California State University, Sacramento
Computer Science Department

CSc 131
Computer Software Engineering

Fall 2022

Lecture # 4

Object-Oriented Analysis & Unified Modeling Language (UML)

-Use Case Model-

UML

The UML is a graphical language for

- Specifying
- Visualizing
- Constructing
- Documenting

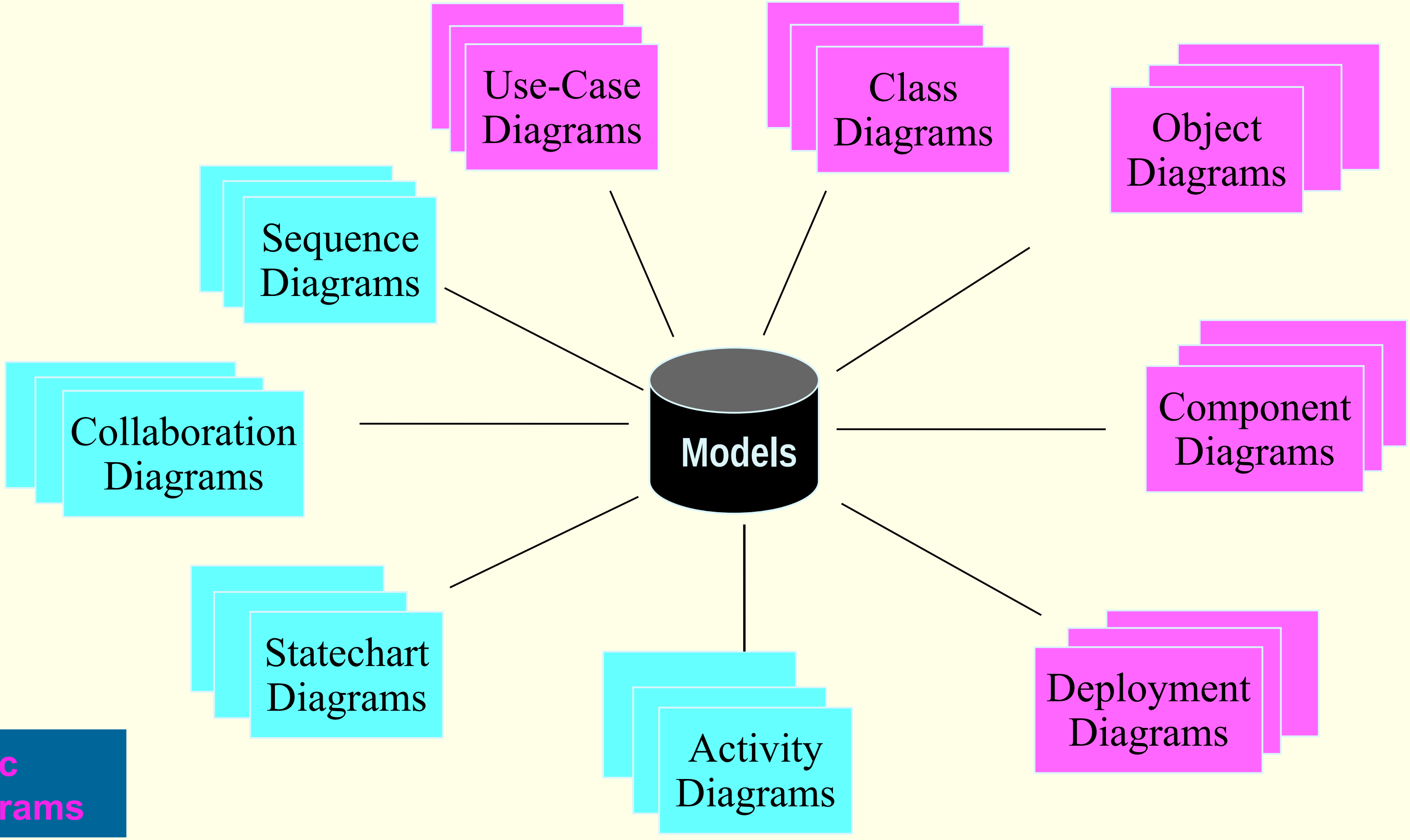
The artifacts of software systems

Why Model Visually?

To:

- ❑ Capture structure and behavior
- ❑ Show how system elements fit together
- ❑ Keep design and implementation consistent
- ❑ Hide or expose details as appropriate
- ❑ Promote unambiguous communication.
- ❑ UML provides one language for all practitioners

Visual Modeling with Unified Modeling Language

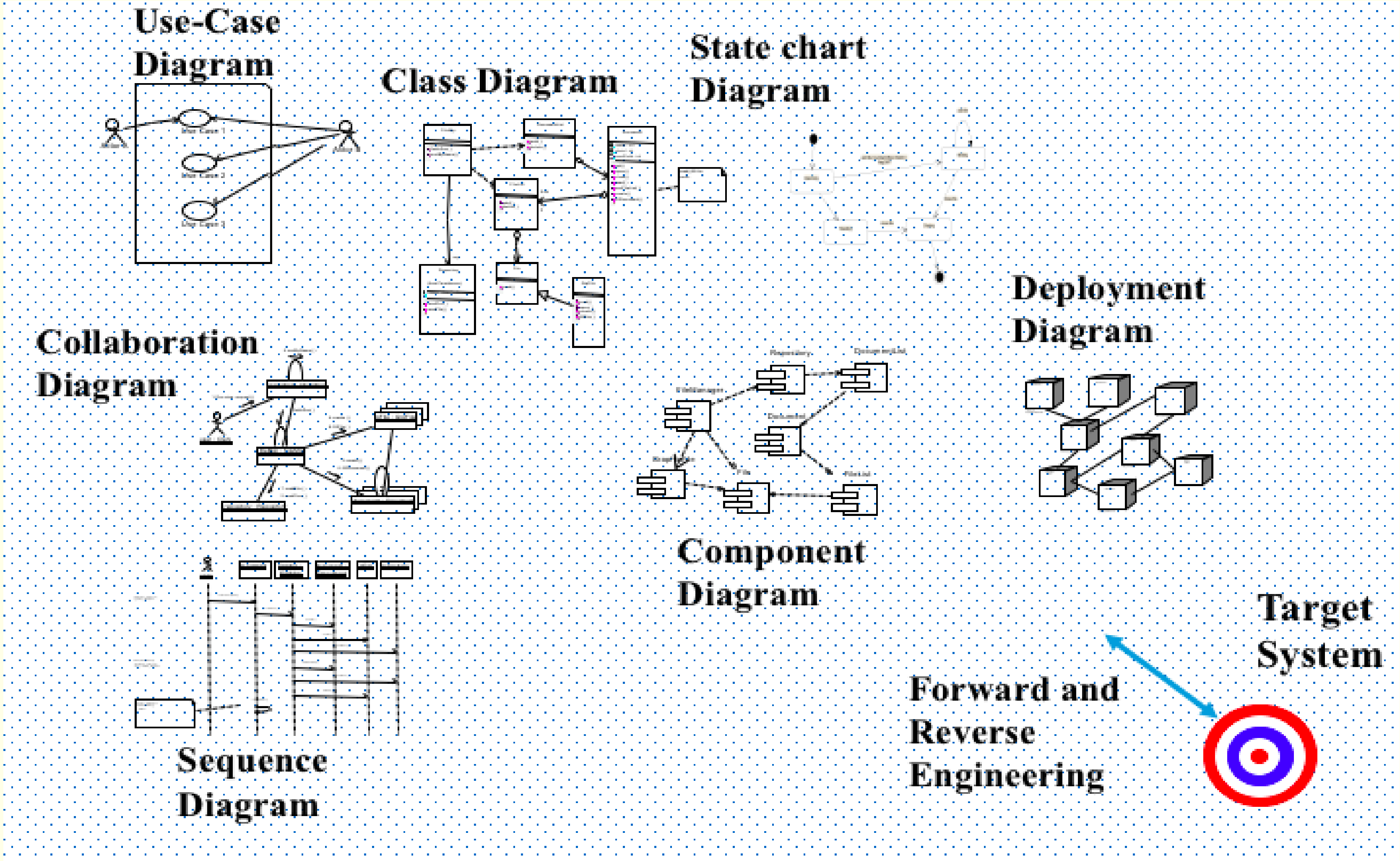


Static Diagrams

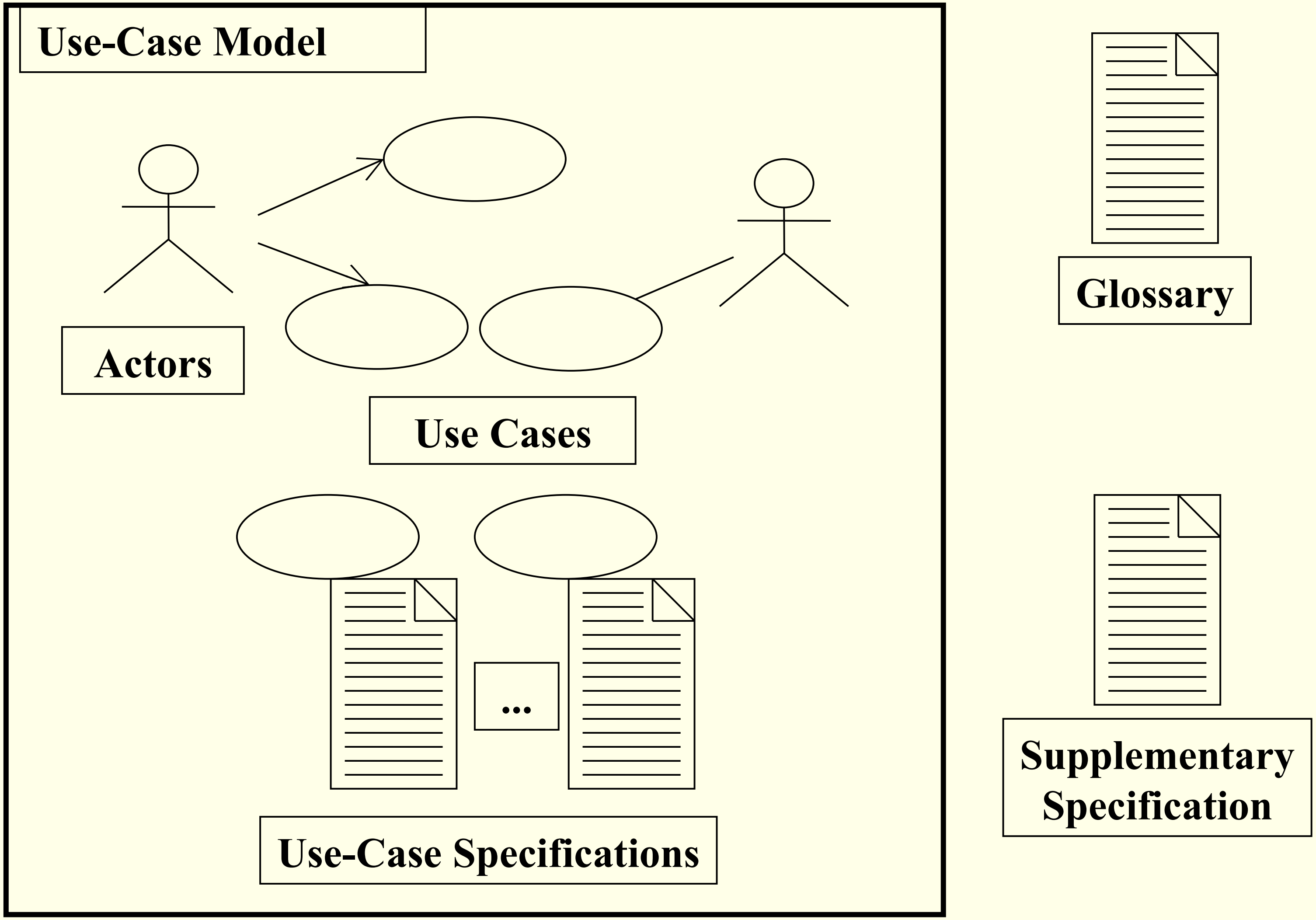
Dynamic Diagrams

- ❑ Multiple views
- ❑ Precise syntax and semantics

Visual Modeling Using UML Diagrams



Relevant Requirements Artifacts



Use-Case Model

Actors

Use Cases

...

Use-Case Specifications

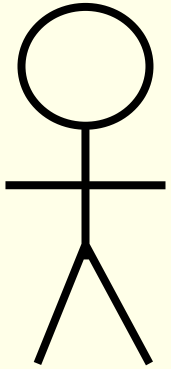
Glossary

Supplementary Specification

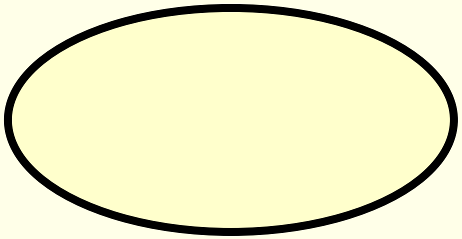
Major Concepts in Use-Case Modeling

- An actor represents anything that interacts with the system.

- A use case defines a set of use-case instances, where each instance is a sequence of actions a system performs that yields an observable result of value to a particular actor.



Actor

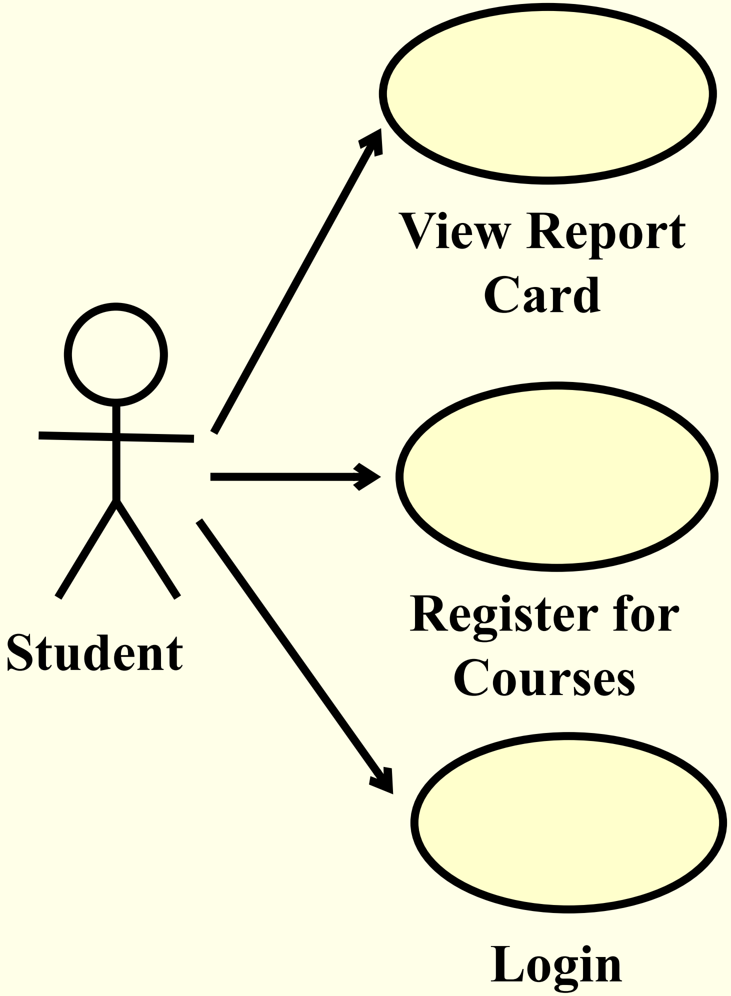


Use Case

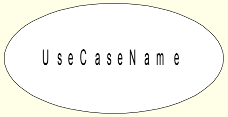
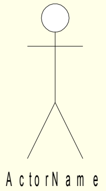
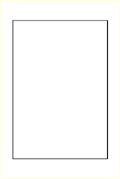
Review: What Is a Use-Case Model?

- ❑ A model that describes a system’s functional requirements in terms of use cases.

- ❑ A model of the system’s intended functionality (use cases) and its environment (actors).



Use Case Modeling: Core Elements

Construct	Description	Syntax
use case	A sequence of actions, including variants, that a system (or other entity) can perform, interacting with actors of the system.	
actor	A coherent set of roles that users of use cases play when interacting with these use cases.	
system boundary	Represents the boundary between the physical system and the actors who interact with the physical system.	

[Continued…]

Use Case Modeling: Core Elements

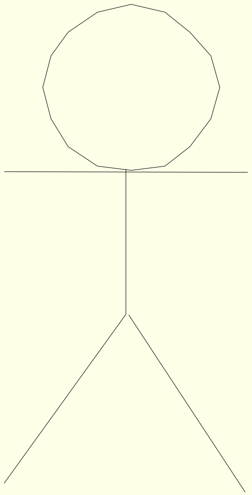
Construct	Description	Syntax
association	The participation of an actor in a use case. i.e., instance of an actor and instances of a use case communicate with each other.	—
extend	A relationship from an *extension* use case to a *base* use case, specifying how the behavior for the extension use case can be inserted into the behavior defined for the base use case.	→ <<extend>>
Include	An relationship from a base use case to an inclusion use case, specifying how the behavior for the inclusion use case is inserted into the behavior defined for the base use case.	→ <<include>>

Review: Benefits of Use-Case Models

- ❑ Used to communicate with the end users and domain experts
  - Insures a mutual understanding of the requirements
- ❑ Used to identify
  - Who interacts with the system and what the system should do
  - The interfaces the system should have
- ❑ Used to verify
  - All requirements have been captured
  - The development team understands the requirements

Useful Questions in Finding Actors

- ❑ Who will supply, use, or remove information?
- ❑ Who will use this functionality?
- ❑ Who is interested in a certain requirement?
- ❑ Where in the organization is the system used?
- ❑ Who will support and maintain the system?
- ❑ What are the system's external resources?
- ❑ What other systems will need to interact with this one?



Actor

Name and Describe Each Actor

- ❑ Actor names should clearly denote the actor’s role
- ❑ Actor description:
  - Area of responsibility
  - Dependency of the actor on the system

Finding Use Cases:
Focus on the Actor

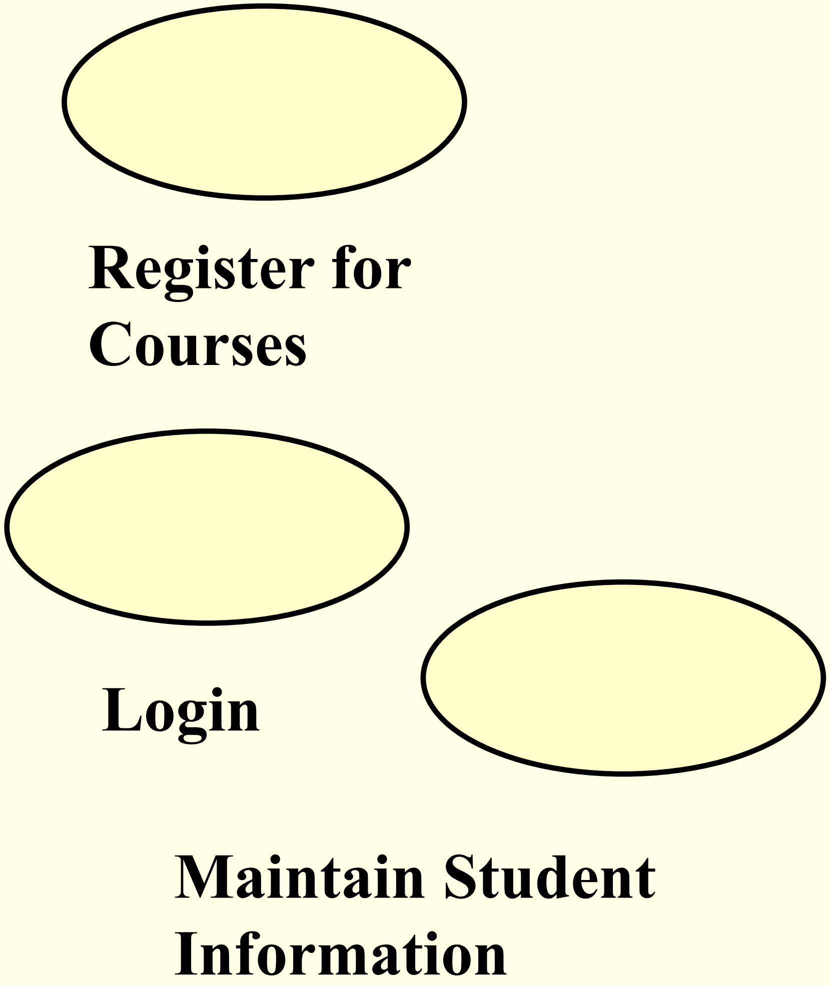
- ❑ The system exists only for its users.
- ❑ Use cases should be based on the user’s needs.

Useful Questions in Finding Use Cases

Answer the following questions to find use cases.

- ❑ For each actor you have identified, what are the tasks the system would be involved in?
- ❑ Does the actor need to be informed about certain occurrences in the system?
- ❑ Will the actor need to inform the system about sudden, external changes?
- ❑ What information must be modified or created in the system?

Naming the Use Case



Register for Courses

Login

Maintain Student Information

- □ The name indicates what is achieved by its interactions with the actor(s).
- □ The name may be several words in length.
- □ No two use cases should have the same name.

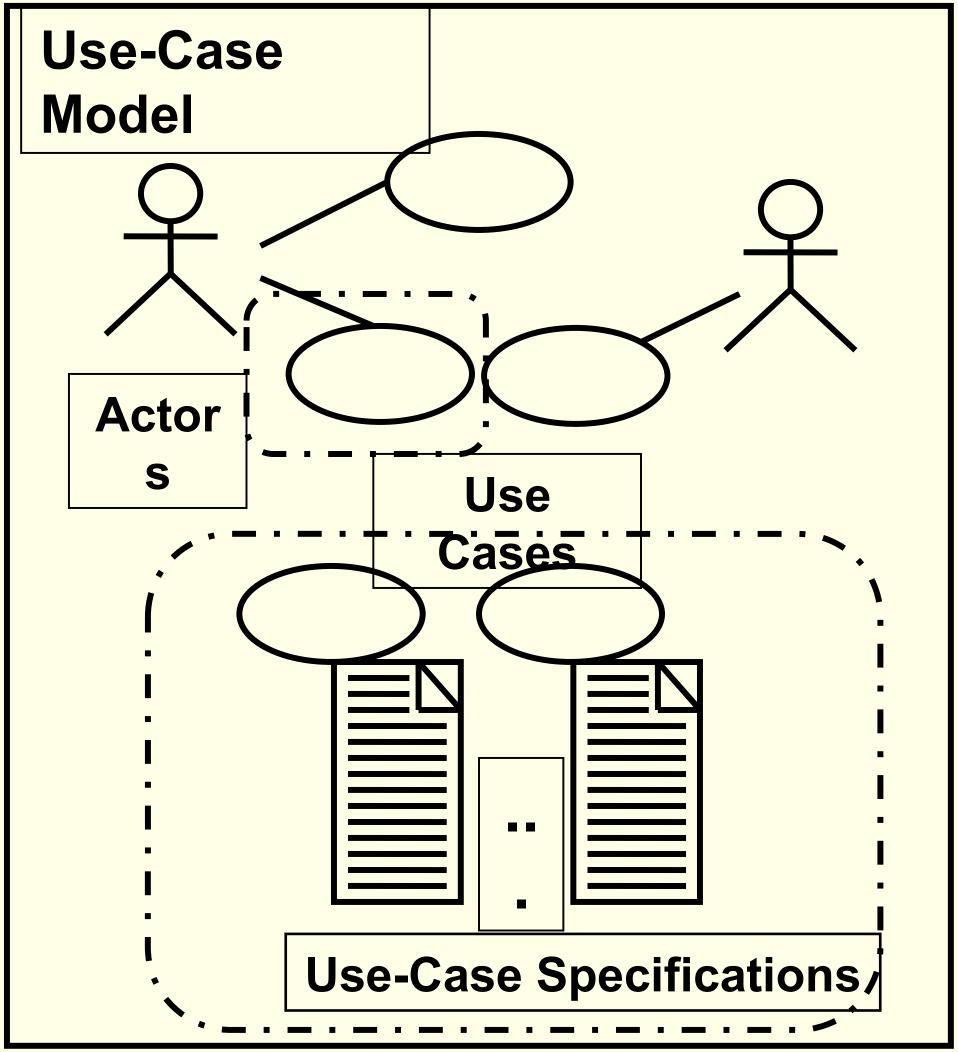
Use Cases and Actors

- A use case models a dialog between actors and the system.

- A use case is initiated by an actor to invoke a certain functionality in the system.

Use-Case Specifications

- ❑ Name
- ❑ Brief description
- ❑ Flows of Events
- ❑ Relationships
- ❑ Use-Case diagrams
- ❑ Special requirements
- ❑ Preconditions
- ❑ Post conditions



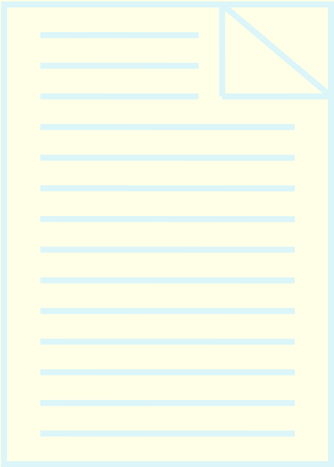
Use Case Flow of Events

- Has one typical/ basic flow or multiple flows
- Several alternative flows
  - Regular variants of the basic flow
  - Odd cases
  - Exceptional flows handling error situations

Supplementary Specification

❑ The nonfunctional requirements and functional requirements not captured by the use cases are included in the supplementary specifications. The Supplementary Specifications include constraints on the implementation.

- Functionality
- Usability
- Reliability
- Performance
- Supportability
- Design constraints



Supplementary Specification

Checkpoints: Use-Case Model

- Is the use-case model understandable?

- By studying the use-case model, can you form a clear idea of the system's functions and how they are related?

- Have all functional requirements been met?

- Is the division of the model into use-case packages appropriate?

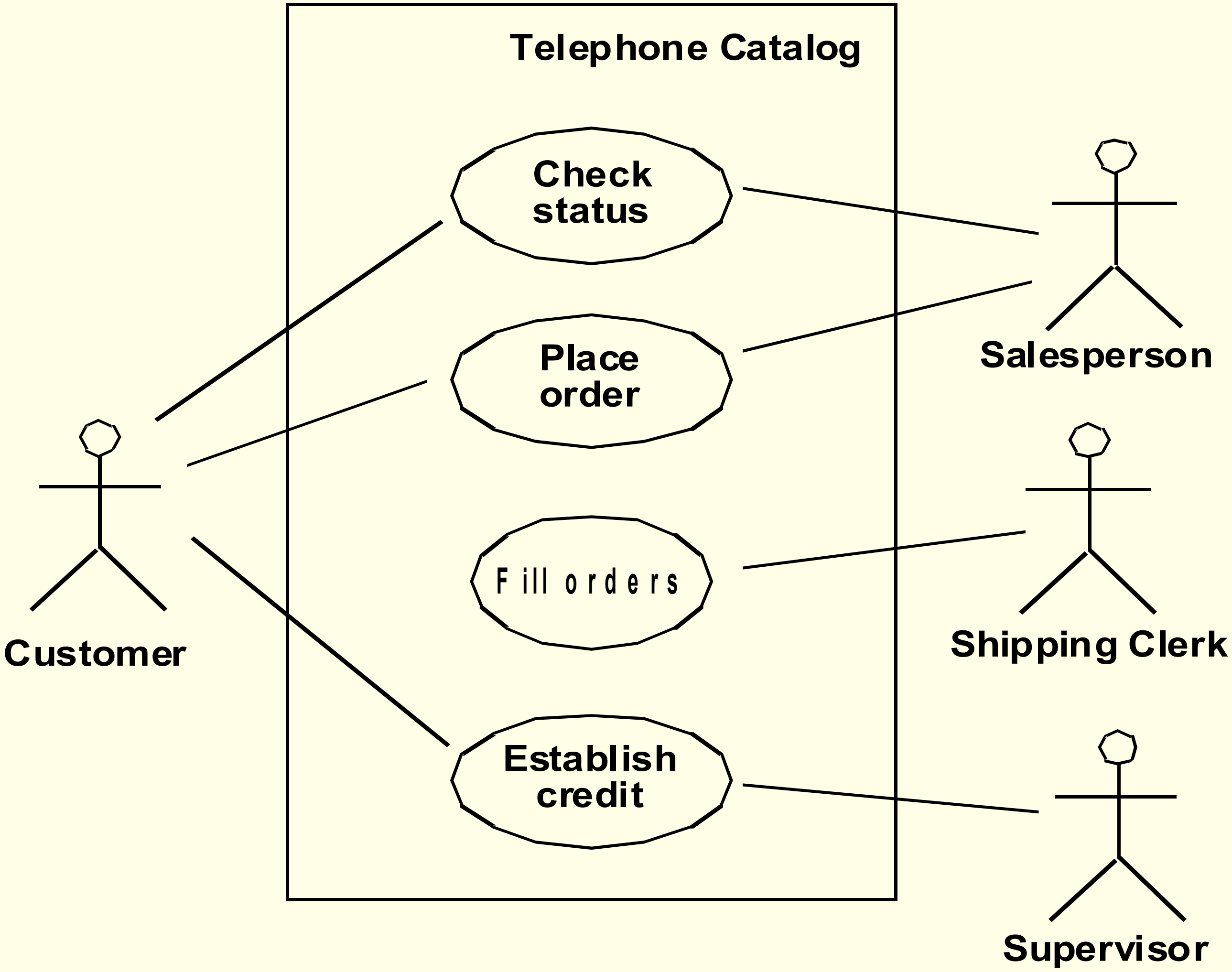
Checkpoints: Actors

- Have all the actors been identified?
- Is each actor involved with at least one use case?
- Is each actor really a role? Should any be merged or split?
- Do two actors play the same role in relation to a use case?
- Do the actors have intuitive and descriptive names? Can both users and customers understand the names?

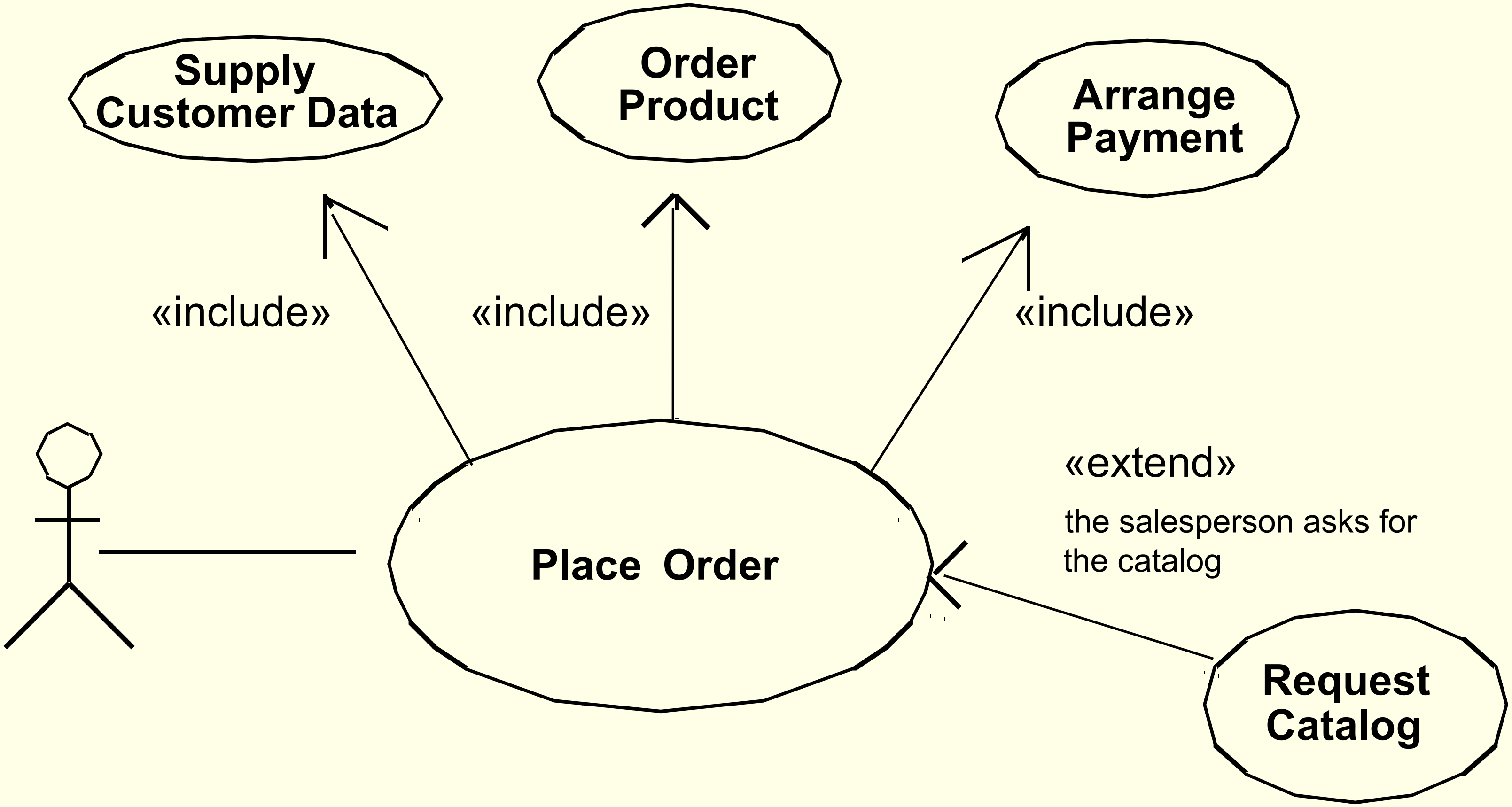
Checkpoints: Use Cases

- ❑ Is each use case involved with at least one actor?
- ❑ Is each use case independent of the others?
- ❑ Do any use cases have very similar behaviors or flows of events?
- ❑ Do the use cases have unique, intuitive, and explanatory names so that they cannot be mixed up at a later stage?
- ❑ Do customers and users understand the names and descriptions of the use cases?

Use Case Diagram Example



Use Case Relationships



Use Case Guidelines

- ❑ Draw a use case diagram
- ❑ Whenever you need to describe use cases in natural language, provide the following information for each one.
  - **Use case number**: a unique number for referencing UC elsewhere in the specification; use cases are numbered " UC1", " UC2", etc.
  - **Use Case Name**: a name indicating what is captured by UC
  - **Authors**: the names of the persons who discovered UC

[Continued…]

Use Case Guidelines Continues

- **Actors**: a list of the actors that participate in UC
- **Overview**: a brief 2- 3 sentence description of UC; this overview serves also as a high- level description of UC.
- **References**: a list of the numbers of all requirements captured by UC .
- **Related Use Cases**: a list of the numbers of all related use cases; for each related use case describe the relationship of the identified use case to UC

[Continued…]

Use Case Guidelines Continues

- Precondition & Postcondition.

- **Typical Flow Description**: description of the most usual instance scenario of UC that leads to the successful outcome of the process that UC captures.

- **Alternative flow**: alternative flow are described here for the UC

Example of Use Case

- **Use Case Nam**e: Buying a Book Online
- **Use Case Numbe**r: UC11
- **Author**s: J. Jones
- **Actor**s: Customer (initiator), credit- card authorization service, book seller

Example of Use Case

- **Overview**: This use case captures the process of purchasing one or more books from an online book seller
- **References**: R19, R4, and R5.
- **Related Use Cases**: UC7
- **Typical Flow Description**: (include precondition & post-condition)
- **Alternative Flow Description: (**include precondition & post-condition)

Next:

More on UML and Use Case Modeling