean different things to different people: it could be some information that the actor wishes to retrieve, some effect that the actor wishes to have on the system, some money, a purchase, or pretty much anything else that might motivate them

Outline

- Introduction
- The Birth of A System
- Use Cases
- Business Perspective
- Developer Perspective
- Exercise 09

Business Use Cases Modeling

- For most projects, we would want to produce an entire business model representing how the business operates, or at least that part of the business that surrounds the system we expect to develop.
- A business use case model contain the use cases themselves plus some other bits and pieces:
  - Actor list (with descriptions)
  - Glossary
  - Use cases (with descriptions and details)
  - Communication diagrams (Optional)
  - Activity diagrams (Optional)

Identifying Business Actors (1)

- An actor is either a person playing some role within the business, a department, or a separate software system
  - Identifying actors helps us to identify the ways in which the business is used, which will, in turn, indicate what the use cases are

Identifying Business Actors (2)

- For example, in the car rental business, the actor list can be:
  - Assistant: An employee at one of our stores who helps Customers to rent Cars and reserve Car Models.
  - Customer: A person who pays us money in return for one of our standard services
  - Member: A Customer whose identity and credit-worthiness have been validated and who, therefore, has access to special services
  - NonMember: A Customer whose identity and credit-worthiness have not been checked and who, therefore, must provide a deposit to make a Reservation or surrender a copy of their license to rent a Car.
  - Auk: The pre-existing system that handles Customer details, Reservations, Rentals and the catalog of available Car Models.
  - DebtDepartment: The department of Nowhere Cars that deals with unpaid fees.
  - LegalDepartment: The department of Nowhere Cars that deals with accidents in which a rented Car has been involved.

Writing the Project Glossary (1)

- A glossary keeps data and process together

Case Study

Nowhere Cars Glossary

Term	Definition
Car (Business object)	Instance of a CarModel kept by a Store for Rental purposes.
CarModel (Business object)	A model in our Catalog, available for Reservation.
Customer (Business actor, business object)	A person who pays us money in return for one of our standard services.
Member (Business object)	A Customer whose identity and credit-worthiness have been validated and who, therefore, has access to special services (such as making Reservations by phone, or over the Internet).
...	

Writing the Project Glossary (2)

- Each entry in the glossary defines a term
- Most of the terms will apply in several contexts
- The relationships that each term has can be recorded to the development phases (for example, business actor, system actor, etc.)

- Business actor: An actor appearing in the business requirements.
- Business object: An object appearing in the business requirements.
- System actor: An actor appearing in the system requirements.
- System object: An object appearing (inside the system) in the system requirements.
- Analysis object: An object appearing in the analysis model.
- Deployment artifact: Something deployed in the system, such as a file.
- Design object: An object appearing in the design model.
- Design node: A computer or process that forms part of the system architecture.
- Design layer: A vertical partition of a subsystem.
- Design package: A logical grouping of classes, used to organize the development.

Identifying Business Use Cases (1)

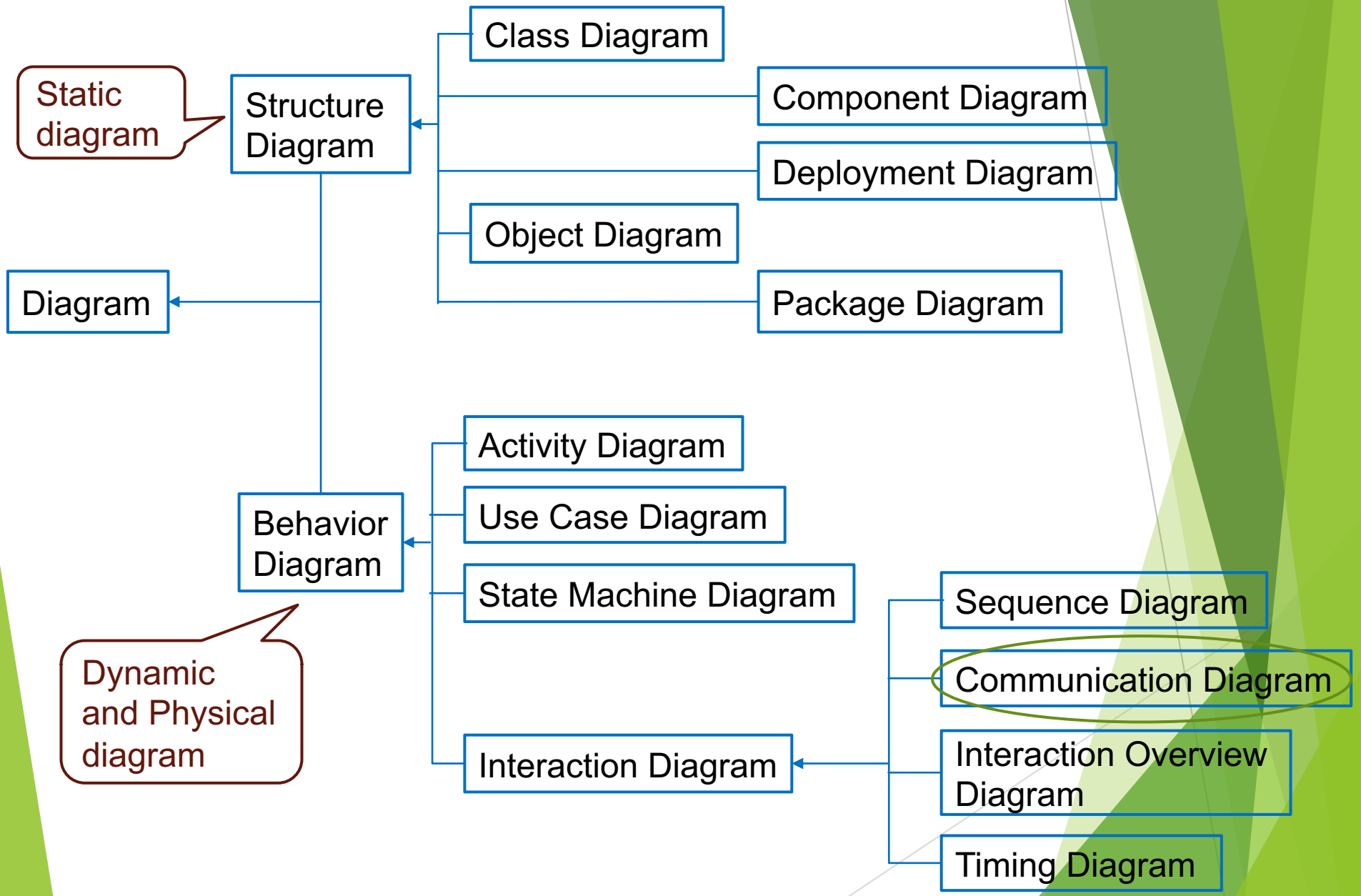
- There is no set rule for deciding how to break the business down into business use cases
  - The common sense, logic and experience will help
  - Working with sponsors will help
  - Talking to an assistant on the shop floor will also help
- At all times, when trying to find use cases, keep the following question at the back of your mind:
  - "What are the key activities that make this business work?"

Identifying Business Use Cases (2)

- For example, the following is a business use case for the car rental business:
  - Use case (B3) "NonMember Reserves CarModel: NonMember pays a deposit to be notified when a CarModel becomes available"

1. NonMember tells Assistant which CarModel to reserve.
2. Assistant finds CarModel on Auk.
3. Assistant asks for a deposit for the reservation.
4. Assistant asks for NonMember's License and phone number.
5. Assistant checks License visually.
6. If License looks okay, assistant creates new Reservation and records Liscense number, phone number and scan of the License in Auk.
7. Assistant gives NonMember a ReservationSlip containing the unique reservation number.

The business use case details


Static diagram
Structure Diagram
Class Diagram
Component Diagram
Deployment Diagram
Object Diagram
Package Diagram
Diagram
Behavior Diagram
Activity Diagram
Use Case Diagram
State Machine Diagram
Interaction Diagram
Sequence Diagram
Communication Diagram
Interaction Overview Diagram
Timing Diagram
Dynamic and Physical diagram

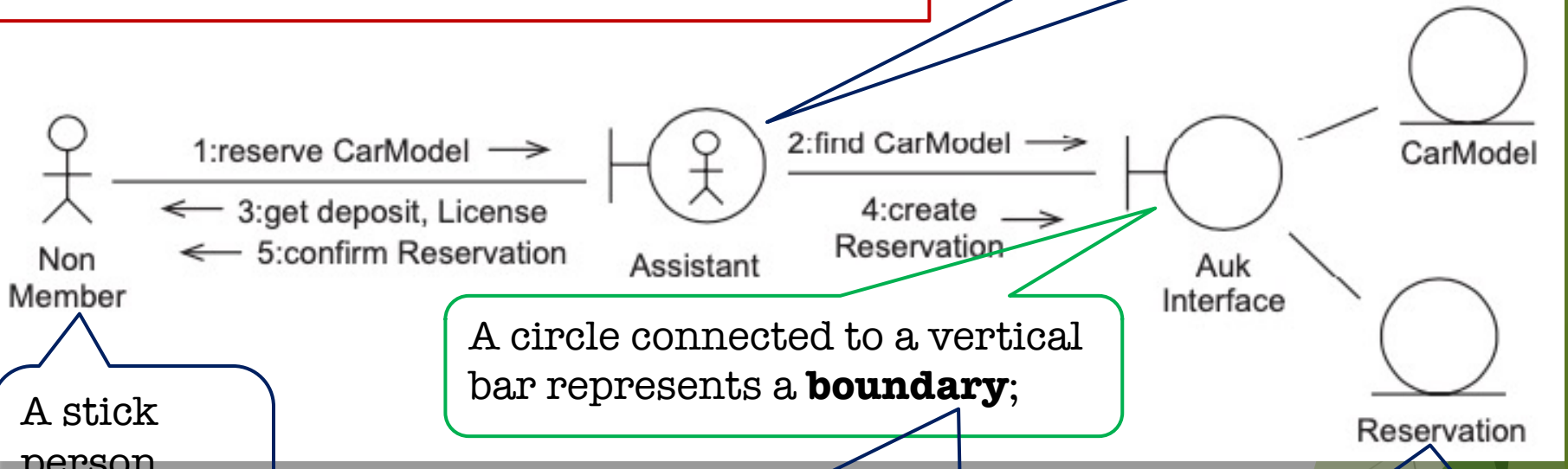
Illustrating Use Cases on Communication Diagram(1)

- A communication diagram shows a series of interactions between actors and objects
  - A sequence diagram focuses on the interactions themselves and the order in which they take place

Illustrating Use Cases on Communication Diagram(2)

- Making a reservation involves a nonmember, an assistant acting as a business boundary, a piece of software called Auk and two business objects

A stick person inside a **boundary** icon represents a human actor playing some kind of interface role;



A stick person represents an actor;

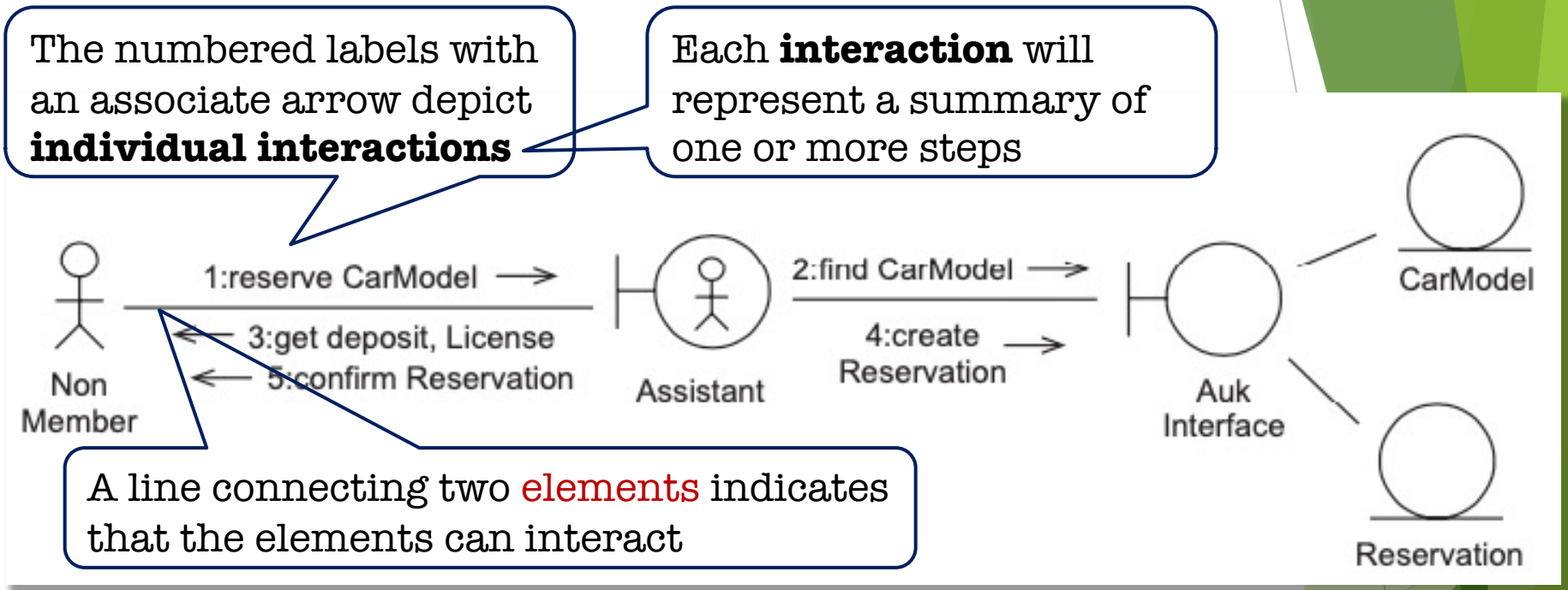
A circle connected to a vertical bar represents a **boundary**;

Something that manages the interaction between other elements – usually this is a piece of software, but it could also be a person;

A circle standing on a line represents a business object or entity;

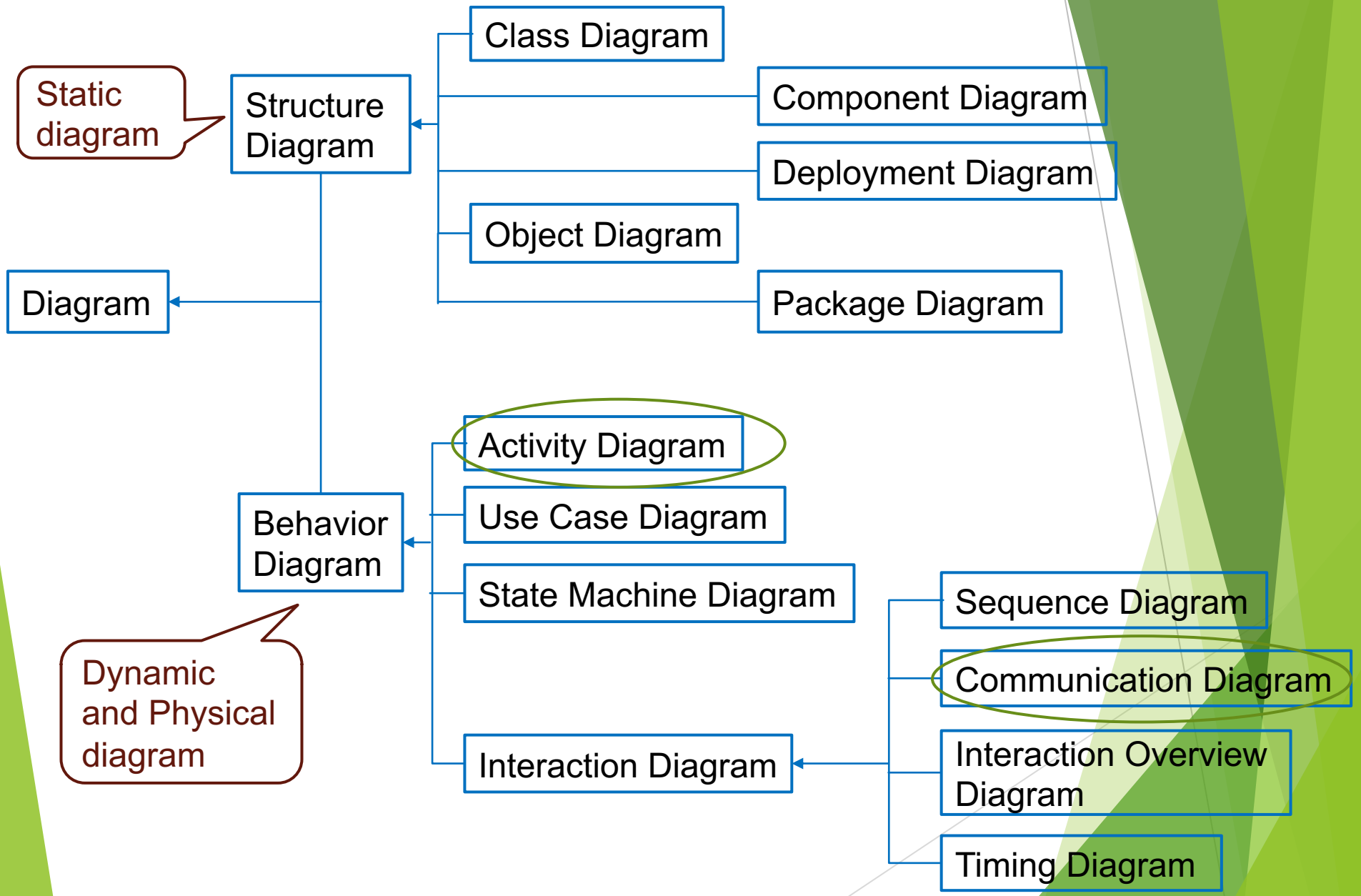
Illustrating Use Cases on Communication Diagram(3)

- Five communicating elements



- One communication diagram per use case is enough, no more

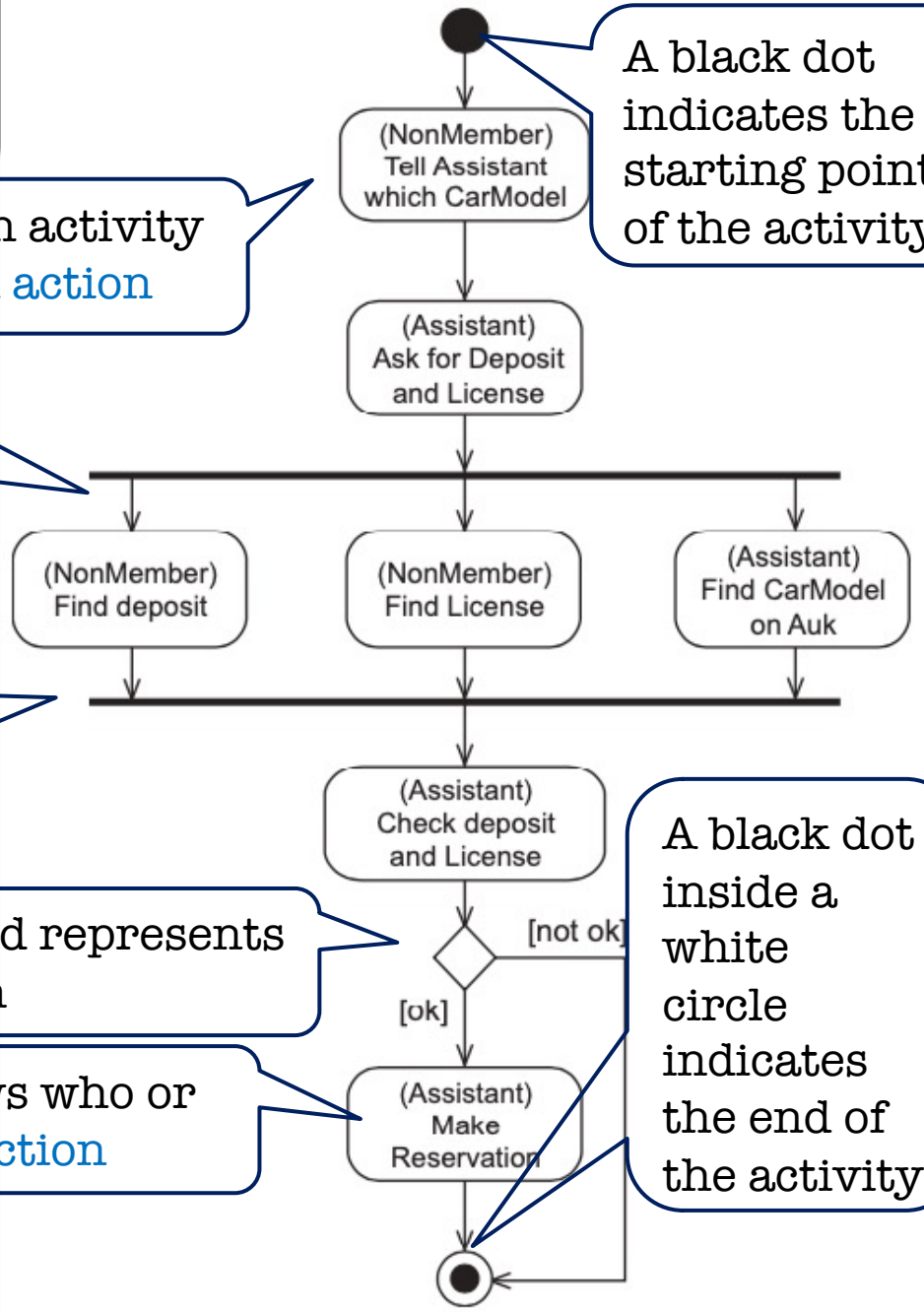
https://astah.net/support/astah-pro/user-guide/communication-diagram/


Static diagram
Structure Diagram
Class Diagram
Component Diagram
Deployment Diagram
Object Diagram
Package Diagram
Diagram
Behavior Diagram
Activity Diagram
Use Case Diagram
State Machine Diagram
Interaction Diagram
Sequence Diagram
Communication Diagram
Interaction Overview Diagram
Timing Diagram
Dynamic and Physical diagram

Illustrating Use Cases on An Activity Diagram (1)

- An activity diagram shows dependencies between (parallel) activities as we move from an initial starting point to a desired goal
  - They are similar to flow charts

Illustrating Use Cases on An Activity Diagram (2)



A black dot indicates the starting point of the activity

Each rounded box in an activity diagram represents an action

An open ended arrow indicaes that the source action must be completed before the destination action is started

Thick black lines are used to indicate the beginning and end of a set of concurrent actions

A diamond represents a decision

A name in parentheses shows who or what is responsible for the action

A black dot inside a white circle indicates the end of the activity

https://astah.net/support/astah-pro/user-guide/activity-diagram/

Outline

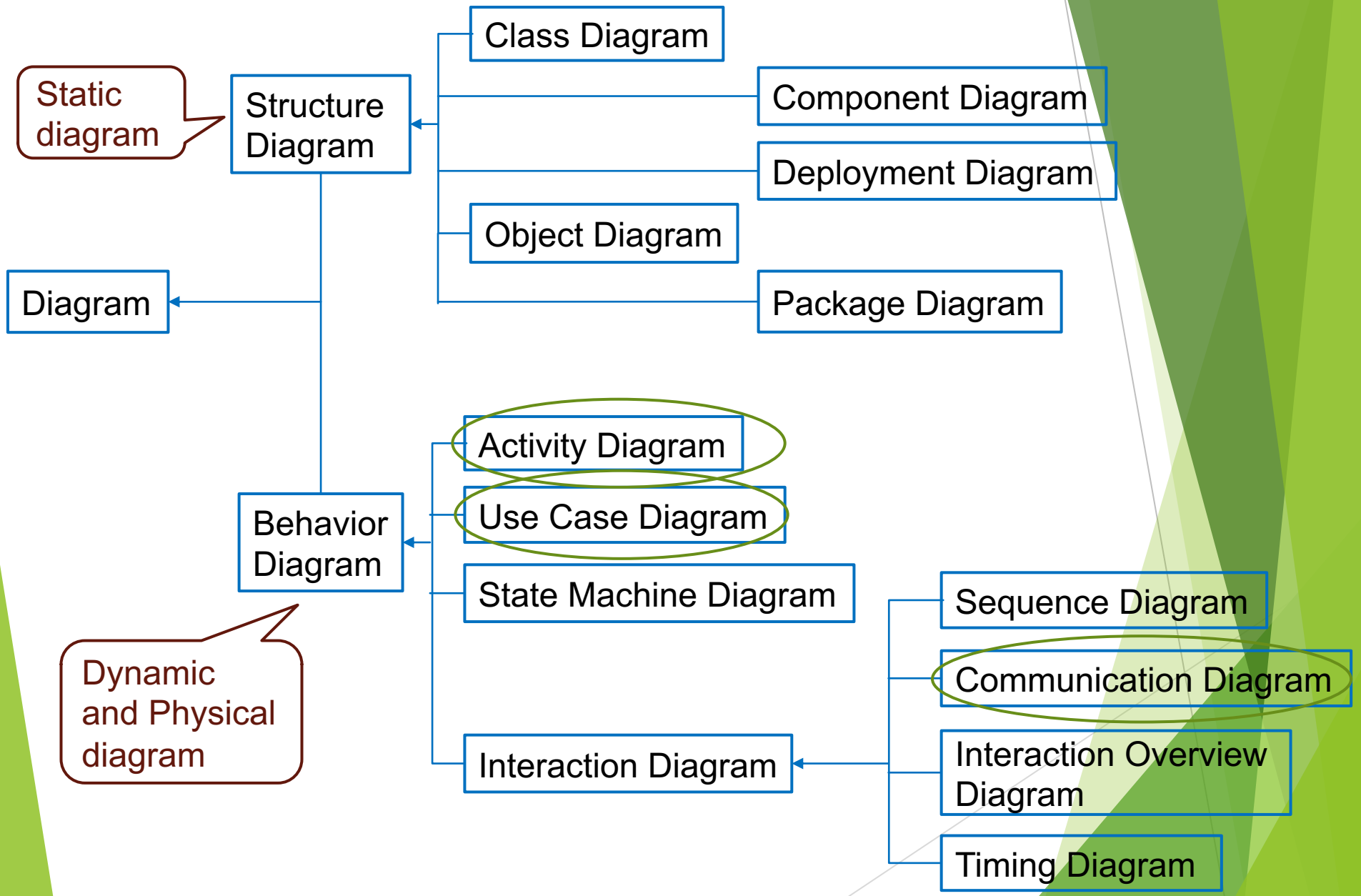
- Introduction
- The Birth of A System
- Use Cases
- Business Perspective
- Developer Perspective
- Exercise 09

Introduction

- A system use case model comprises:
  - An actor list (with descriptions)
  - A use case list (with descriptions)
  - A use case diagram
  - Use case details (including any related nonfunctional requirements)
  - A use case survey
  - Supplementary requirements (system requirements that don't fit with any particular use case)
  - User interface sketches
  - An enhanced glossary
  - Use case priorities

System Actor List

- Assume the new system we are going to implement is called **iCoot**
- **iCoot** system actor list will be:
  - Customer: A person using a Web browser to access **iCoot**
  - Member: A Customer who has presented their name, address and CreditCard details at one of our Stores; each Member is given an Internet password to accompany their membership number
  - Assistant: An employee at a Store who contacts Members to tell them about the progress of their Reservations.


Static diagram
Structure Diagram
Class Diagram
Component Diagram
Deployment Diagram
Object Diagram
Package Diagram
Diagram
Behavior Diagram
Activity Diagram
Use Case Diagram
State Machine Diagram
Interaction Diagram
Sequence Diagram
Communication Diagram
Interaction Overview Diagram
Timing Diagram
Dynamic and Physical diagram

System Use Case List

- **iCoot** system use case list will be:

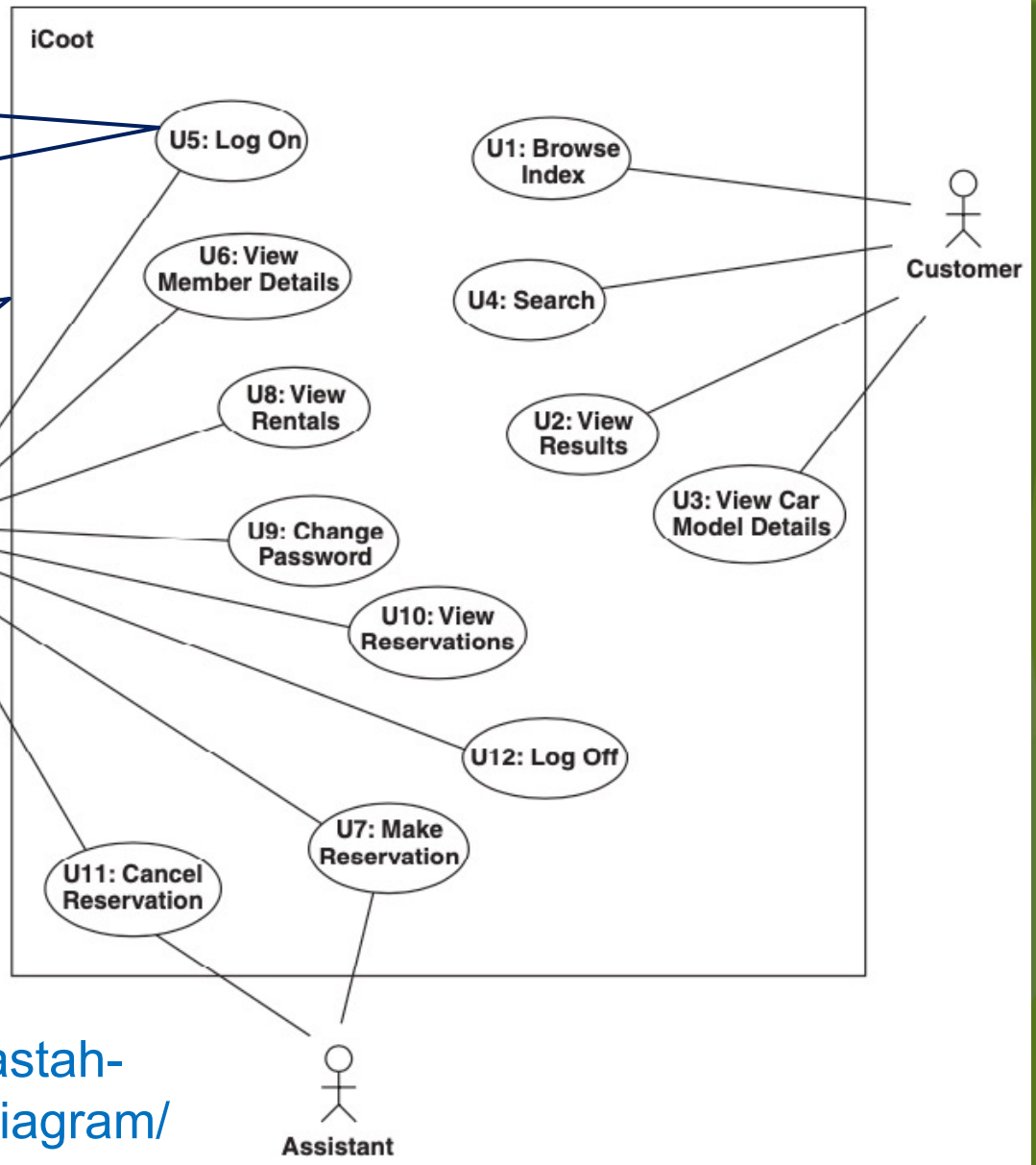
- (U1) Browse Index: A Customer browses the index of CarModels
- (U2) View Results: A Customer is shown the subset of CarModels that was retrieved
- (U3) View CarModel Details: A Customer is shown the details of a retrieved CarModel, such as description and advert
- (U4) Search: A Customer searches for CarModels by specifying Categories, Makes and engine sizes
- (U5) Log On: A Member logs on to **iCoot** using their membership number and current password
- (U6) View Member Details: A Member views a subset of their details stored by **iCoot**, such as name, address and CreditCard details
- (U7) Make Reservation: A Member reserves a CarModel when viewing its details
- (U8) View Rentals: A Member views a summary of the Cars they're currently renting
- (U9) Change Password: A Member changes the password that they use to log on
- (U10) View Reservations: A Member views a summary of Reservations that are not yet concluded, such as date, time and CarModel
- (U11) Cancel Reservation: A Member cancels a Reservation that is not yet concluded
- (U12) Log Off: A Member logs off from **iCoot**.

A System Use Case Diagram

Each use case is shown as a number and a title, inside a bubble

The box around all the use cases represents the boundary of the system – the system name can be put inside the box

The actors are added outside of the boundary



https://astah.net/support/astah-pro/user-guide/usecase-diagram/

Class Vocabularies

Use Case, Actors, Glossary, Communication Diagram, Activity Diagram, Use Case Diagram

Summary

- The importance of specifying functional requirements and nonfunctional requirements in a requirements phase before any coding begins.
- Modeling the business context and system functionality using high-level business use cases and identifying actors.
- Modeling the system requirements with a complete use case model comprising use cases, use case diagram, etc.

Outline

- Introduction
- The Birth of A System
- Use Cases
- Business Perspective
- Developer Perspective
- Exercise 09

Exercise 09

- Deadline: **2022/06/16 (Mon.) 9:00**
- Please submit your answer file “UML_Ex09_Your group member names.pdf” to “Exercise 09” under “Assignments” tab in Manaba +R
- The maximum points for “Exercise 09” will be **10p**
- If you put a wrong file name or wrong file format, your assignment will not be evaluated. Please be careful!

Ex09 (Group work)

- Case-based exercise: Please choose one of the following business to design:
  1. One communication diagram for one use case
  2. One activity diagram for another use case
  3. One use case diagram (A document to describe each use case is necessary)
- Be careful, once you decide the business, you will stay designing a system for the business in the future.
- The grading will be based on your designs. Please try your best to think as much as possible.
- Two business:
  1. A simplified system of an automatic teller machine
  2. A simplified system of the book management in a library