SWE3002-42: Introduction to Software Engineering

Lecture 5 – System Modeling

Sooyoung Cha

Department of Computer Science and Engineering

Topics covered

01 System Modeling

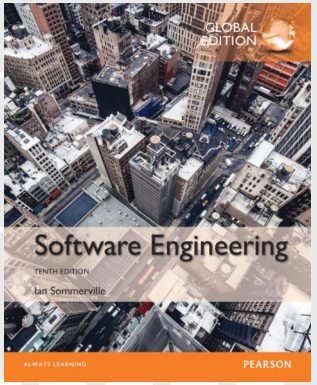
02 Context Models

03 Interaction Models

04 Structural Models

05 Behavioral Models

06 Model-driven Engineering

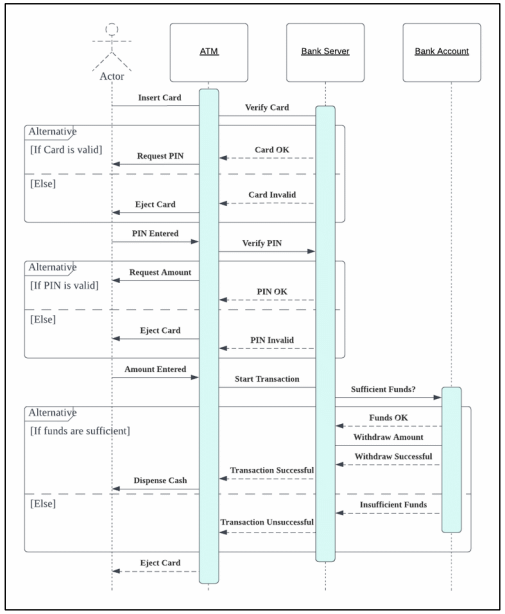


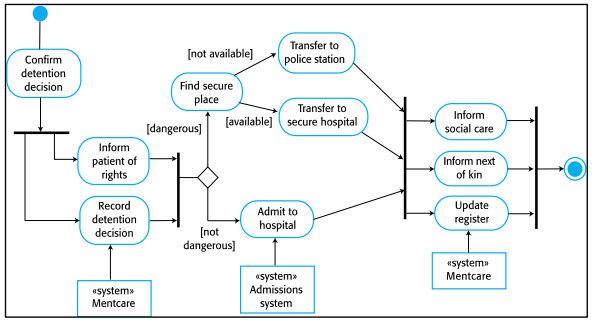
Chapter 5. (p.138 ~ p.166)

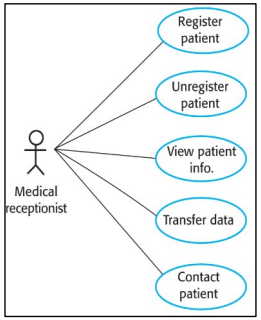
System Modeling

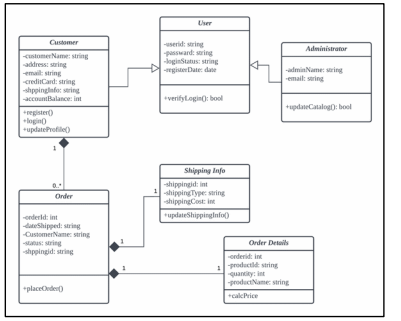
[System Modeling]

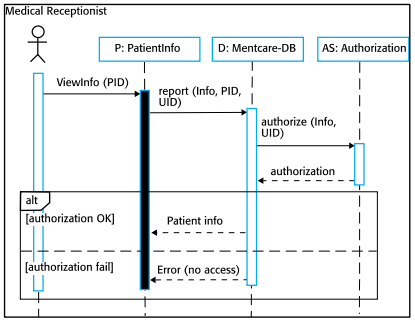
- A process of developing abstract models of a system.
  - Each model presents a different view or perspective of the system.
  - Each model usually represent a system by using diagram types in UML.

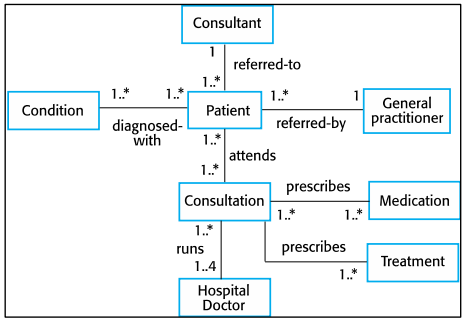


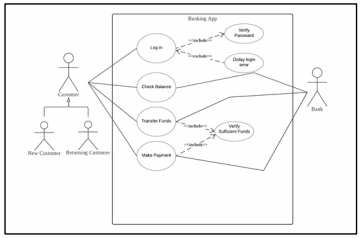












System Modeling

[Models of both the existing system and the new system]

- Models of the existing system.
  - Clarifying what the existing system does.
  - Discussing the strengths and weaknesses of the existing system.

- Models of the new system.
  - Explaining the proposed requirements to other system stakeholders.
  - Discussing design proposals.
  - Documenting the system for implementation.

System Modeling

[Different perspectives for representing the system]

- An external perspective
  - Modeling the context or environment of the system.
- An interaction perspective
  - Modeling the interactions between a system and its environment.
- A structural perspective
  - Modeling the organization of a system or the structure of its data.
- A behavioral perspective
  - Modeling the dynamic behavior of the system.

System Modeling

[Different ways to use graphical models]

- Focusing discussion about an existing or proposed system.
  - The models may be incomplete and use informal notation.
- Documenting an existing system.
  - The models do not have to be complete, but correct.
- Generating a system implementation.
  - The models do have to be complete and correct.

UML diagrams

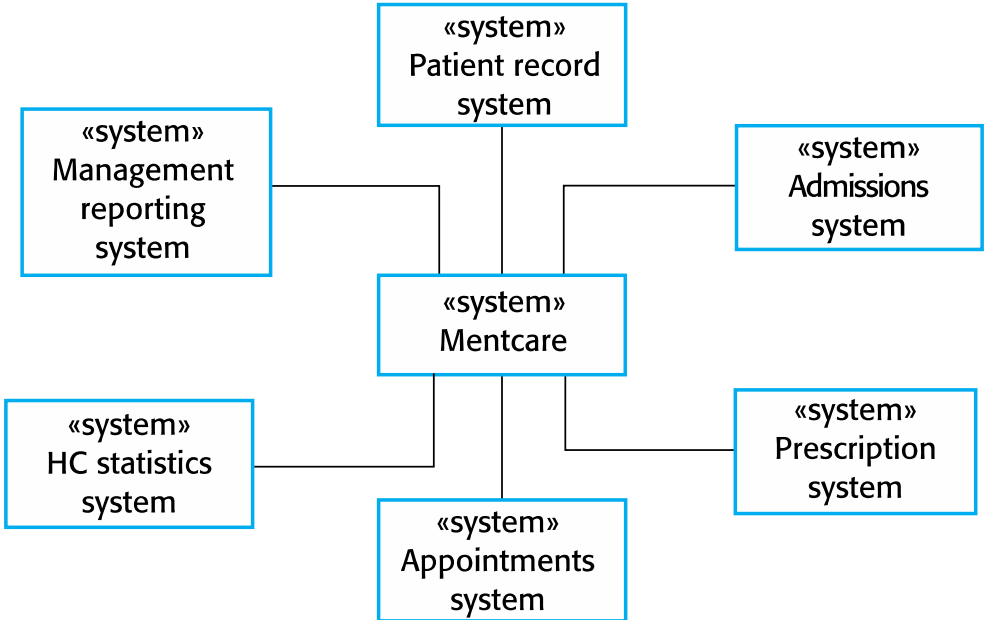
[UML: A standard language for object-oriented modeling]

Use case diagrams	which show the interactions between a system and its environment.
Sequence diagrams	which show interactions between actors and the system and between system components.
Class diagrams	which show the object classes in the system and the associations between these classes.
Activity diagrams	which show the activities involved in a process or in data processing.
State diagrams	which show how the system reacts to internal and external events.

Context Models

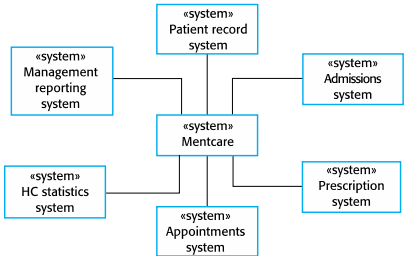
[Roles of Context models]

- Showing how a system is positioned in an environment with other systems and processes.



"A context model of Mentcare system and the other systems"

Context Models



[Simple Context models]

- Do not show the types of relationships between the systems in the environment and the system that is being specified.
  - Producing data for or consuming data from the system.
  - Physical locations of the systems (e.g., same building)

↓

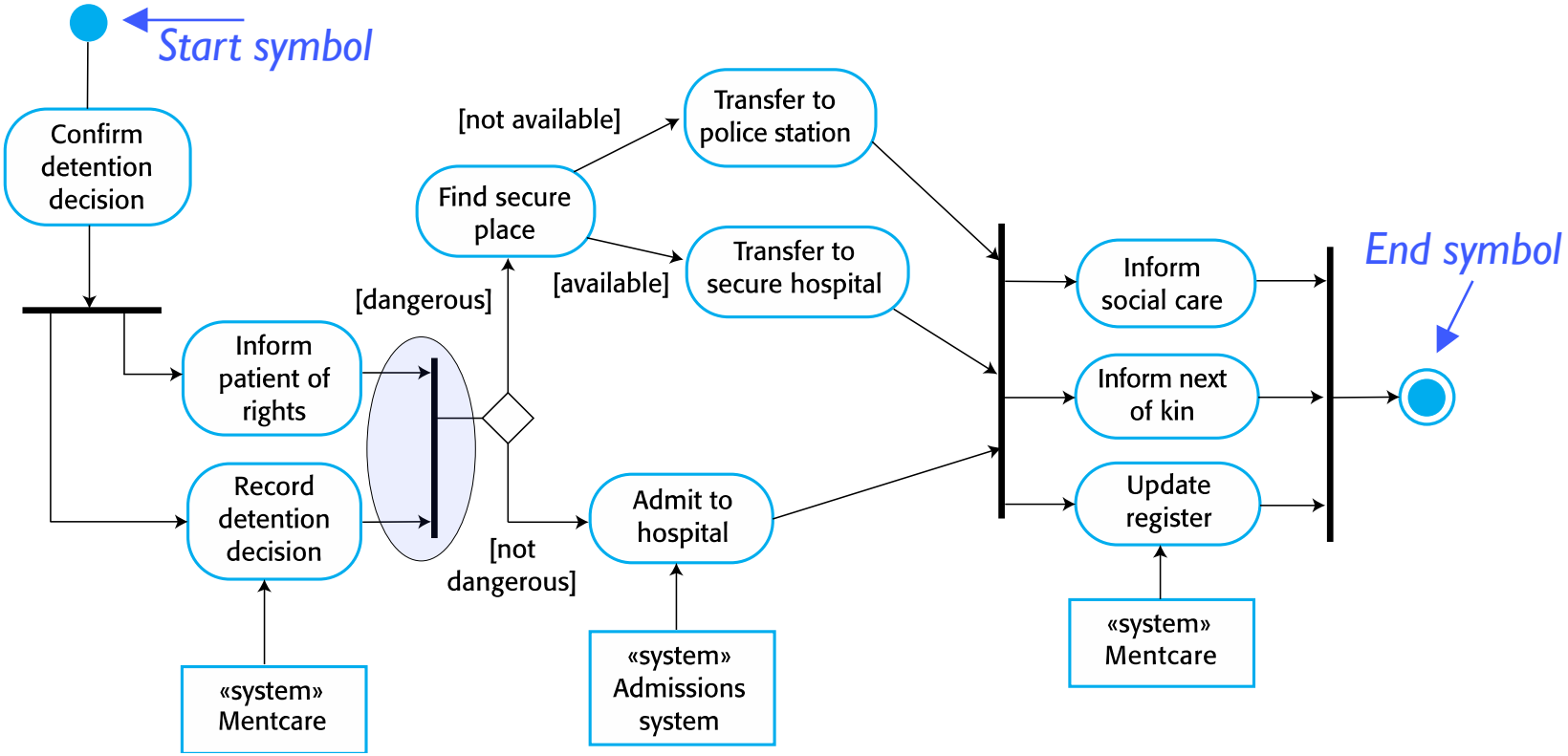
- The relations affect the requirements and design of the system.

↓

- Simple context models are used along with other models, such as business process models.

Context Models

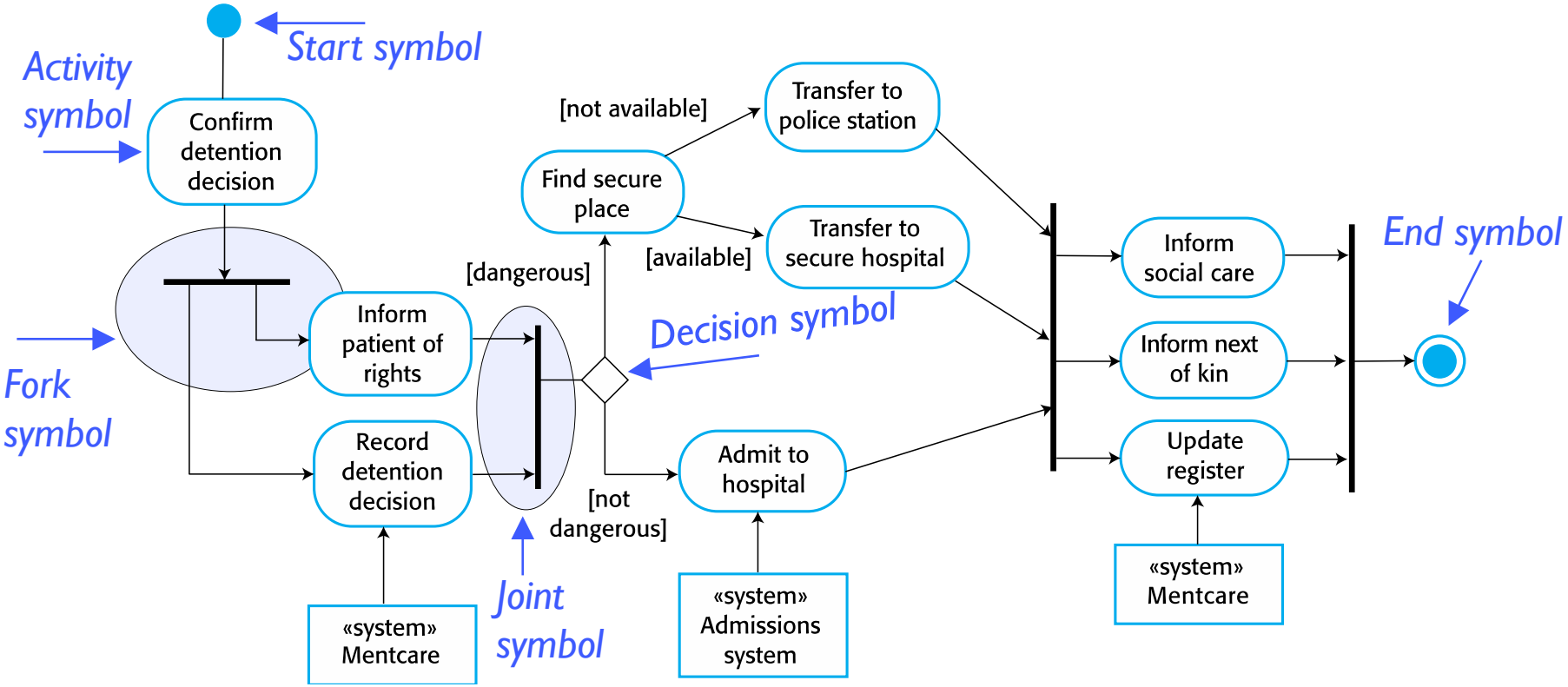
[Simple Context model + Activity model]



"A process model of involuntary detention"

Context Models

[Simple Context model + Activity model]



"A process model of involuntary detention"

Interaction Models

[Interaction Models]

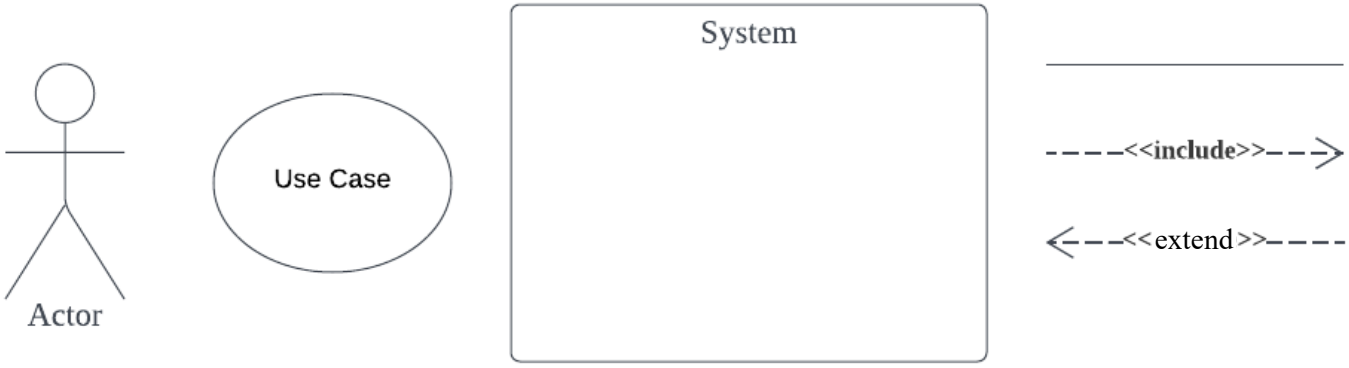
- Use Case Modeling.
  - Modeling interactions between a system and external agents (human users or other systems).
- Sequence diagram.
  - Modeling interactions between system components, although external agents may also be included.

Use case diagrams

[Roles of use case diagrams]

- Simply describing what a user expects from a system in the interaction.
- Giving a simple overview of an interaction.

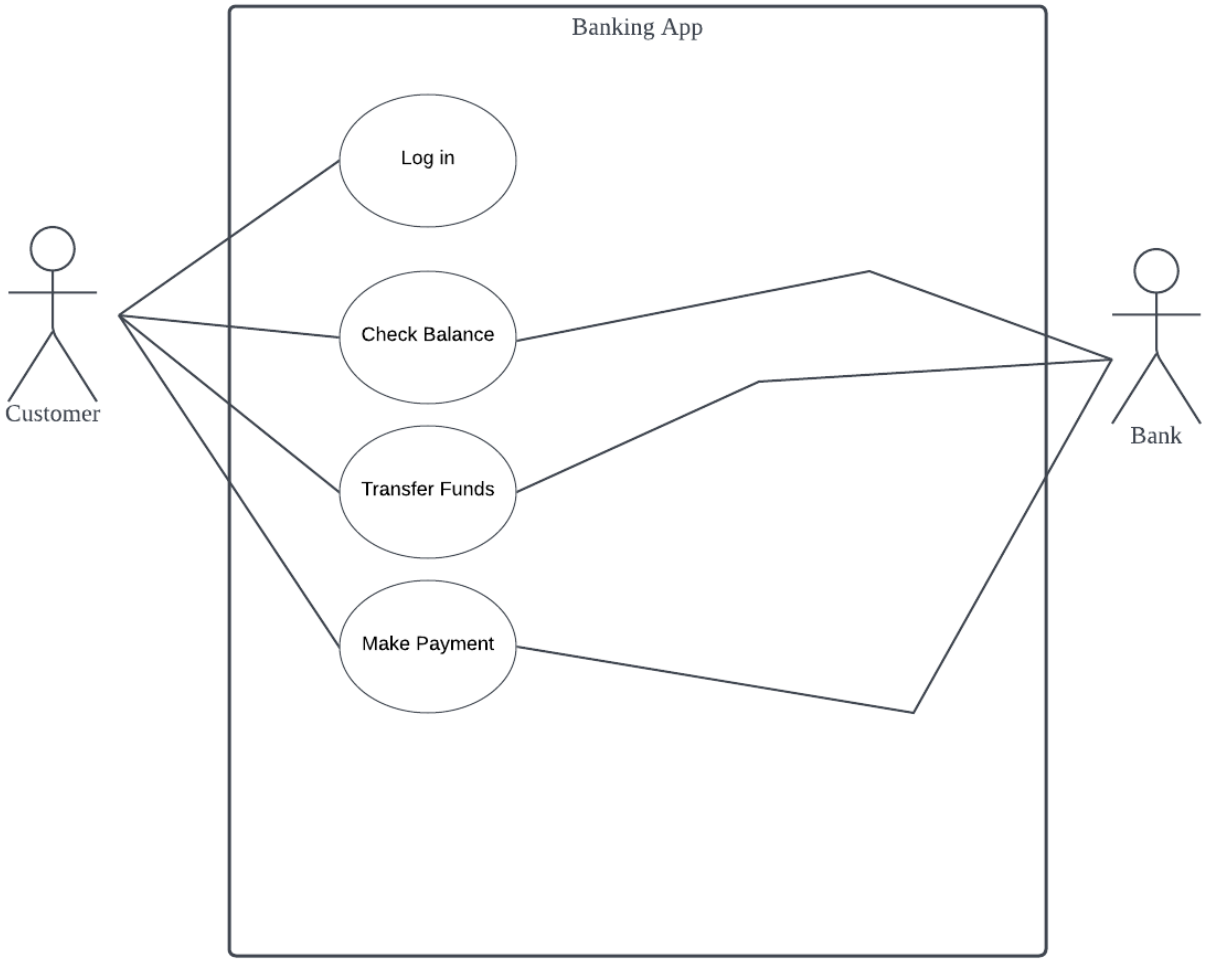
Actor	Use Case	System	Relationship



Use case diagrams

[Example of use case diagrams]

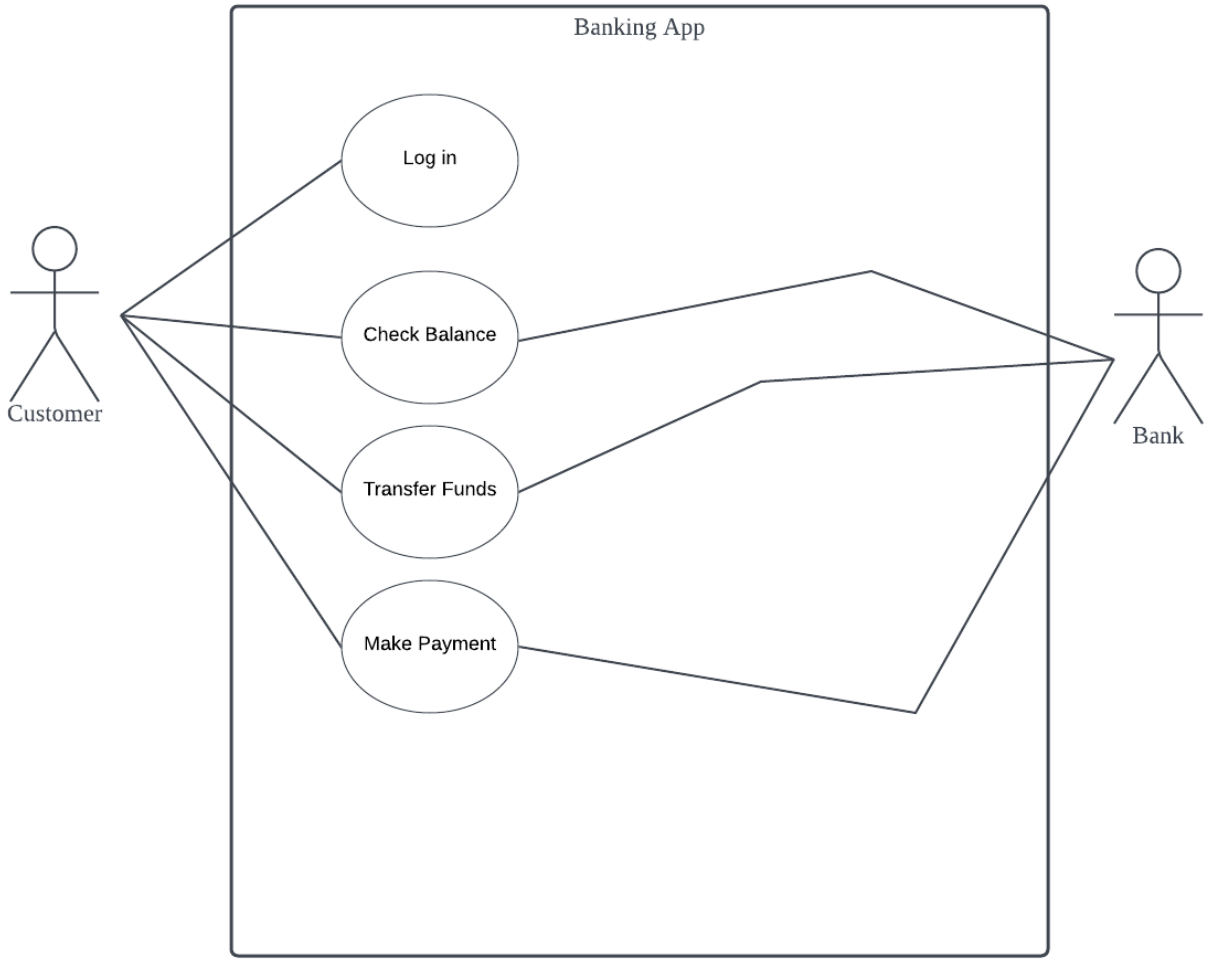
- Actor
  - Primary actors
  - Secondary actors
- System



Use case diagrams

[Example of use case diagrams]

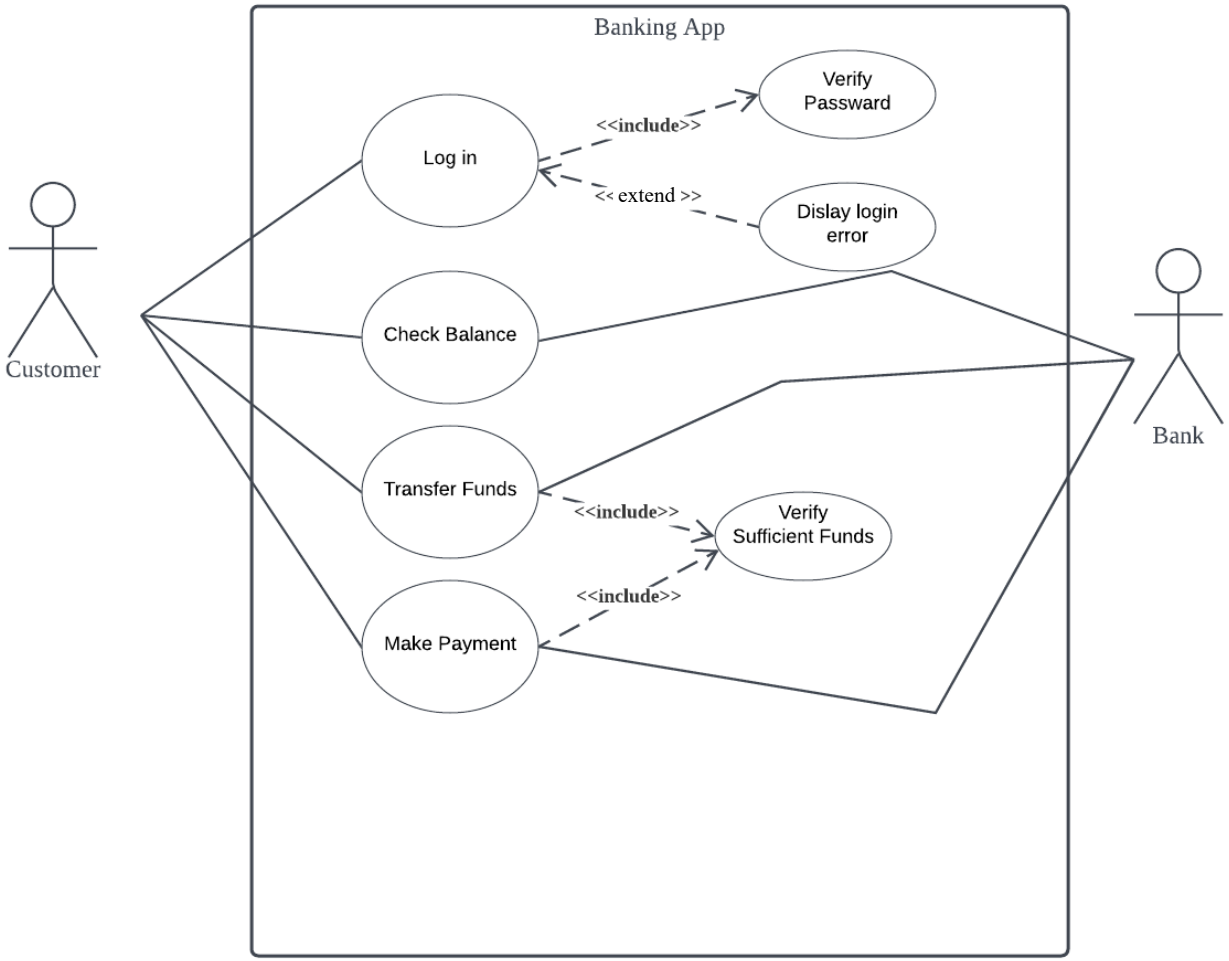
- Relationship
  - Association



Use case diagrams

[Example of use case diagrams]

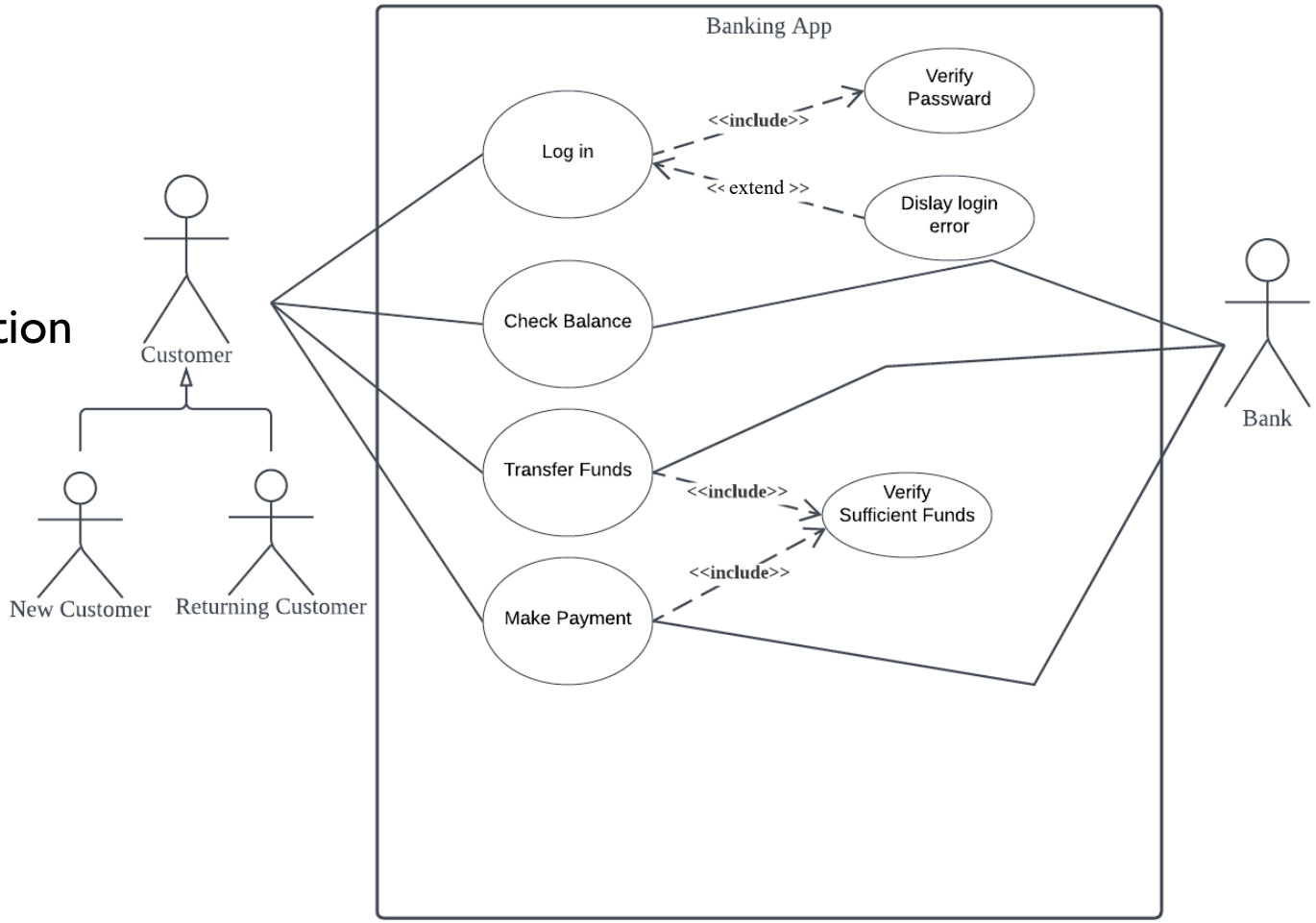
- Relationship
  - Include
  - Extend



Use case diagrams

[Example of use case diagrams]

- Relationship
  - Include
  - Extend
  - Generalization



Use case diagrams

[Tabular description of the Transfer-data use case]

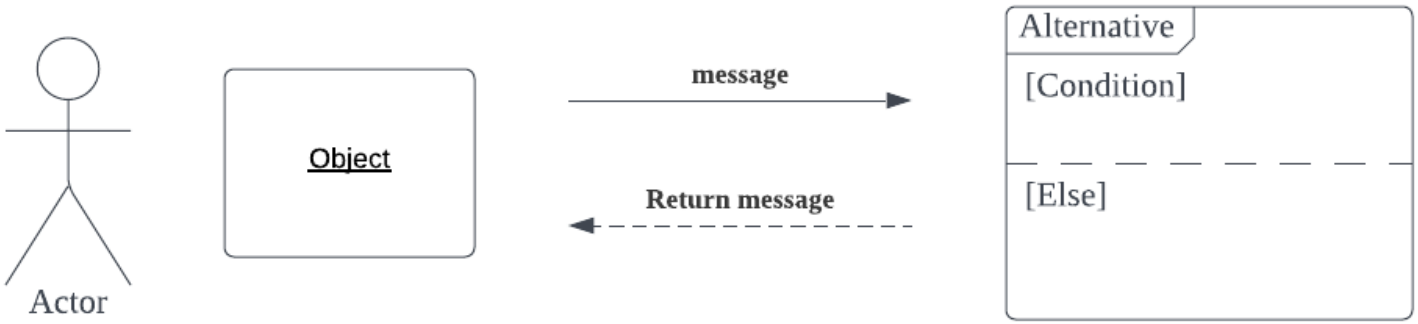
Mentcares system: Transfer data	
Actors	Medical receptionist, patient records system (PRS)
Description	A receptionist may transfer data from the Mentcare system to a general patient record database that is maintained by a health authority. The information transferred may be updated personal information or a summary of the patient's diagnosis and treatment.
Data	Patient's personal information, treatment summary
Stimulus	User command issued by medical receptionist
Response	Confirmation that PRS has been updated
Comments	The receptionist must have appropriate security permissions to access the patient information and the PRS.

Sequence diagrams

[Roles of sequence diagrams]

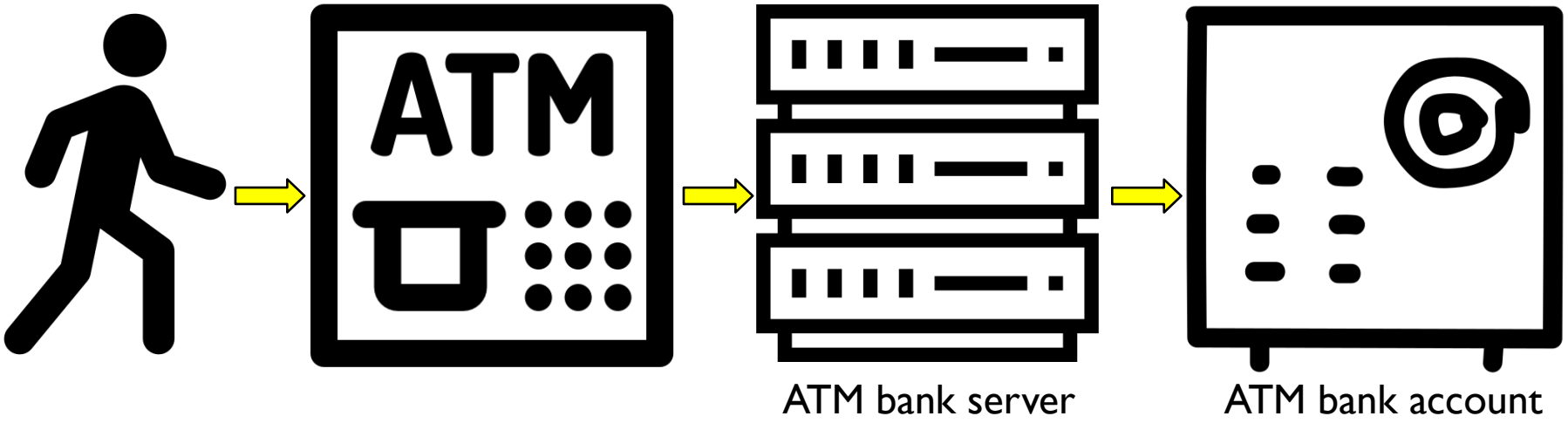
- Modeling the interactions between the actors and the objects in a system and the interactions between the objects.

- Showing the sequence of interactions that take place during a particular use case.

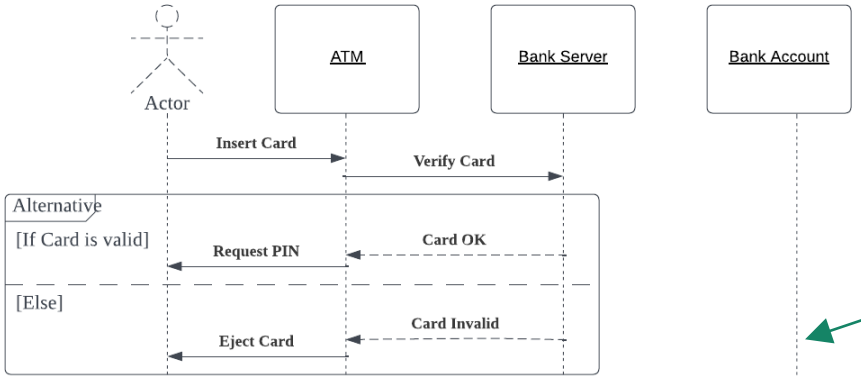


Sequence diagrams

[ATM system of sequence diagrams]



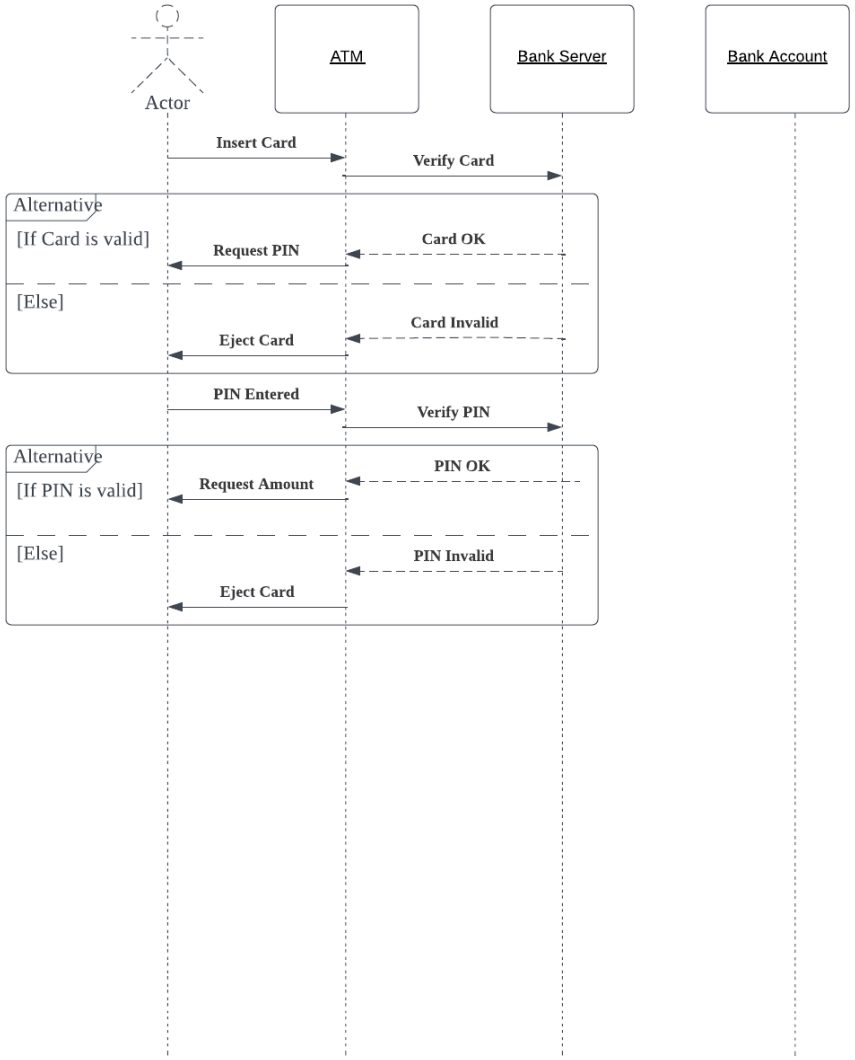
Sequence diagrams



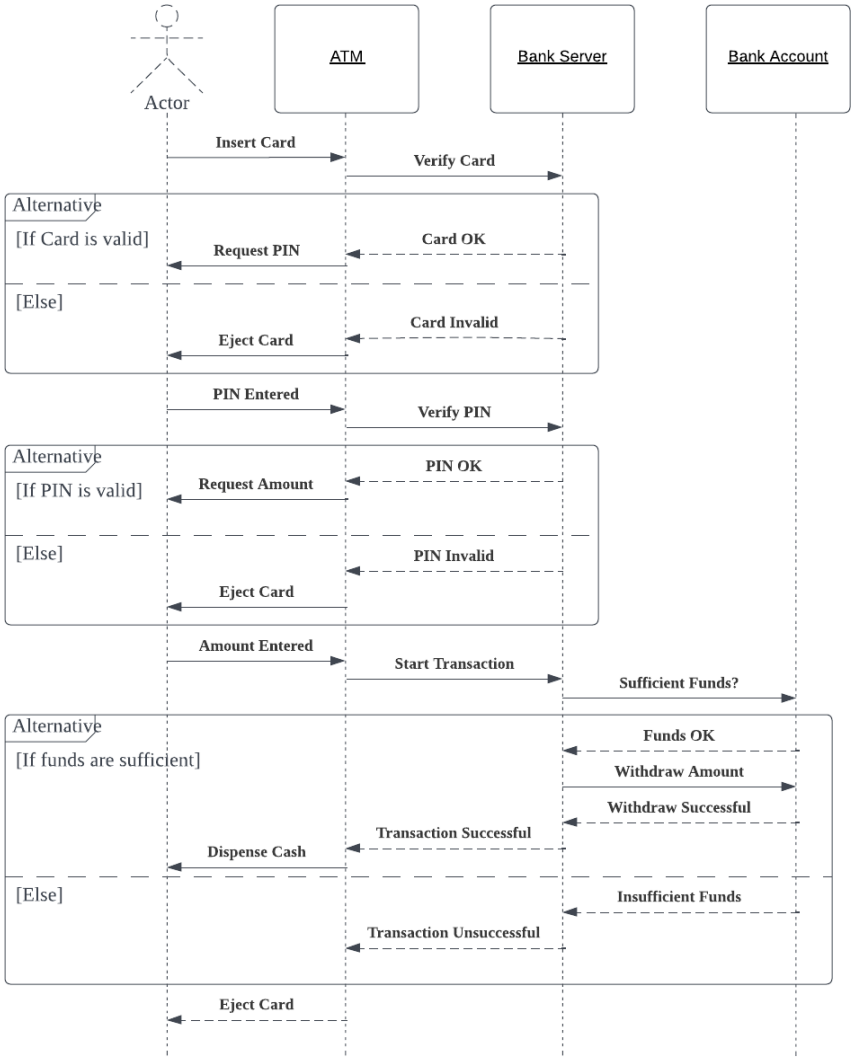
Lifeline

More time is passing

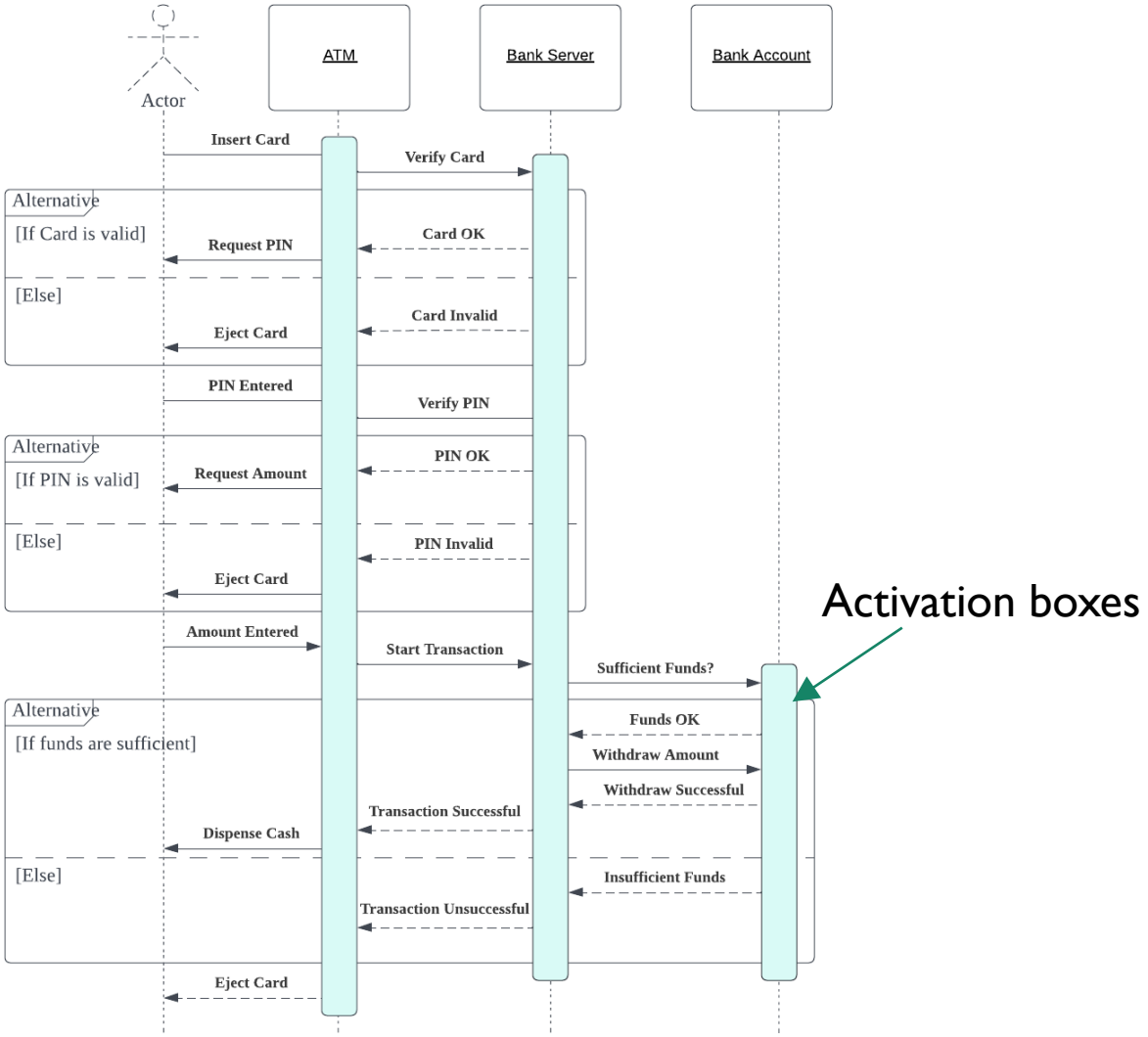
Sequence diagrams



Sequence diagrams



Sequence diagrams



Structural models

[Structural models]

- Showing the organization of a system in terms of the components that make up that system and their relationships.

[Class diagrams]

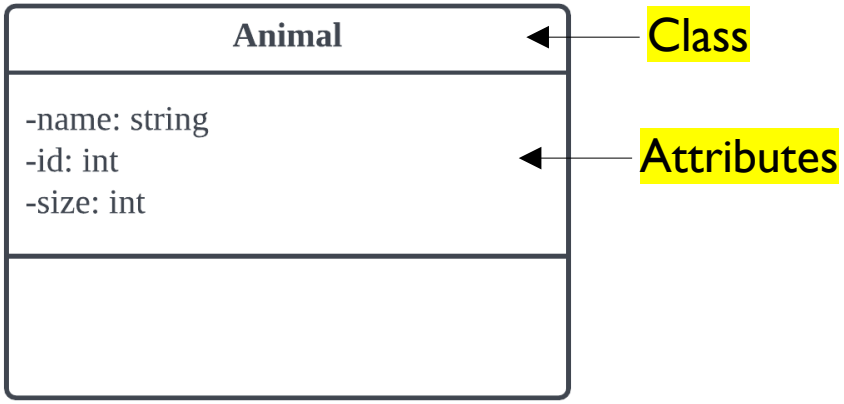
- Modeling the static structure of the object classes in a SW system.

- Developing an object-oriented system model to show the classes in a system and the associations between these classes.

Structural models

[Roles of class diagrams]

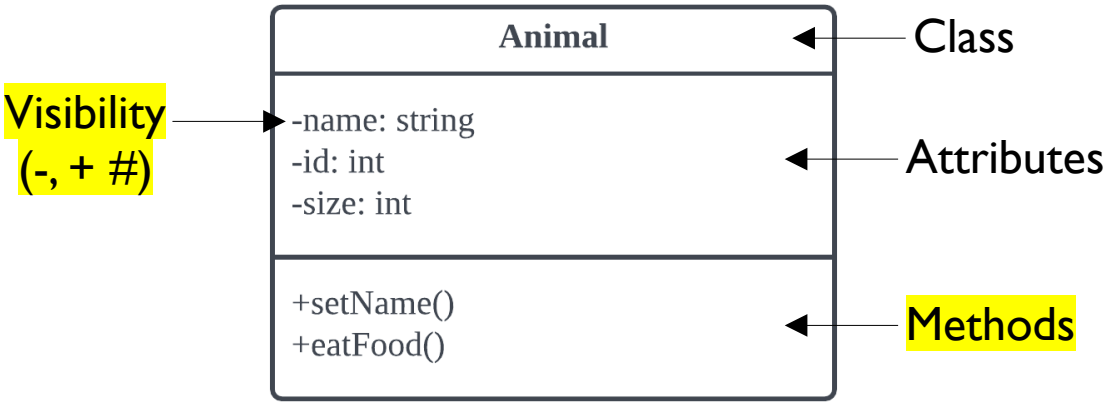
- Consist of class, attributes, methods, relationships.
  - ex) Zoo system.



Structural models

[Roles of class diagrams]

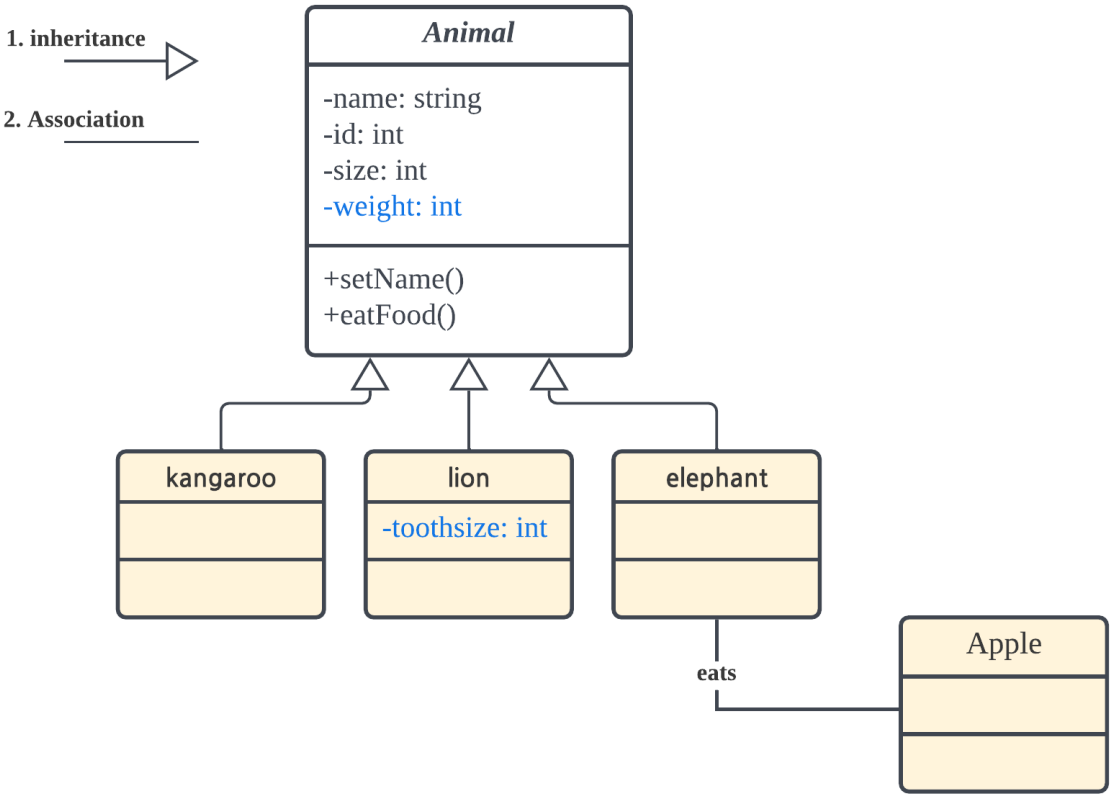
- Consist of class, attributes, methods, relationships.
  - ex) Zoo system.



Structural models

[Roles of class diagrams]

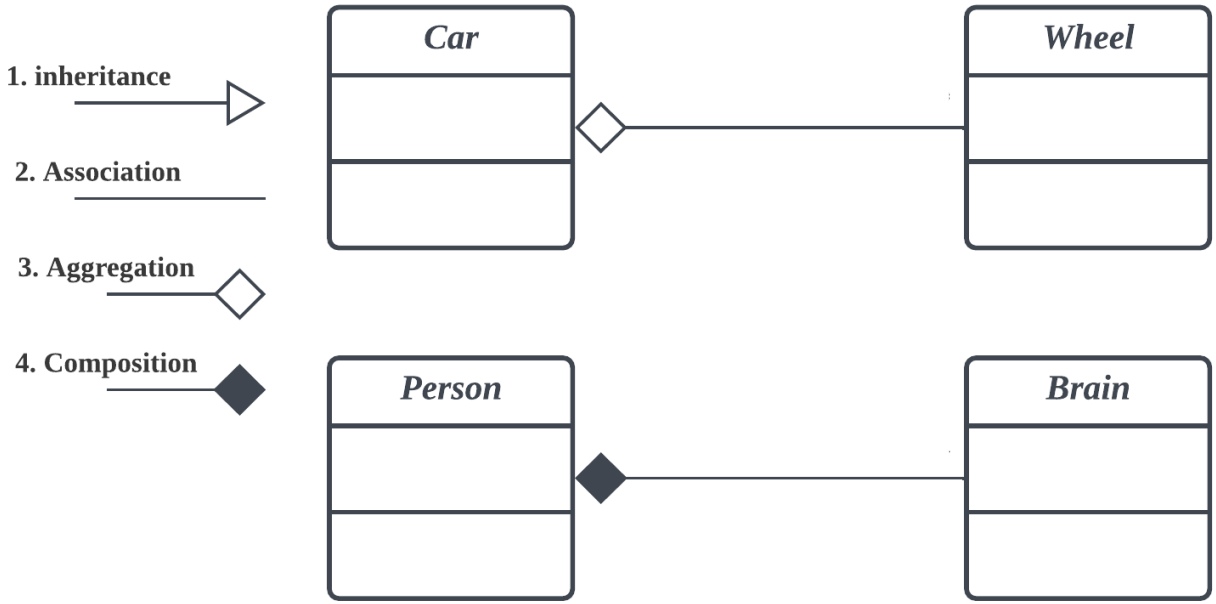
- Consist of class, attributes, methods, relationships.



Structural models

[Roles of class diagrams]

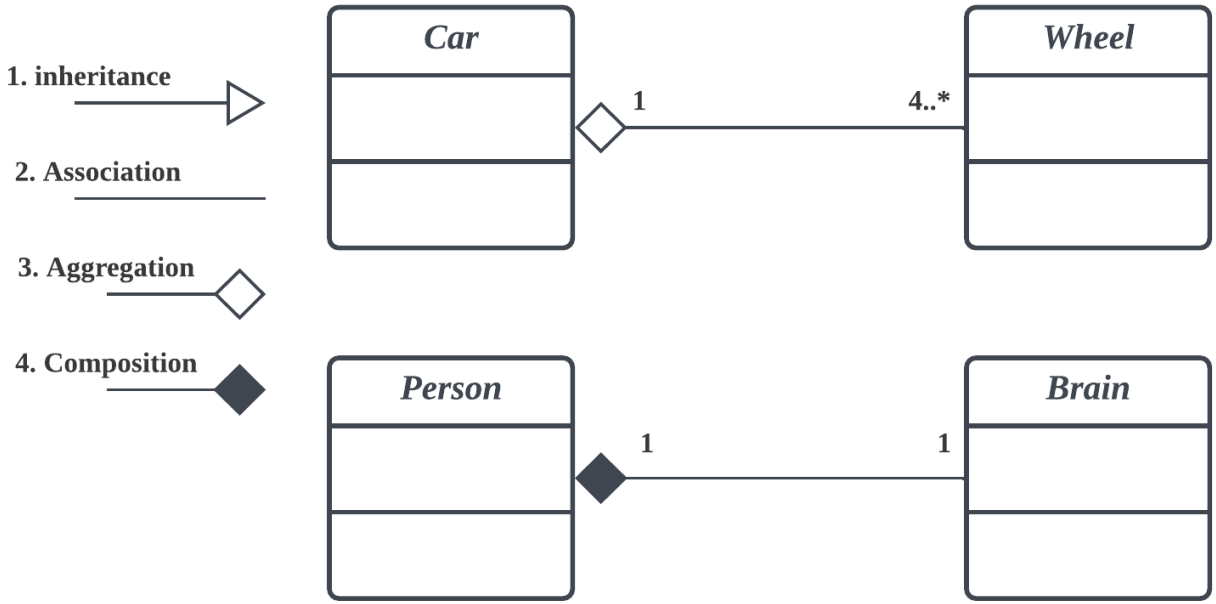
- Consist of class, attributes, methods, relationships.
  - Aggregation, Composition,



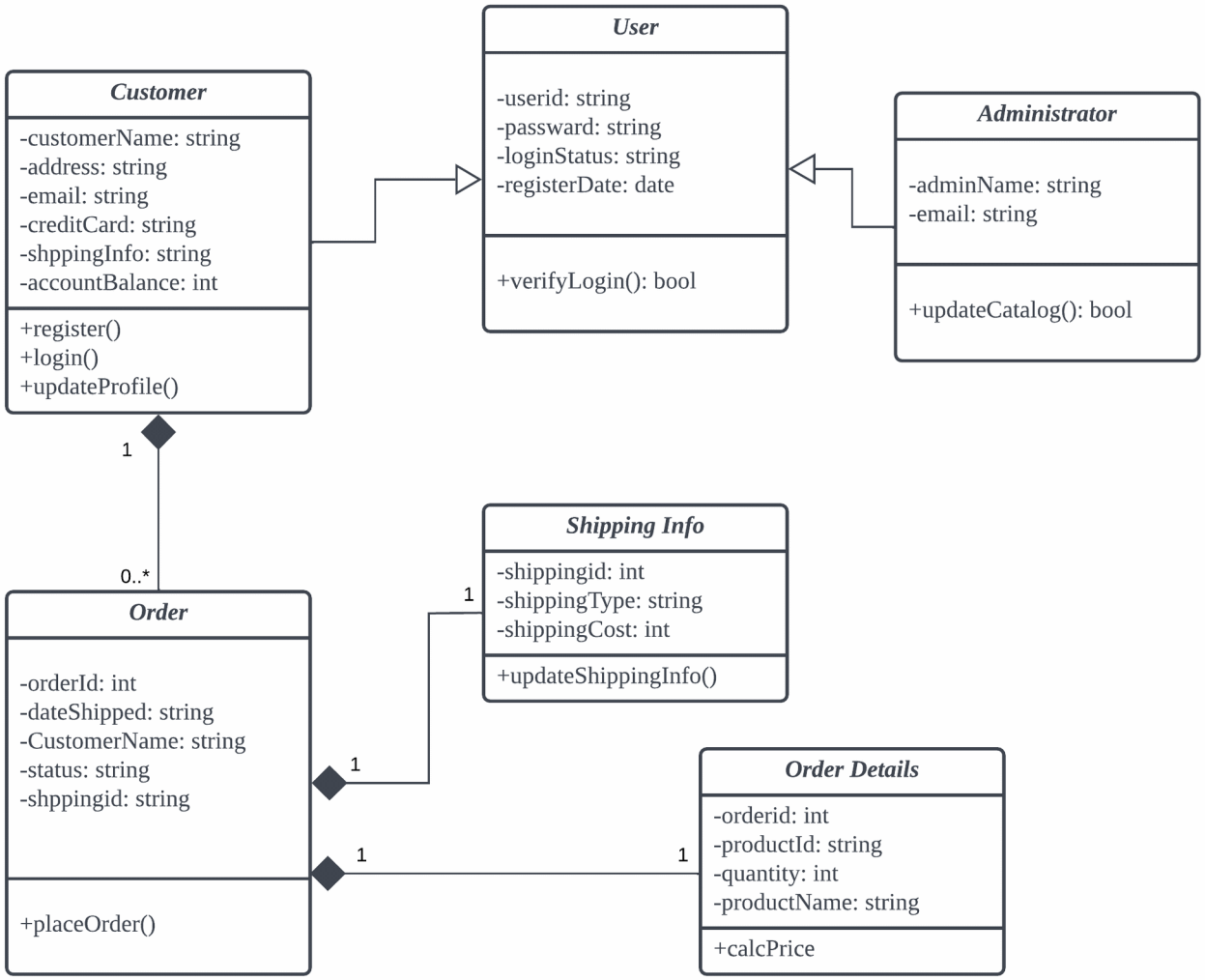
Structural models

[Roles of class diagrams]

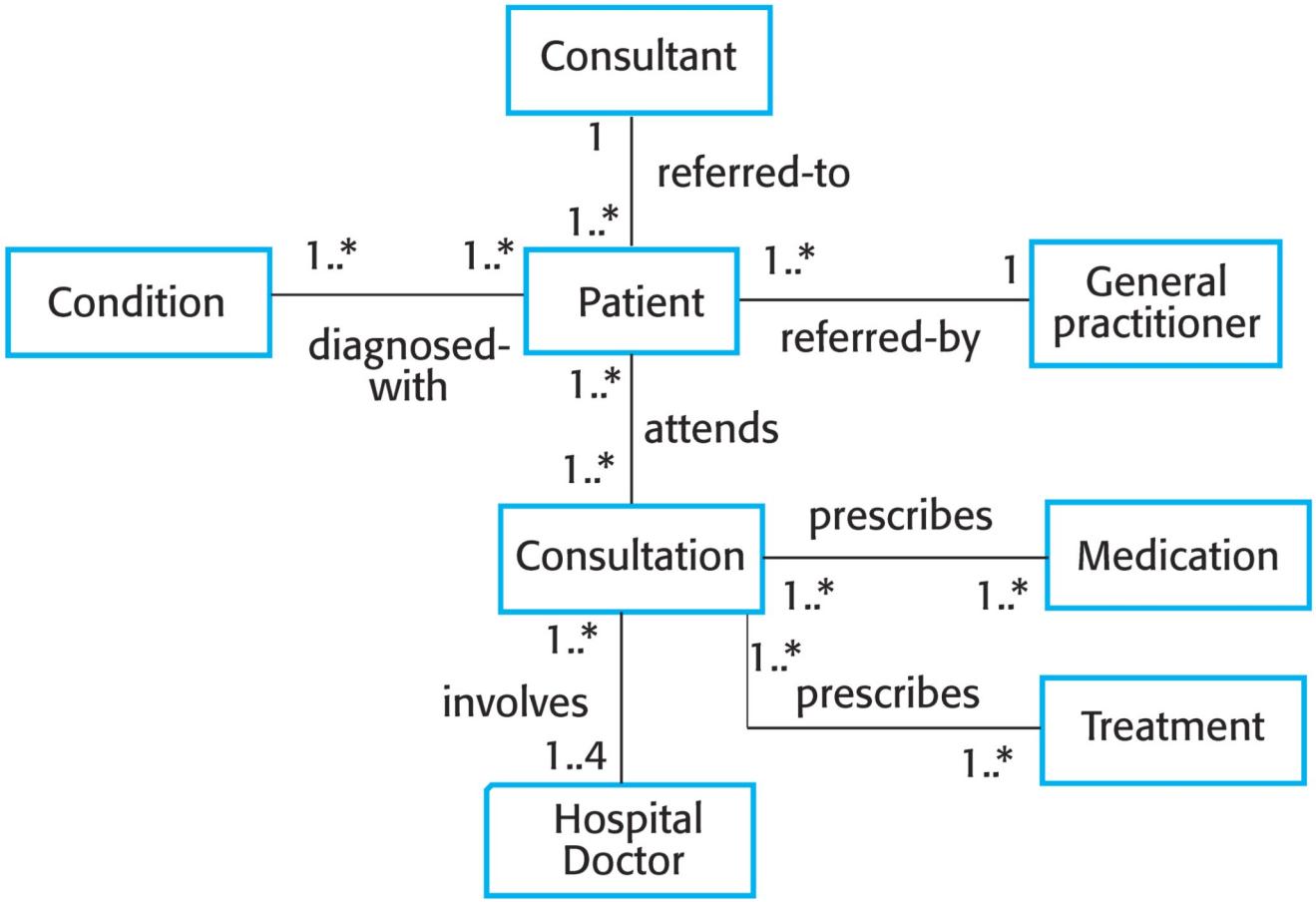
- Consist of class, attributes, methods, relationships.
  - Aggregation, Composition, Multiplicity



Class diagrams



Class diagrams



Behavioral models

[Behavioral models]

- Models of the dynamic behavior of a system as it is executing.
- Showing what happens when a system responds to a stimulus.
  - Stimulus = data or events
- Many business systems are data-driven.
  - A phone billing system (calls → cost calculation → bill generation)
- Real-time systems are usually event-driven.
  - A landline phone switching system

    (handset activation → a dial tone) (pressing keys → capturing the number)

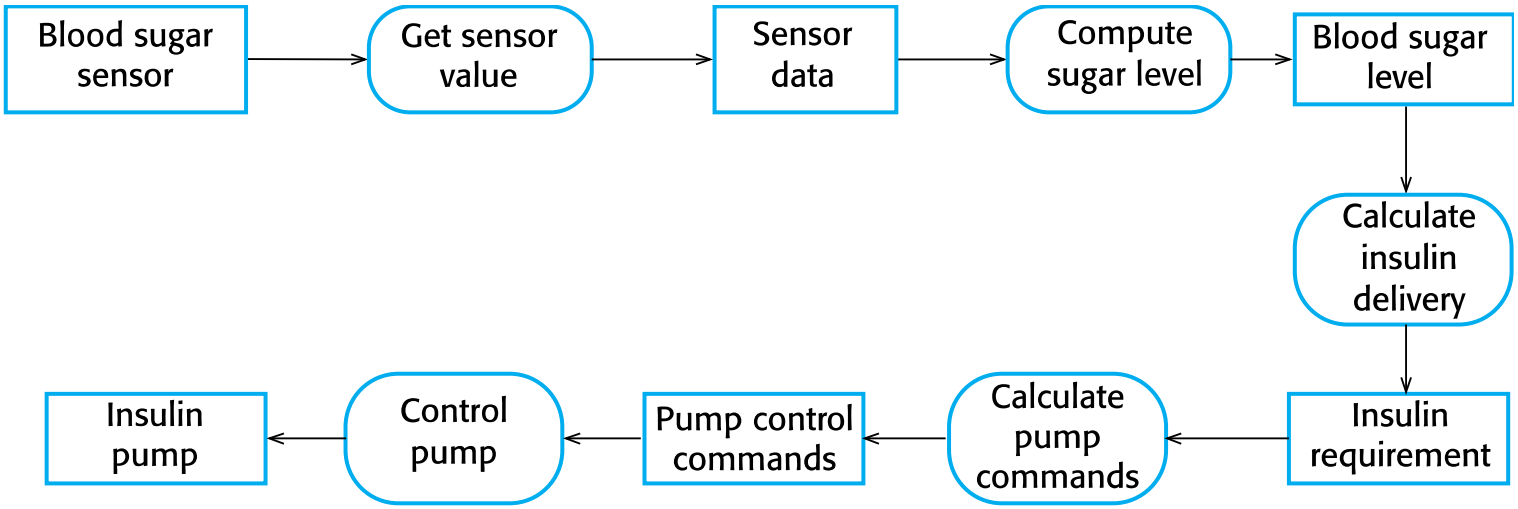
Data-driven modeling

[Data-driven model]

- Show the sequence of actions involved in processing input/output.

[Activity diagram]

- Consist of activity (rounded rectangles) and data flowing between these steps (rectangles).



Event-driven modeling

[Event-driven modeling]

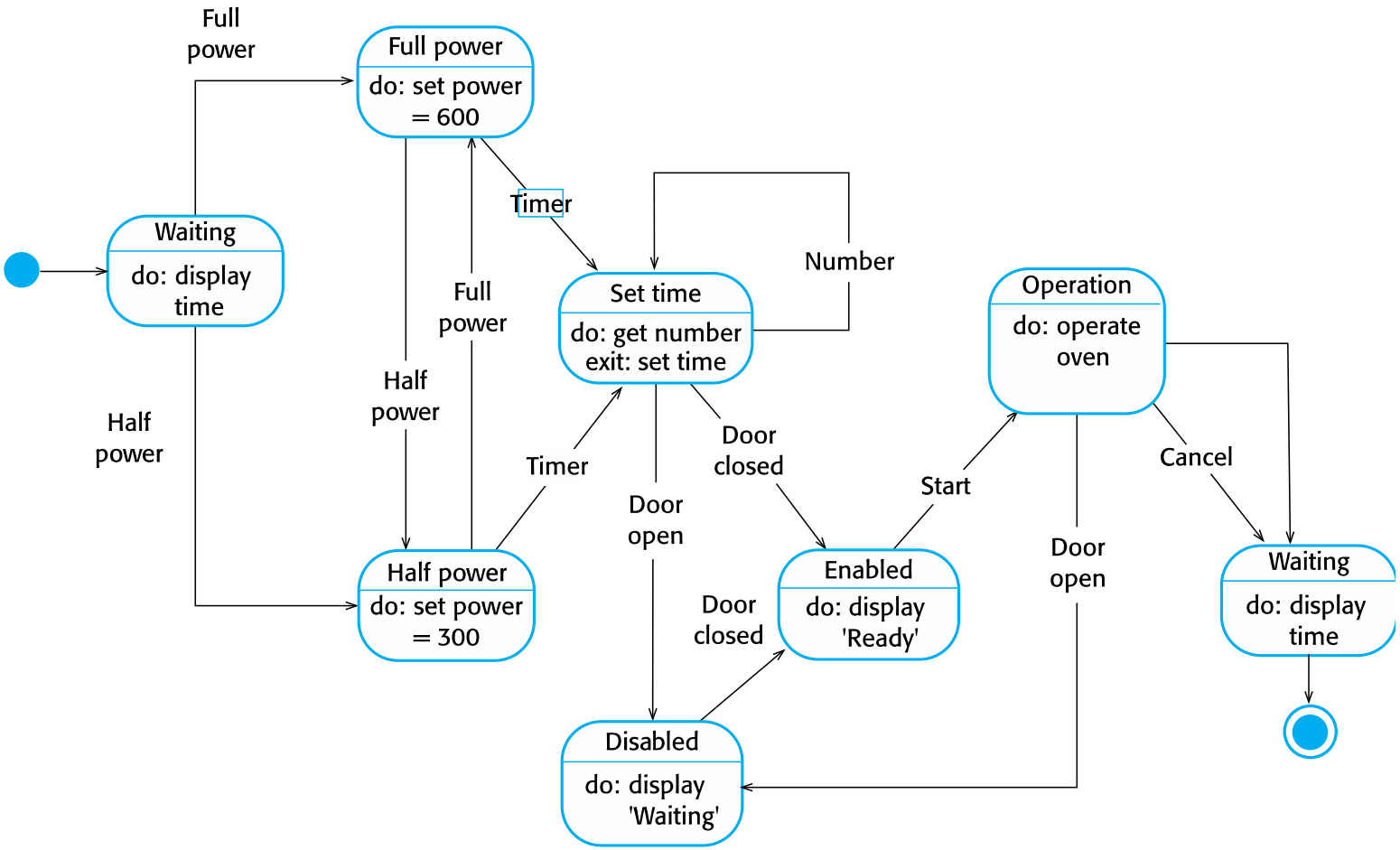
- Showing how a system responds to external and internal events.
- Assuming that a system has a finite number of states and events cause a transition from one state to another.

[State diagram]

- System states (Rounded rectangles)
  - Including a brief description of the actions taken in the state.
- Transition (labeled arrows)
  - Representing stimuli that force a transition from one state to another.

Event-driven modeling

[A state diagram of a microwave oven]



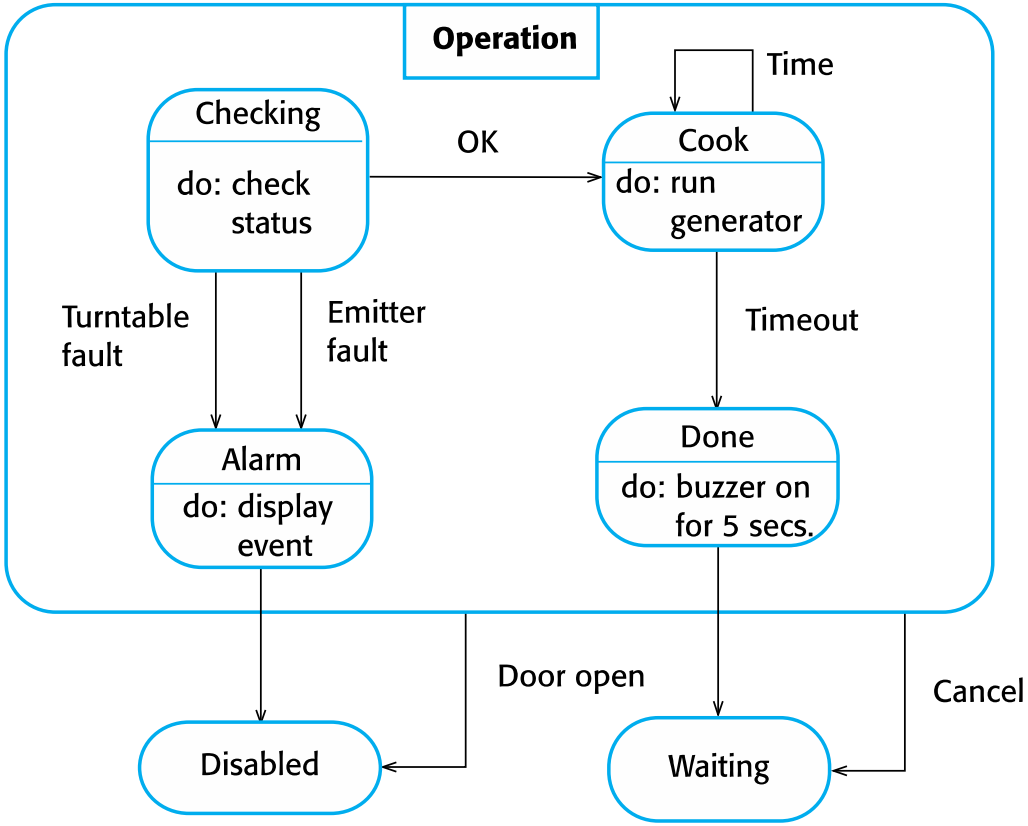
Event-driven modeling

[A state model of the Operation state]

Increase the possible states

Hide detail in the models

Use the notion of a “superstate”



Event-driven modeling

[States and stimuli for the microwave oven]

State	Description
Waiting	The oven is waiting for input. The display shows the current time.
Half power	The oven power is set to 300 watts. The display shows 'Half power'.
Full power	The oven power is set to 600 watts. The display shows 'Full power'.
Set time	The cooking time is set to the user's input value. The display shows the cooking time selected and is updated as the time is set.
Disabled	Oven operation is disabled for safety. Interior oven light is on. Display shows 'Not ready'.
Enabled	Oven operation is enabled. Interior oven light is off. Display shows 'Ready to cook'.
Operation	Oven in operation. Interior oven light is on. Display shows the timer countdown. On completion of cooking, the buzzer is sounded for five seconds. Oven light is on. Display shows 'Cooking complete' while buzzer is sounding.

Event-driven modeling

[States and stimuli for the microwave oven]

Stimulus	Description
Half power	The user has pressed the half-power button.
Full power	The user has pressed the full-power button.
Timer	The user has pressed one of the timer buttons.
Number	The user has pressed a numeric key.
Door open	The oven door switch is not closed.
Door closed	The oven door switch is closed.
Start	The user has pressed the Start button.
Cancel	The user has pressed the Cancel button.

Model-driven architecture

[Model-Driven Architecture (MDA)]

- A model-focused approach to software design and implementation
- Expectation or Hypothesis
  - The programs that execute on a SW platform are generated automatically from the models ????

- Three types of abstract system model for MDA
  - A computation independent model (CIM)
  - A platform-independent model (PIM)
  - A Platform-specific model (PSM)

Model-driven engineering

[Three types of abstract system model for MDA]

CIM

- CIMs (a domain model) model the important domain abstractions used in a system. (e.g., models of the actual people, places, things of a domain)

PIM

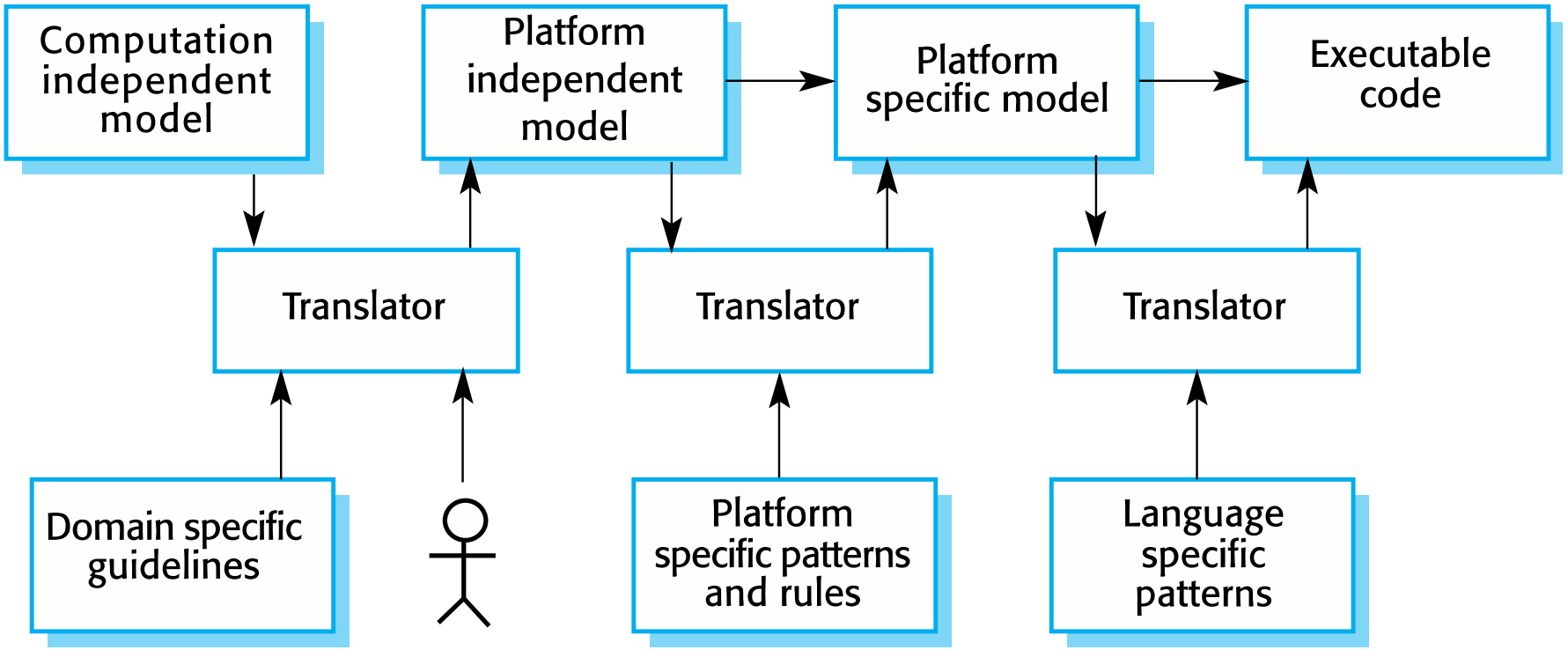
- PIM enables its mapping to one or more platforms by defining a set of services in a way that abstracts out technical details.

PSM

- A PSM combines the specifications in the PIM with the details required to stipulate how a system uses a particular type of platform.

Model-driven engineering

[MDA transformations]



Model-driven architecture

[MDA is not a mainstream approach to SW development]

- The abstractions that are useful for discussions are not the right abstractions for implementation.
- For complex systems, implementation is not the major problem.
  - ex) requirements engineering, security and dependability, testing, ...
- The widespread adoption of agile methods has diverted attention away from model-driven approaches.

Chapter 5. System Modeling

Summary