Week 5:
System Modeling

Nguyễn Thị Minh Tuyền

KHOA CÔNG NGHỆ T
TRƯỜNG ĐẠI HỌC K

Adapted from slides of Ian Sommerville

Topics covered

1. Context models
2. Interaction models
3. Structural models
4. Behavioral models
5. Model-driven engineering

System modeling

- Is the process of developing abstract models of a system
  - each model presents a different view or perspective of that system.
- Represent a system using some kind of graphical notation
  - based on notations in the Unified Modeling Language (UML).
- Helps the analyst to understand the functionality of the system and models are used to communicate with customers.

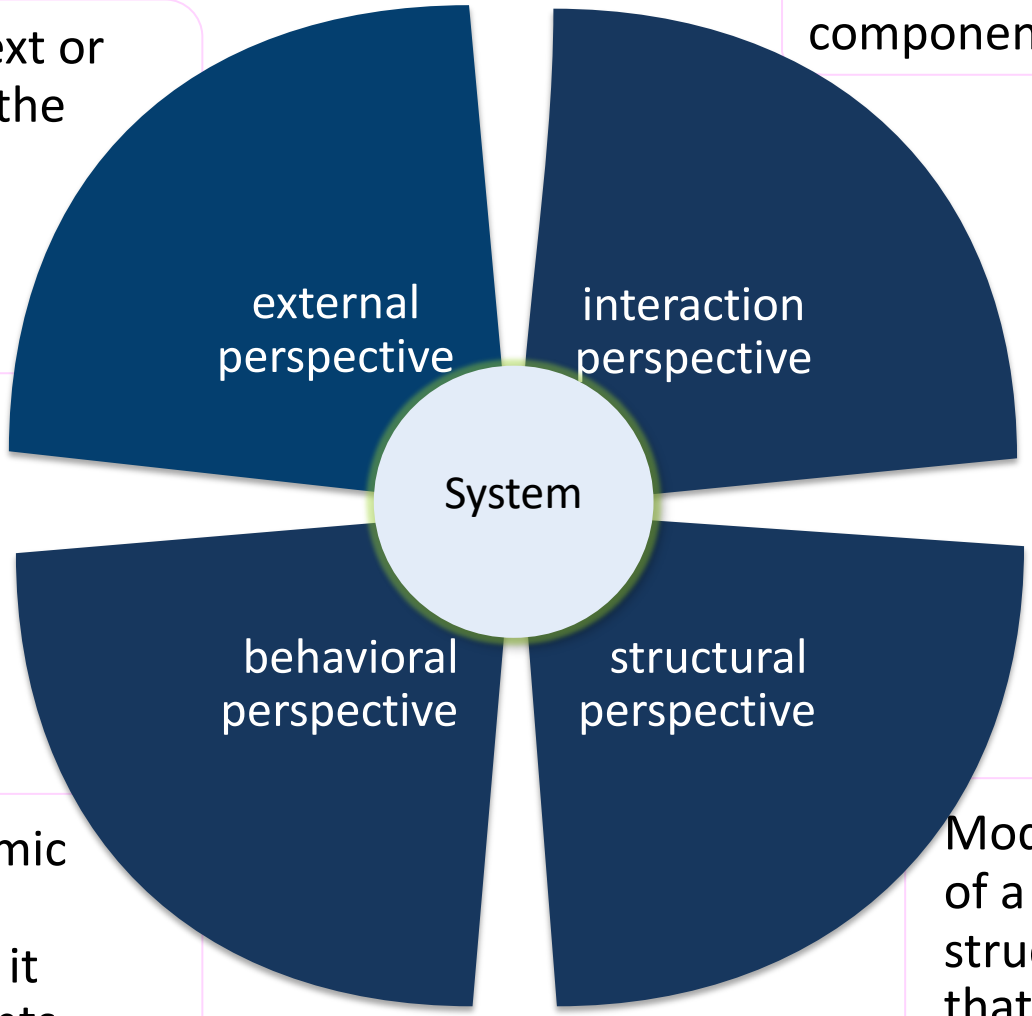
Existing and planned system models

- Models of the existing system are used during requirements engineering.
  - **Clarify what the existing system does** and
  - Are used as a **basis for discussing its strengths and weaknesses**.
- Models of the new system are used during requirements engineering
  - Help explain the proposed requirements to other system stakeholders.
  - Are used for **discussing design proposals** and for **documenting the system for implementation**.
- In a model-driven engineering process, it is possible to generate a complete or partial system implementation from the system model.

System perspectives

Model the interactions between a system and its environment, or between the components of a system

Model the context or environment of the system



Model the dynamic behavior of the system and how it responds to events.

Model the organization of a system or the structure of the data that is processed by the system.

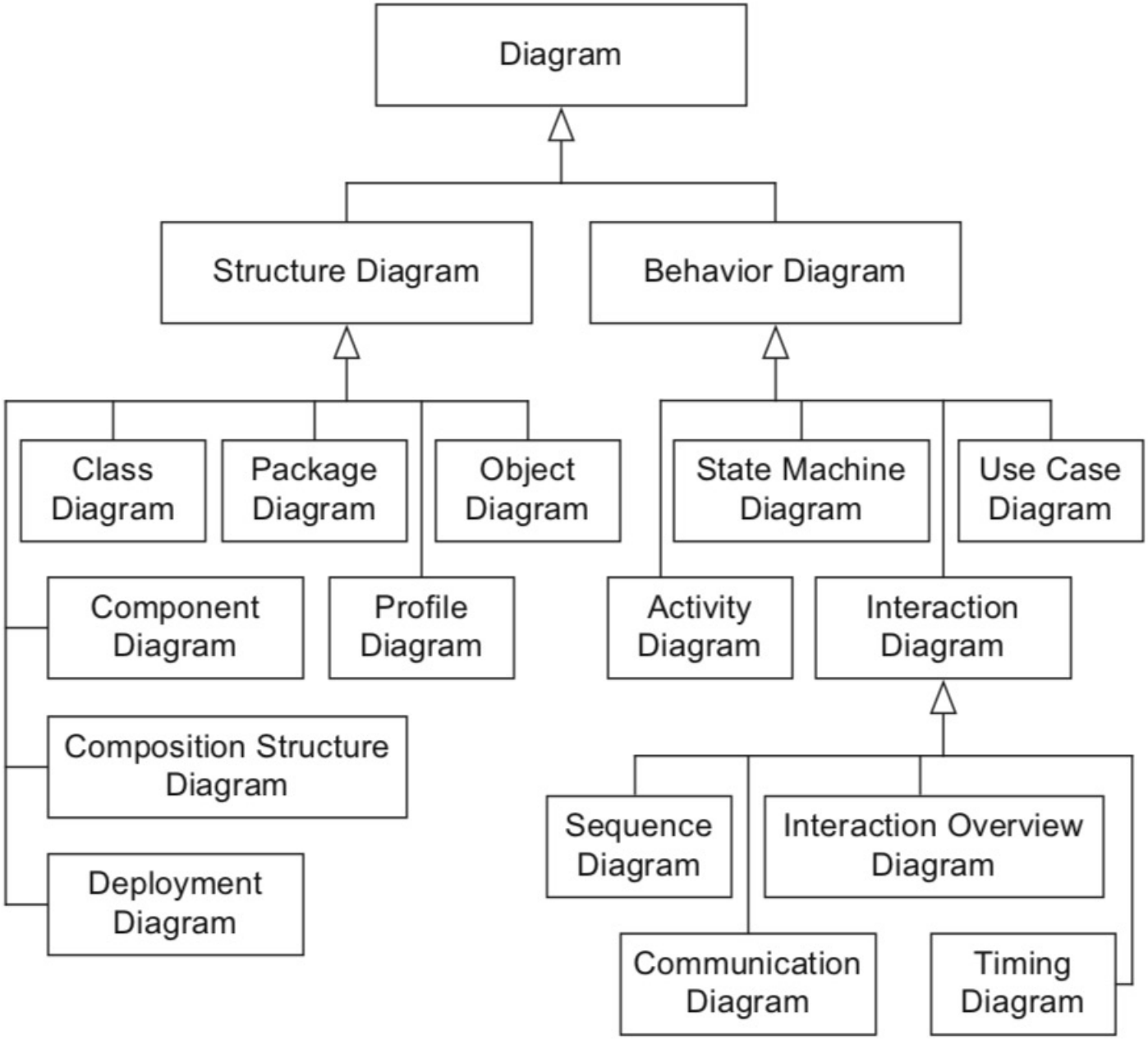
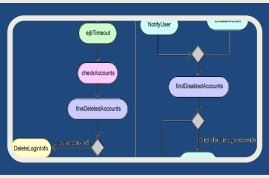


Diagram
Structure Diagram
Behavior Diagram
Class Diagram
Package Diagram
Object Diagram
State Machine Diagram
Use Case Diagram
Component Diagram
Profile Diagram
Activity Diagram
Interaction Diagram
Composition Structure Diagram
Sequence Diagram
Interaction Overview Diagram
Deployment Diagram
Communication Diagram
Timing Diagram

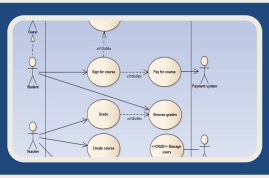
UML diagram types

The UML has many diagram types and supports many different types of system model.
Five diagram types could represent the essentials of a system:



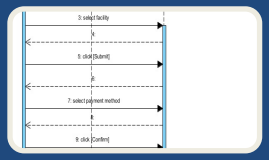
Activity diagram

Show the activities involved in a process or in data processing.



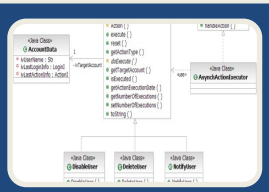
Use case diagram

Show the interactions between a system and its environment.



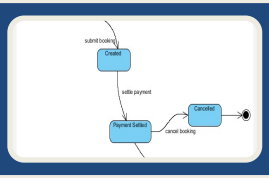
Sequence diagram

Show interactions between actors and the system and between system components.



Class diagram

Show the object classes in the system and the associations between these classes.



State diagram

Show how the system reacts to internal and external events.

Use of graphical models

- As a means of facilitating discussion about an existing or proposed system
  - Incomplete and incorrect models are OK as their role is to support discussion.
- As a way of documenting an existing system
  - Models should be an accurate representation of the system but need not be complete.
- As a detailed system description that can be used to generate a system implementation
  - Models have to be both correct and complete.

Topics covered

1. Context models
2. Interaction models
3. Structural models
4. Behavioral models
5. Model-driven engineering

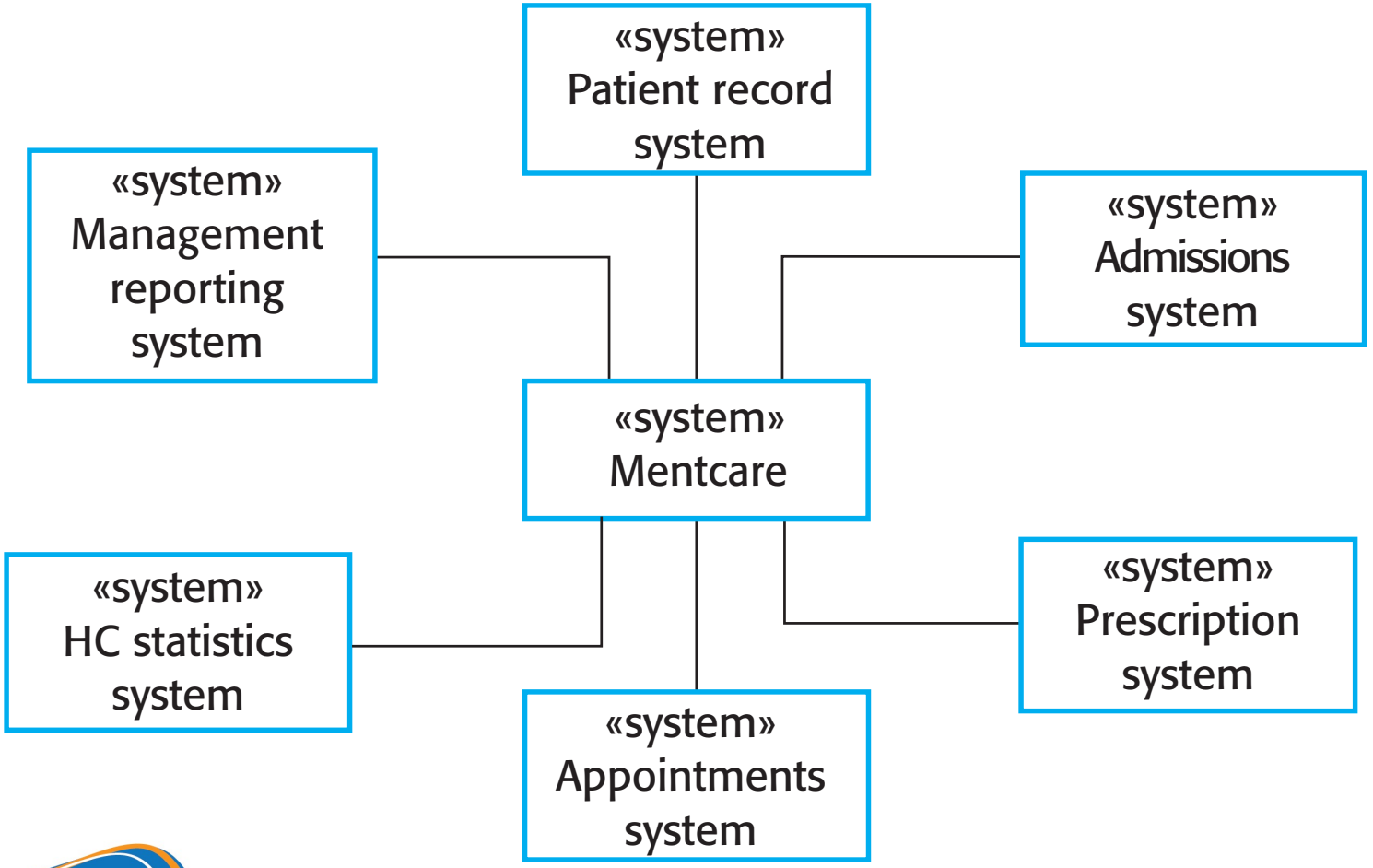
Context models

- Context models are used to illustrate the operational context of a system - they show what lies outside the system boundaries.
- Social and organisational concerns may affect the decision on where to position system boundaries.
- Architectural models show the system and its relationship with other systems.

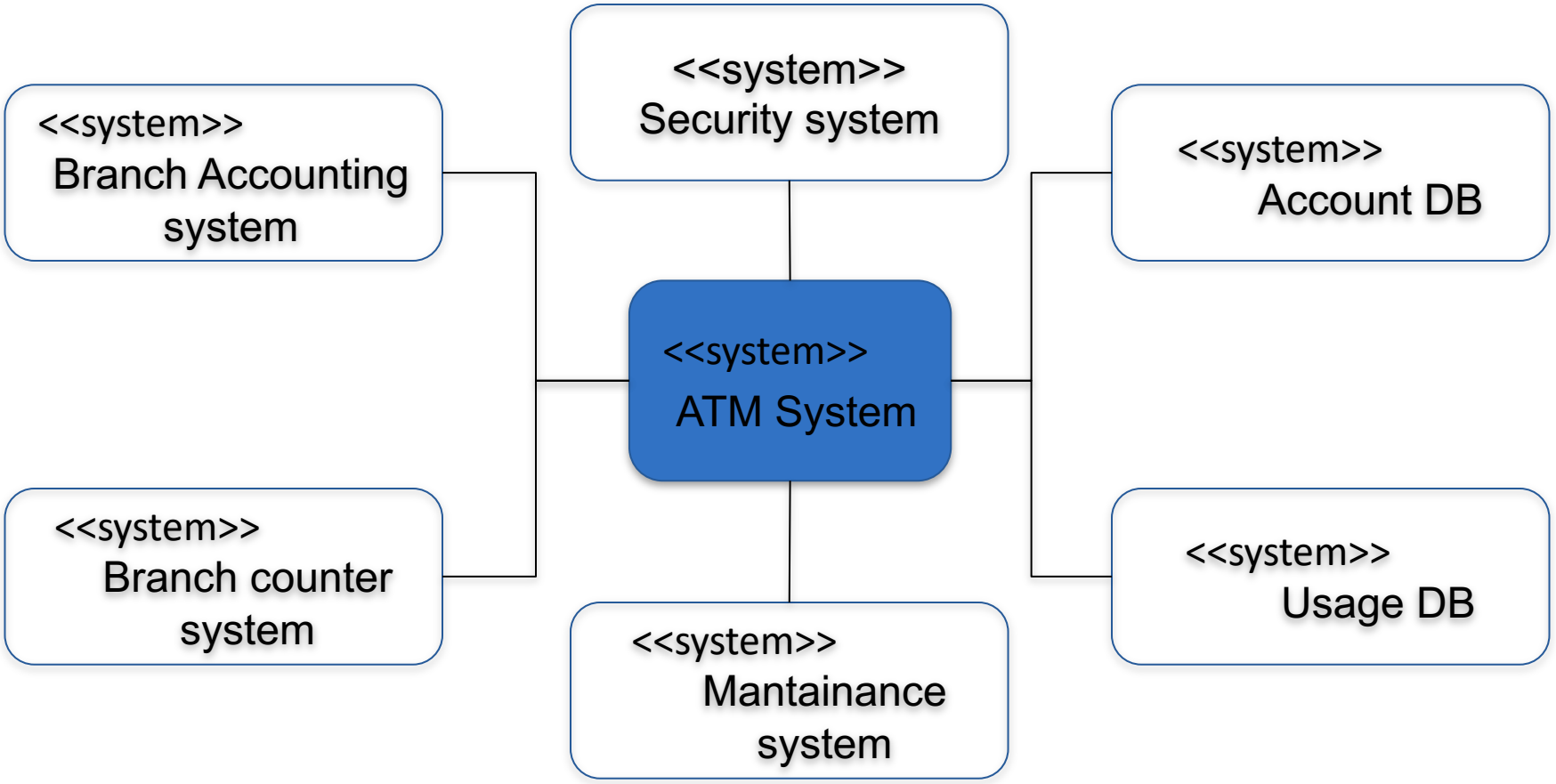
System boundaries

- System boundaries are established to define what is inside and what is outside the system.
  - They show other systems that are used or depend on the system being developed.
- The position of the system boundary has a profound effect on the system requirements.
- Defining a system boundary is a political judgment
  - There may be pressures to develop system boundaries that increase / decrease the influence or workload of different parts of an organization.

The context of the Mentcare system



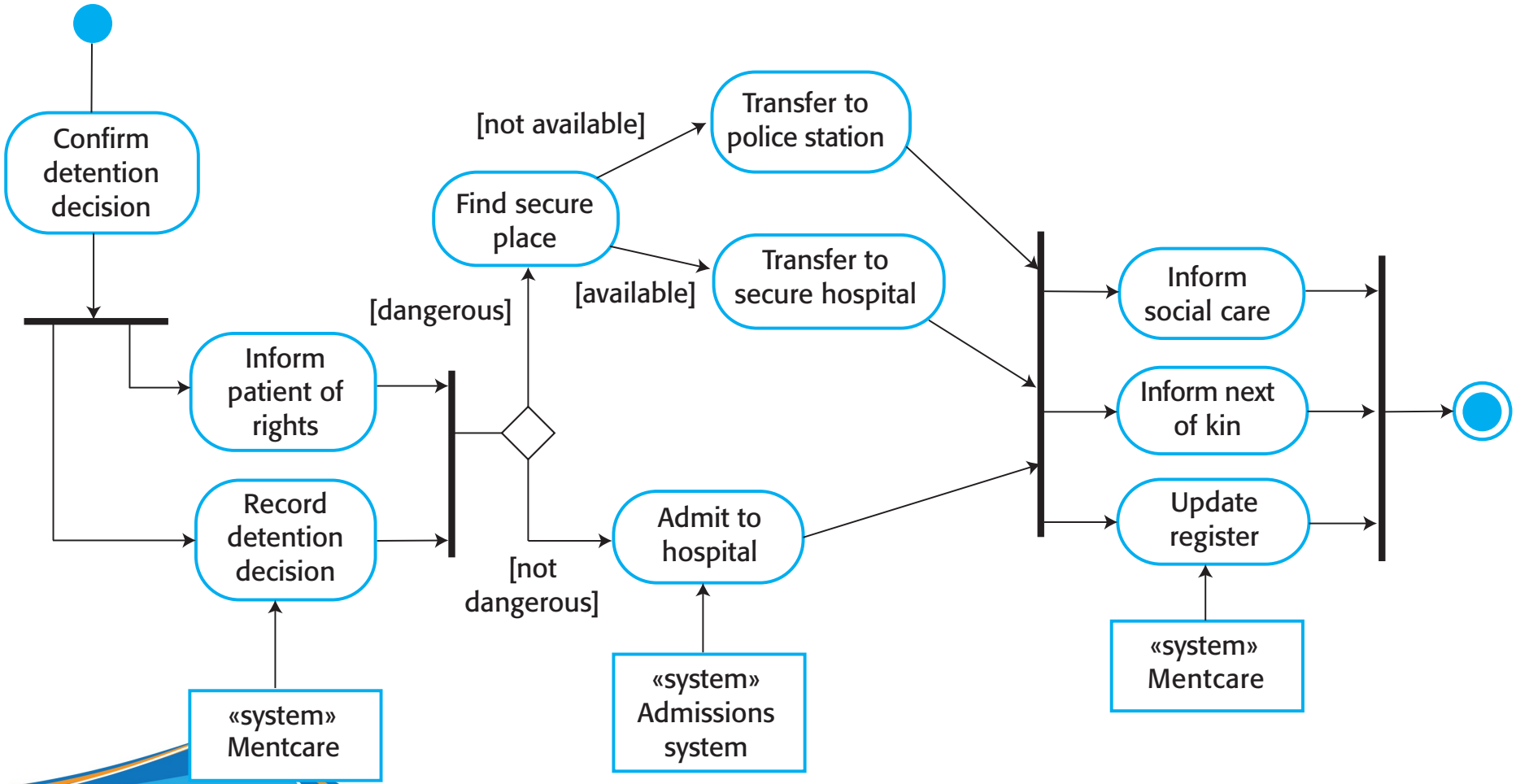
The context of the ATM system



Process perspective

- Context models simply show the other systems in the environment, not how the system being developed is used in that environment.
- Process models reveal how the system being developed is used in broader business processes.
- UML activity diagrams may be used to define business process models.

Process model of involuntary detention



Topics covered

1. Context models
2. Interaction models
3. Structural models
4. Behavioral models
5. Model-driven engineering

Interaction models

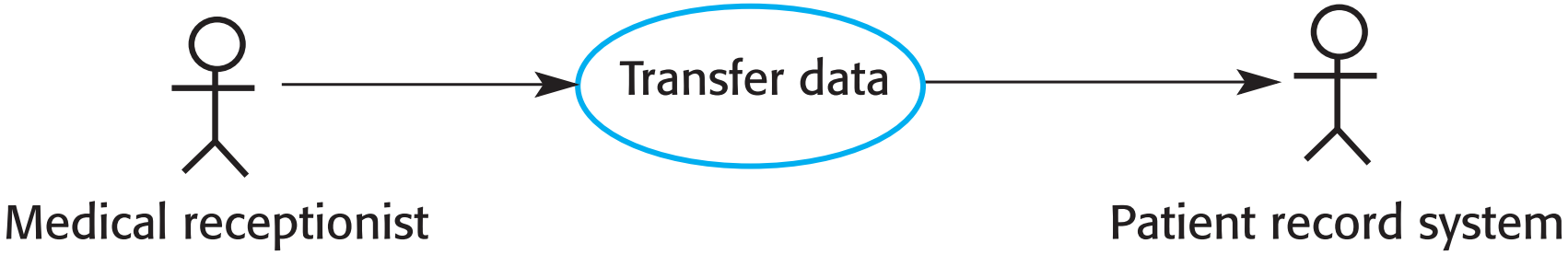
- Modeling user interaction is important as it helps to identify user requirements.
- Modeling system-to-system interaction highlights the communication problems that may arise.
- Modeling component interaction helps us understand if a proposed system structure is likely to deliver the required system performance and dependability.
- Two approaches to interaction modeling:
  - Use case diagrams and
  - sequence diagrams.

Use case modeling

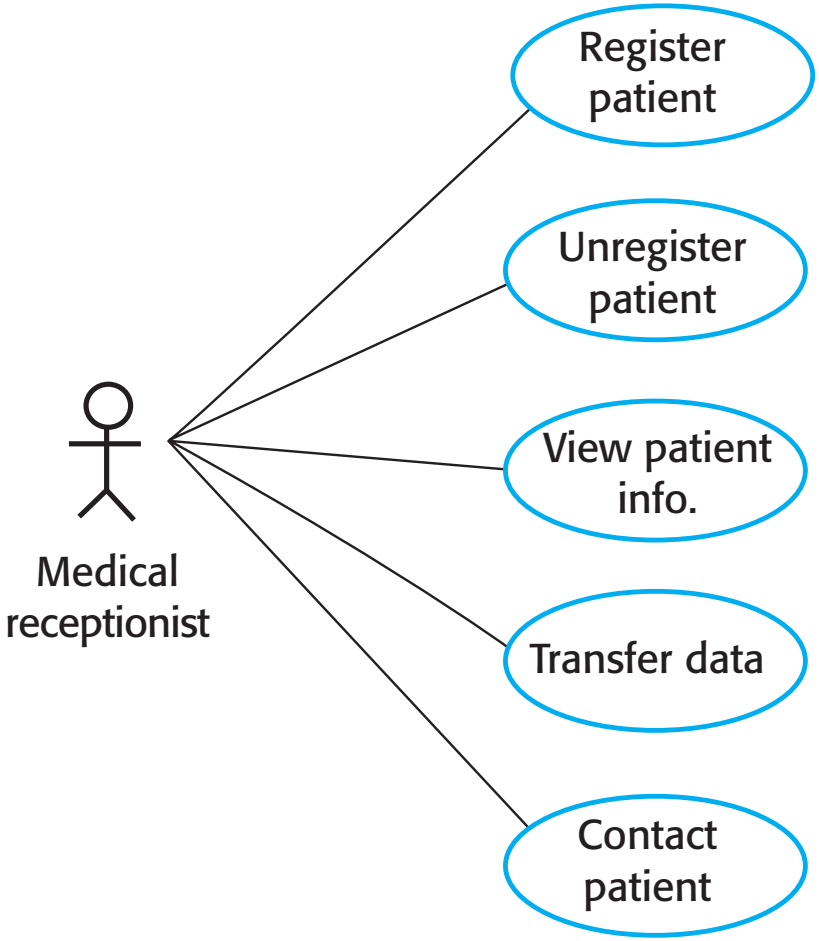
- Use cases were developed originally to support requirements elicitation and now incorporated into the UML.
- Each use case represents a discrete task that involves external interaction with a system.
- Actors in a use case may be people or other systems.
- Represented diagrammatically to provide an overview of the use case and in a more detailed textual form.

Transfer-data use case

- A use case in the Mentcare system



Use cases in the Mentcare system involving the role ‘Medical Receptionist’



Example: Use-case specification

Use case name	**Use case: PayVAT**
Unique identifier	**ID: UC1**
The actors involved in the use case	**Actors:** Time Government
The system state before the use case can begin	**Preconditions:** 1. It is the end of a business quarter.
The actual steps of the use case	**Flow of events:** 1. The use case starts when it is the end of the business quarter. 2. The system determines the amount of Value Added Tax (VAT) owed to the Government. 3. The system sends an electronic payment to the Government.
The system state when the use case is over	**Postconditions:** 1. The Government receives the correct amount of VAT.

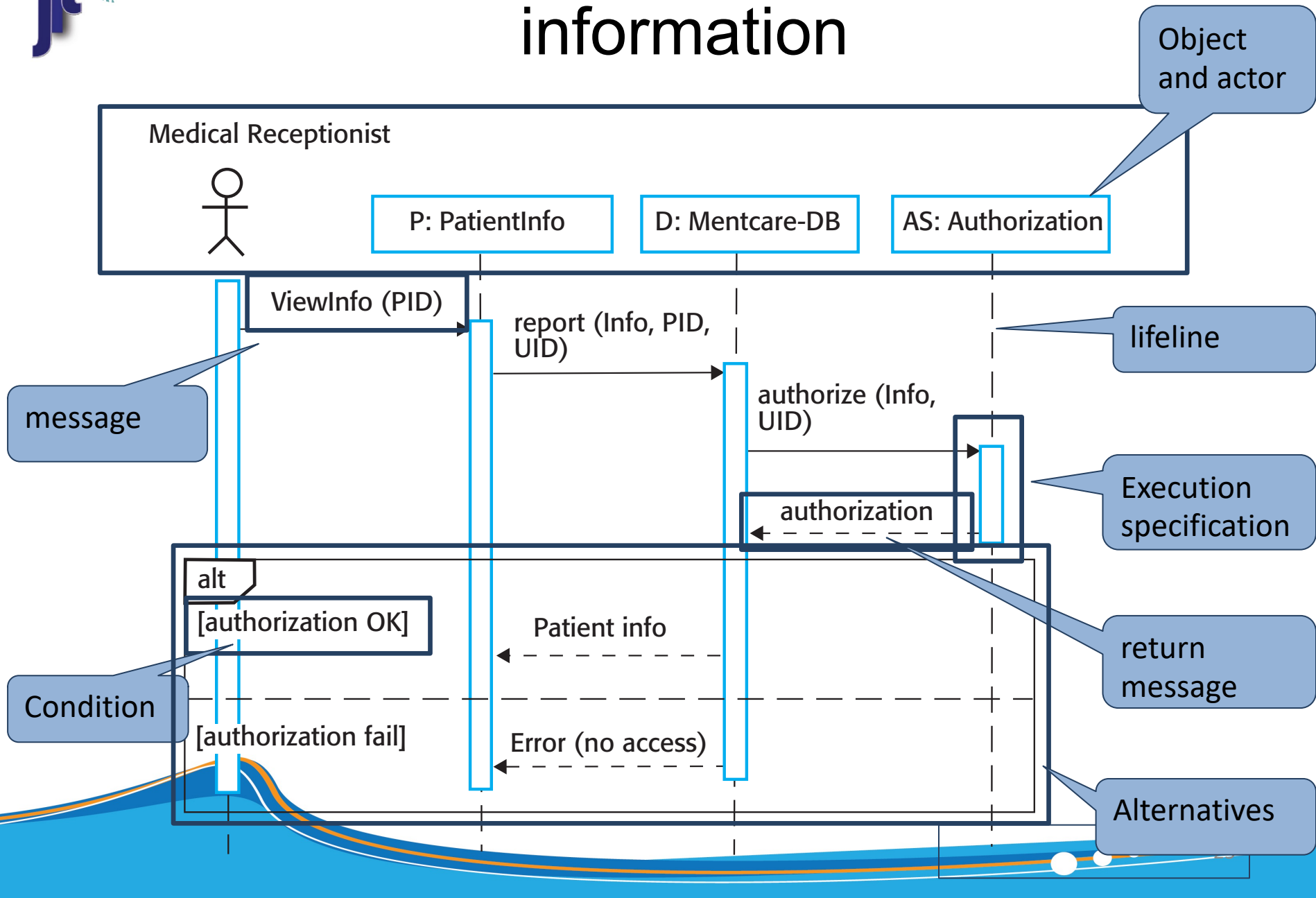
Use case: DisplayBasket
ID: UC11
Actors: Customer
Preconditions: 1. The Customer is logged on the system.
Flow of events: 1. The use case starts when the Customer selects "display basket". 2. If there are no items in the basket 2.1 The system informs the Customer that there are no items in the basket yet. 2.2 The use case terminates. 3. The system displays a list of all items in the Customer's shopping basket including product ID, name, quantity and item price.
Postconditions:
Alternative flow 1: 1. At any time the Customer may leave the shopping basket screen.
Postconditions:
Alternative flow 2: 1. At any time the Customer may leave the system.
Postconditions:

Use case: FindProduct
ID: UC12
Actors: Customer
Preconditions:
Flow of events: 1. The Customer selects “find product”. 2. The system asks the Customer for search criteria. 3. The Customer enters the requested criteria. 4. The system searches for products that match the Customer’s criteria. 5. If the system finds some matching products then 5.1. For each product found 5.1.1. The system displays a thumbnail sketch of the product. 5.1.2. The system displays a summary of the product details. 5.1.3. The system displays the product price. 6. Else 6.1. The system tells the Customer that no matching products could be found.
Postconditions:
Alternative flow: 1. At any point the Customer may move to different page.
Postconditions:

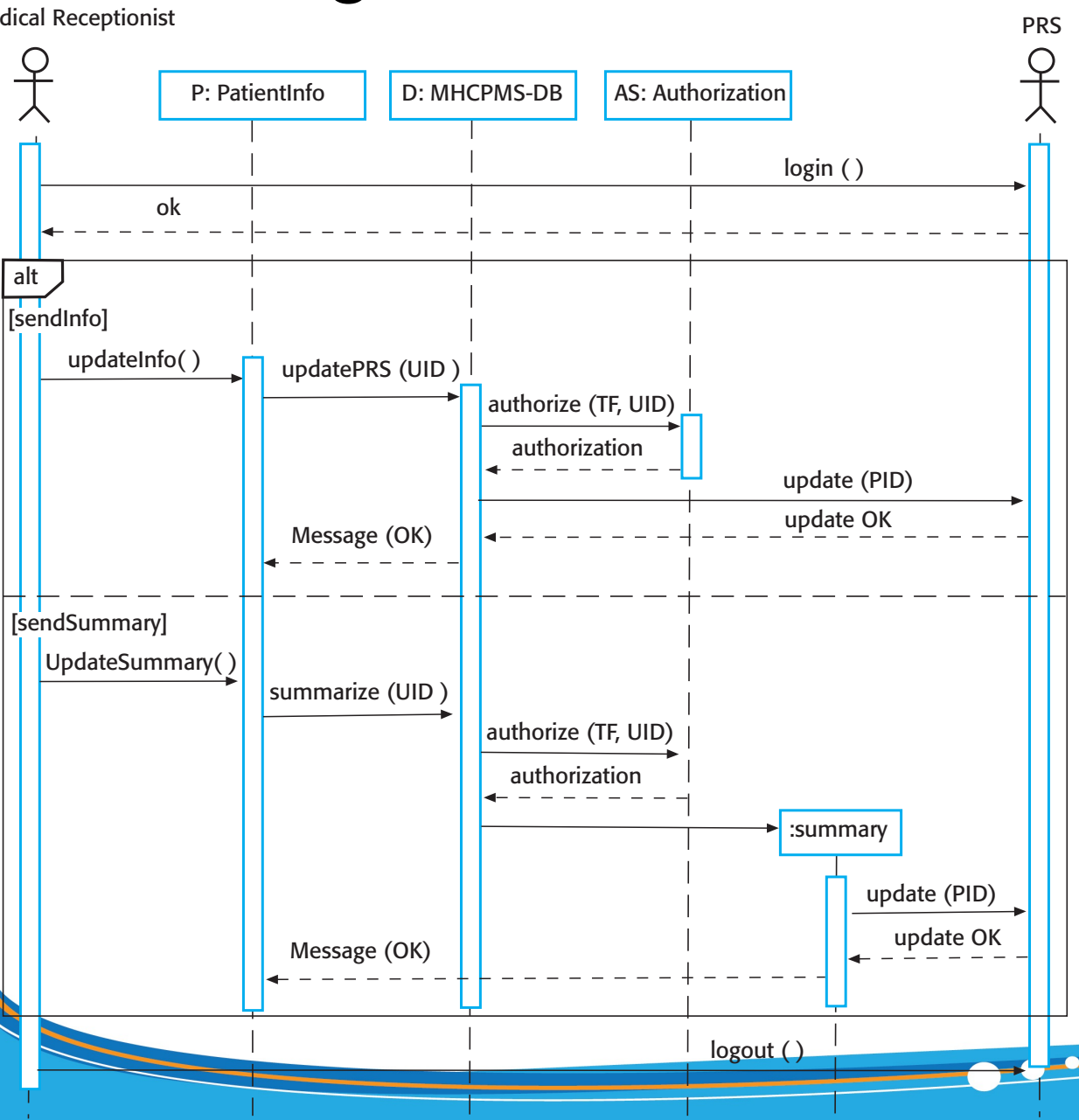
Sequence diagrams

- Sequence diagrams are part of the UML and are used to model the interactions between the actors and the objects within a system.
- A sequence diagram shows the sequence of interactions that take place during a particular use case or use case instance.
- The objects and actors involved are listed along the top of the diagram, with a dotted line drawn vertically from these.
- Interactions between objects are indicated by annotated arrows.

Sequence diagram for View patient information



Sequence diagram for Transfer Data



Topics covered

1. Context models
2. Interaction models
3. Structural models
4. Behavioral models
5. Model-driven engineering

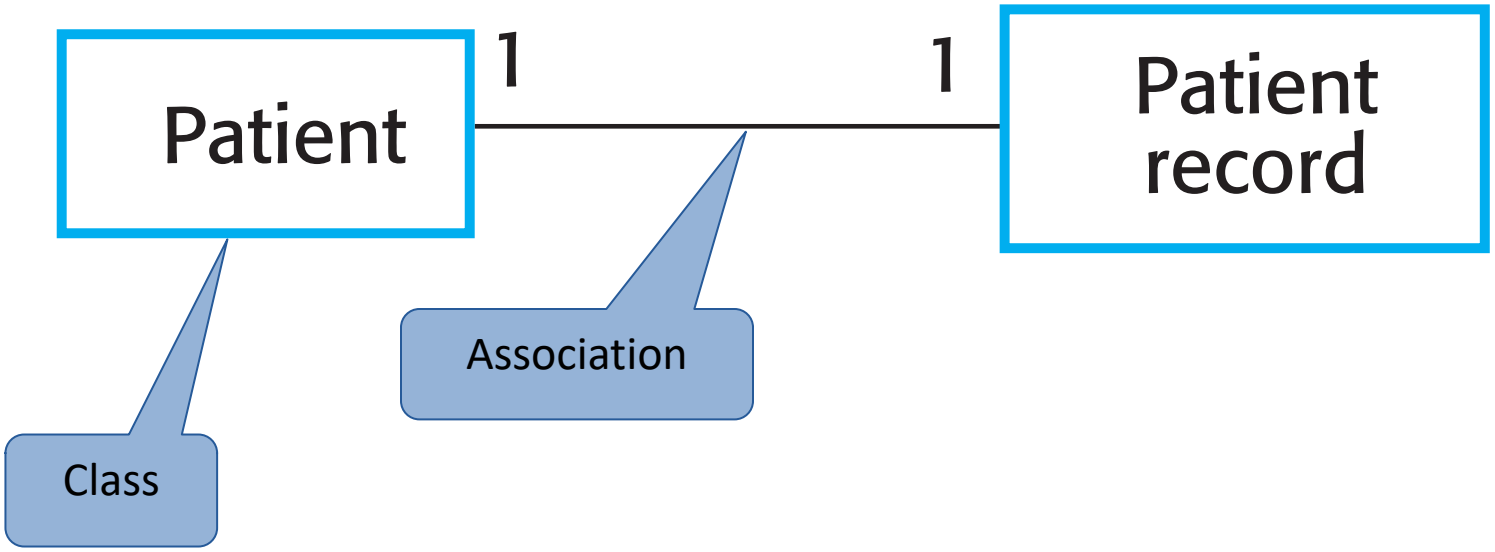
Structural models

- Display the organization of a system in terms of the components that make up that system and their relationships.
- May be
  - **static models**, which show the structure of the system design, or
  - **dynamic models**, which show the organization of the system while executing.
- Create structural models of a system when you are discussing and designing the system architecture.

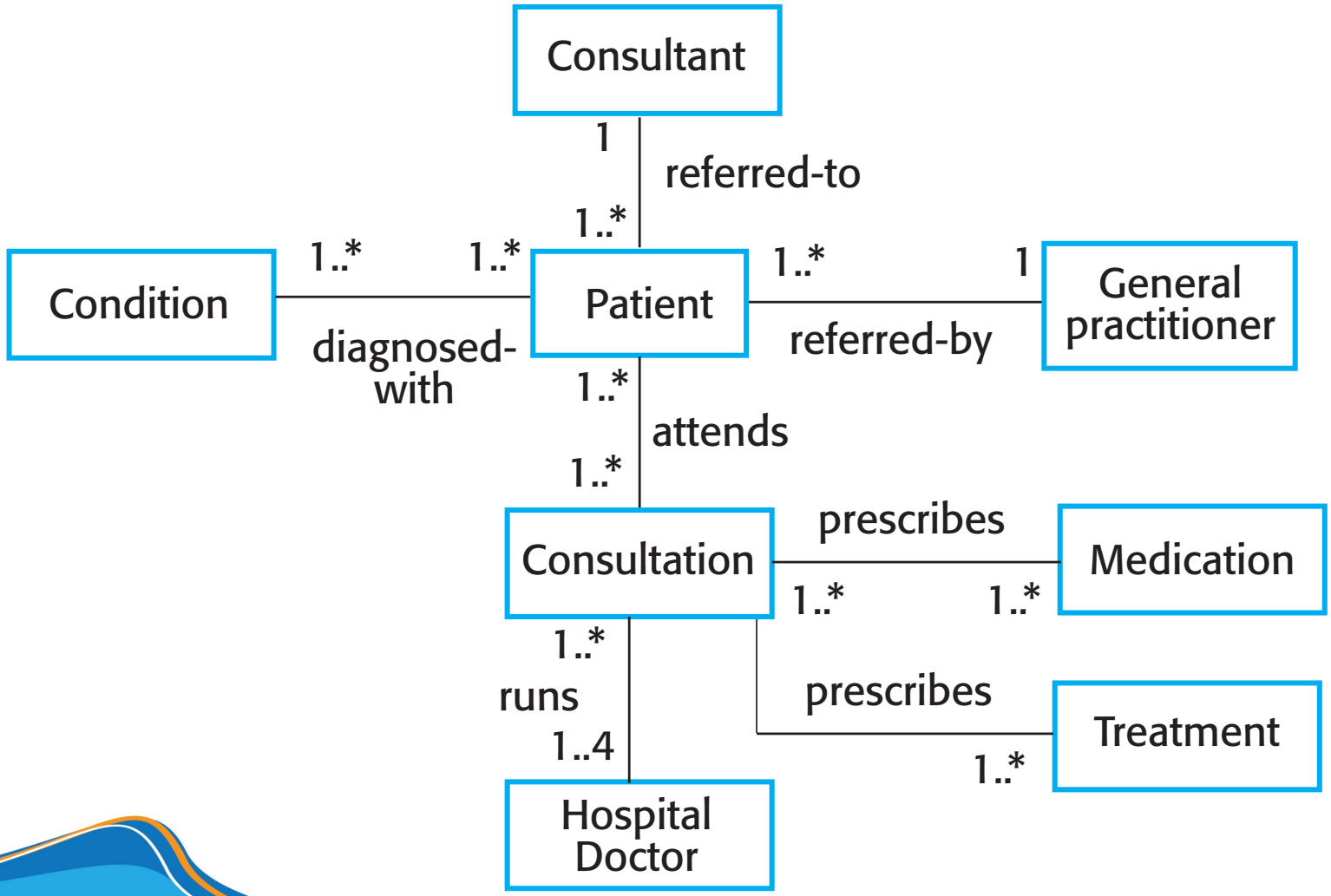
Class diagrams

- Used when developing an object-oriented system model to show the classes in a system and the associations between these classes.
- An object class can be thought of as a general definition of one kind of system object.
- An association is a link between classes.
- During the early stages of the software engineering process, objects represent something in the real world.
  - For example: a patient, a prescription, doctor, etc.

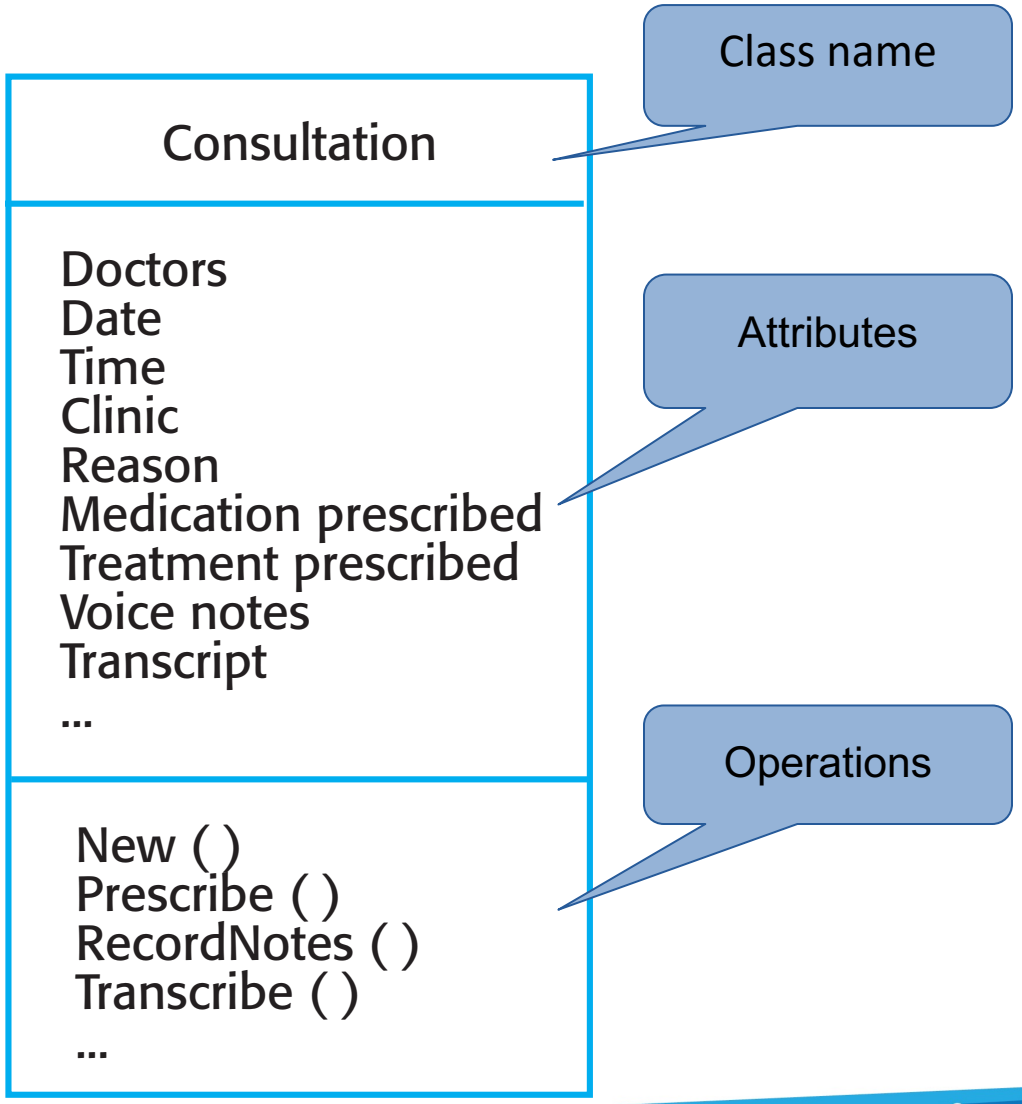
UML classes and association



Classes and associations in the MHC-PMS



The Consultation class



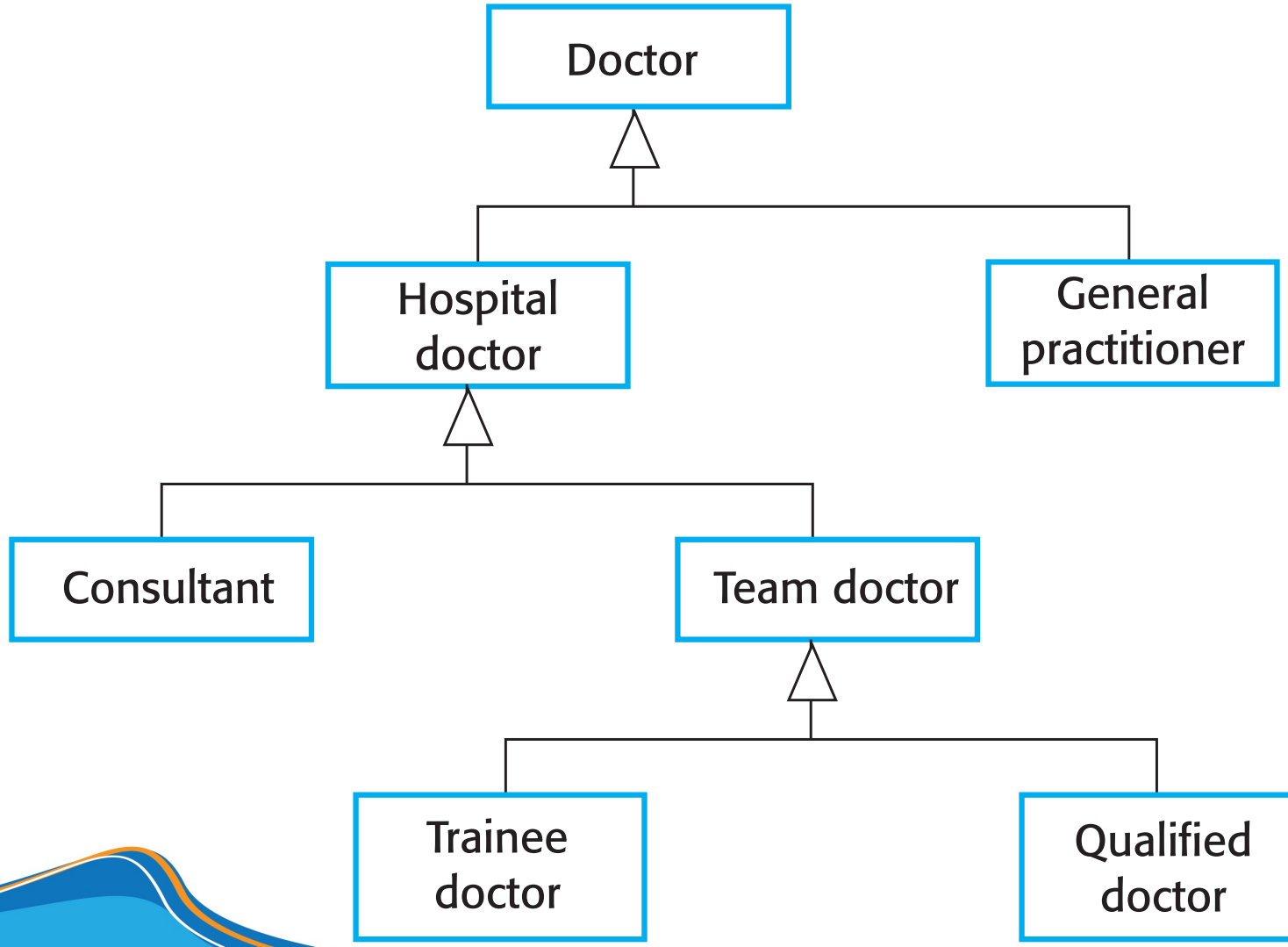
Generalization [1]

- Is an everyday technique to manage complexity.
- Rather than learn the detailed characteristics of every entity, we place these entities in more general classes (animals, cars, houses, etc.) and learn the characteristics of these classes.
- Allows us to infer that different members of these classes have some common characteristics.

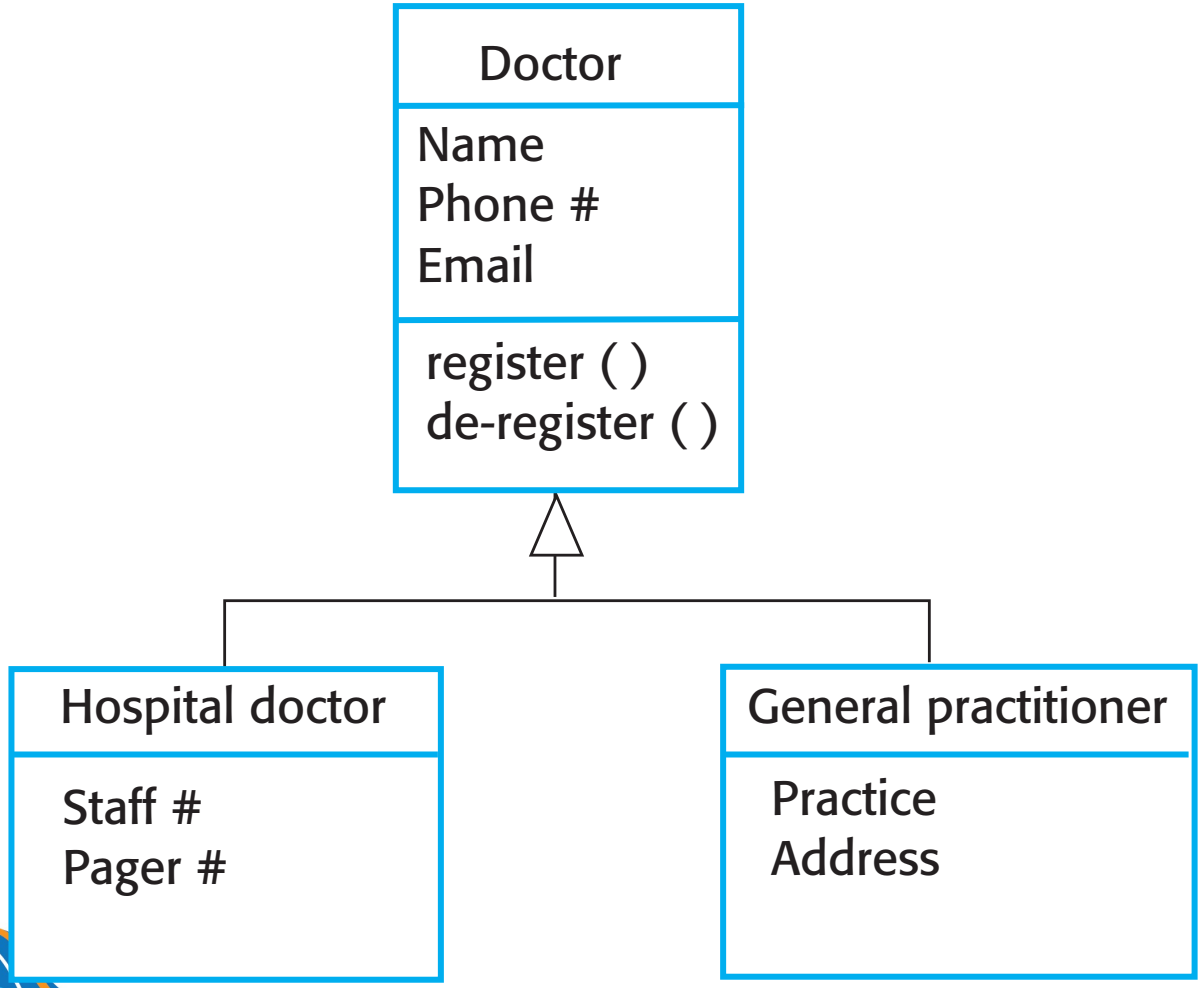
Generalization [2]

- In object-oriented languages, such as Java, generalization is implemented using the class inheritance mechanisms built into the language.
- In a generalization:
  - The attributes and operations associated with higher-level classes are also associated with the lower-level classes.
  - The lower-level classes are subclasses inherit the attributes and operations from their superclasses. These lower-level classes then add more specific attributes and operations.

A generalization hierarchy



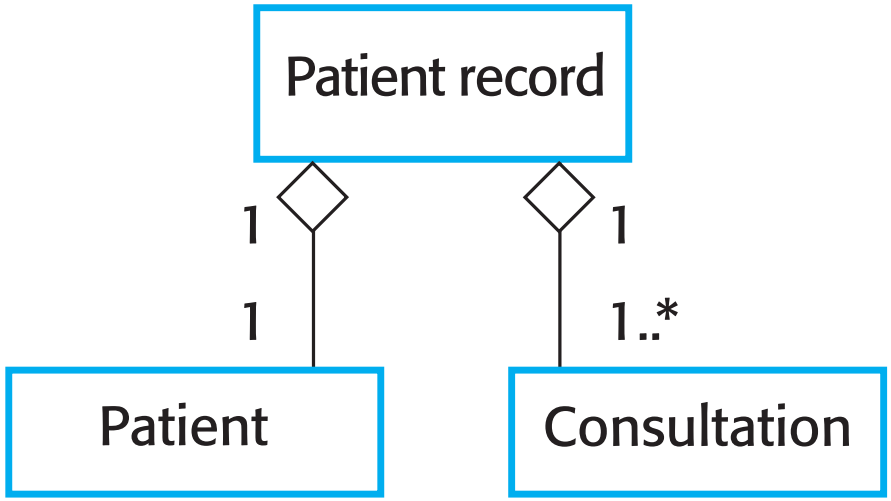
A generalization hierarchy with added detail



Object class aggregation models

- Show how classes that are collections are composed of other classes.
- Are similar to the part-of relationship in semantic data models.

The aggregation association



Topics covered

1. Context models
2. Interaction models
3. Structural models
4. Behavioral models
5. Model-driven engineering

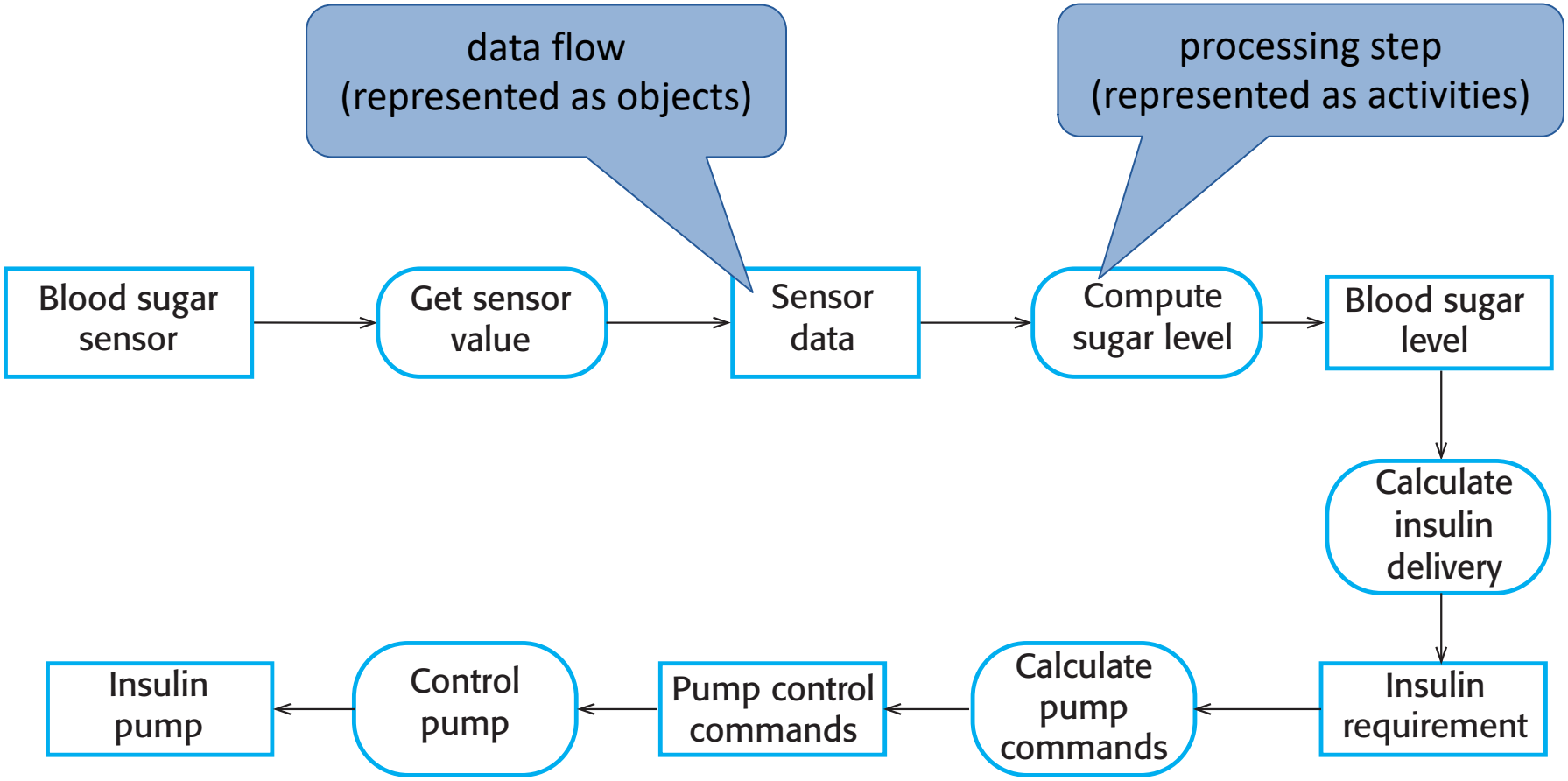
Behavioral models

- Models of the dynamic behavior of a system as it is executing.
  - They show what happens or what is supposed to happen when a system responds to a stimulus from its environment.
- You can think of these stimuli as being of two types:
  - **Data** Some data arrives that has to be processed by the system.
  - **Events** Some event happens that triggers system processing. Events may have associated data, although this is not always the case.

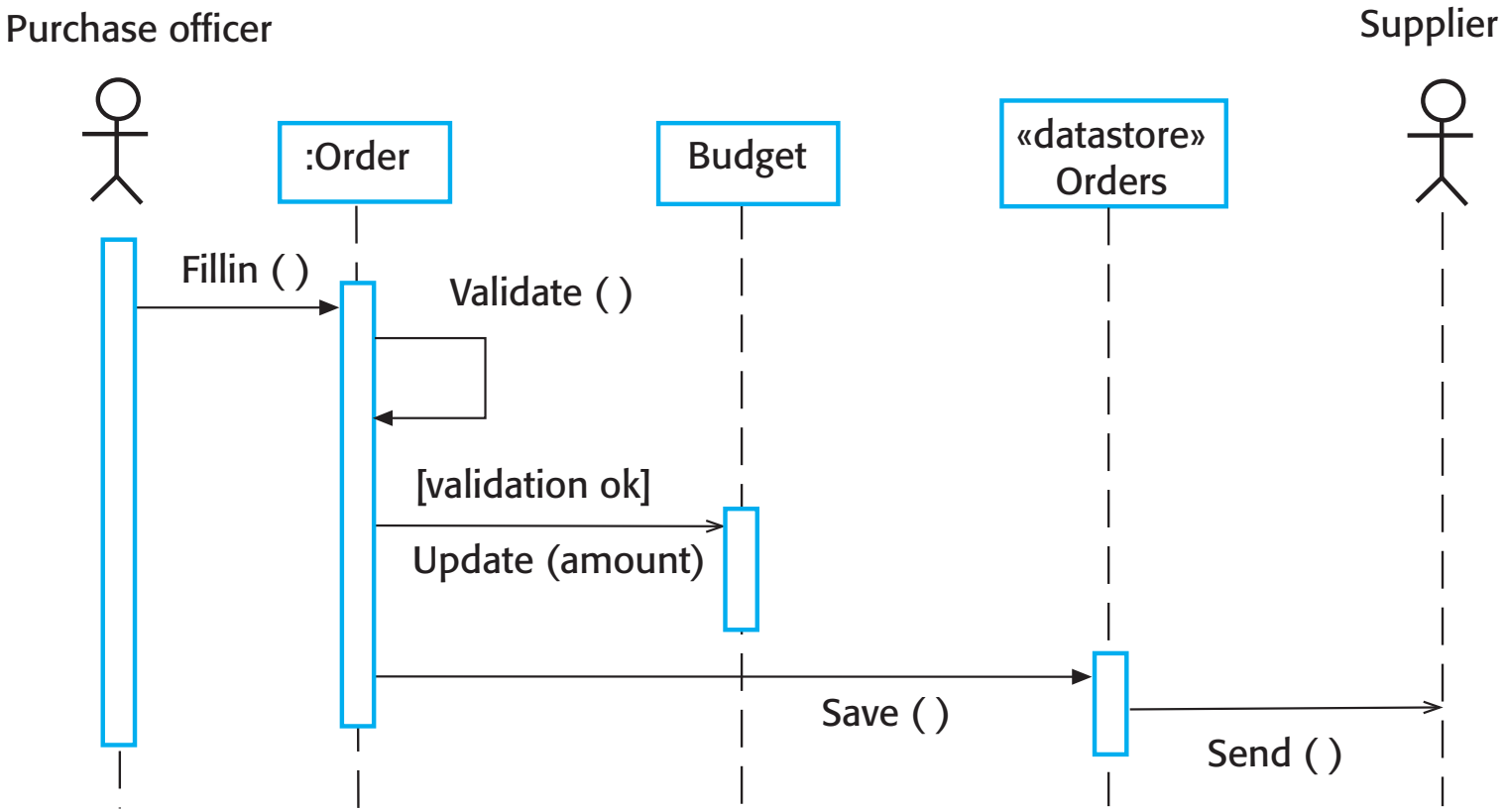
Data-driven modeling

- Many business systems are data-processing systems that are primarily driven by data.
  - They are controlled by the data input to the system, with relatively little external event processing.
- Data-driven models show the sequence of actions involved in processing input data and generating an associated output.
- They are particularly useful during the analysis of requirements as they can be used to show end-to-end processing in a system.

An activity model of the insulin pump's operation



Order processing



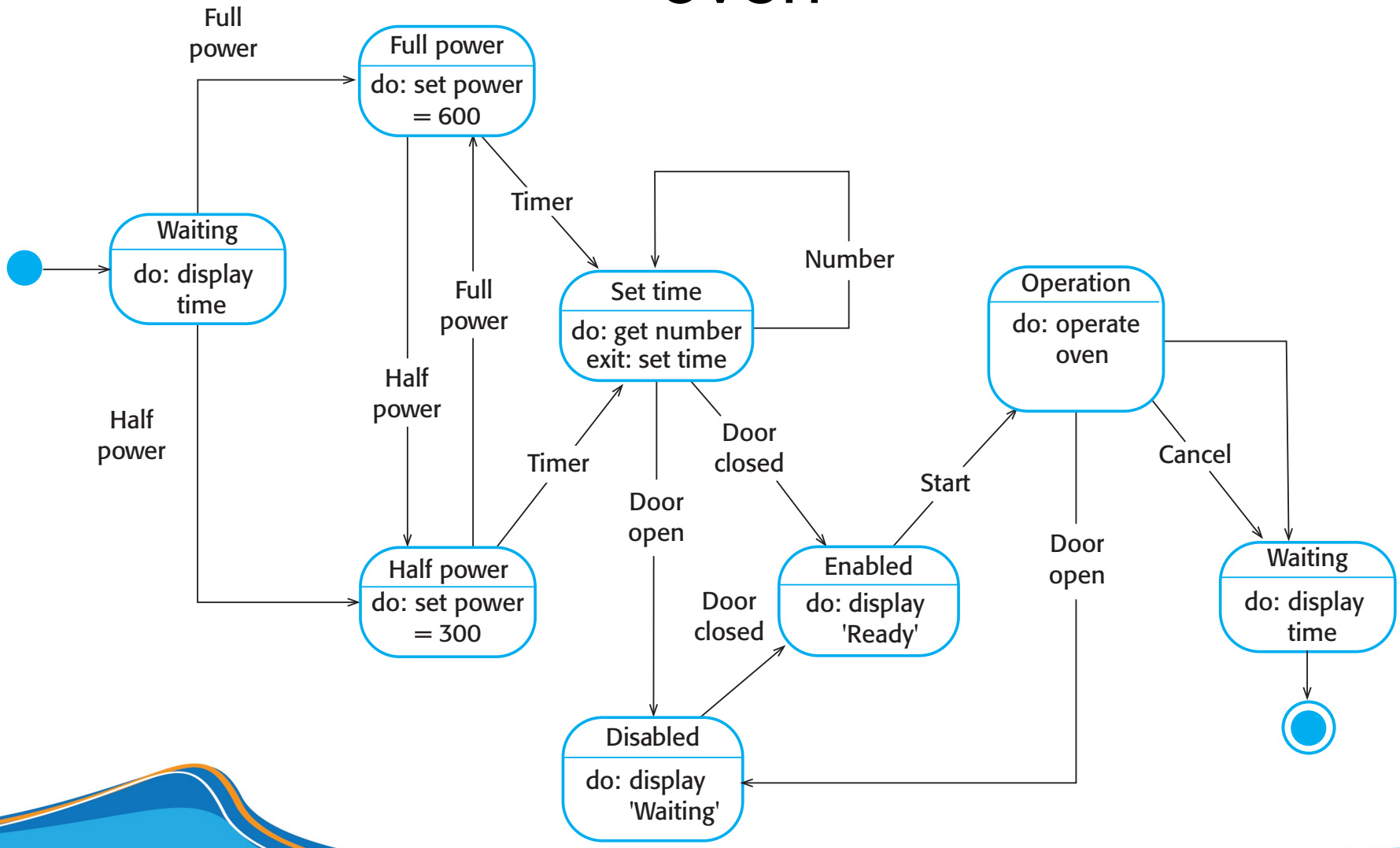
Event-driven modeling

- Real-time systems are often event-driven, with minimal data processing.
- Event-driven modeling shows how a system responds to external and internal events.
- It is based on the assumption that
  - a system has a finite number of states and
  - events (stimuli) may cause a transition from one state to another.

State machine models

- These model the behaviour of the system in response to external and internal events.
- They show the system's responses to stimuli so are often used for modelling real-time systems.
- State machine models show system states as nodes and events as arcs between these nodes.
  - When an event occurs, the system moves from one state to another.
- State charts are an integral part of the UML and are used to represent state machine models.

State diagram of a microwave oven



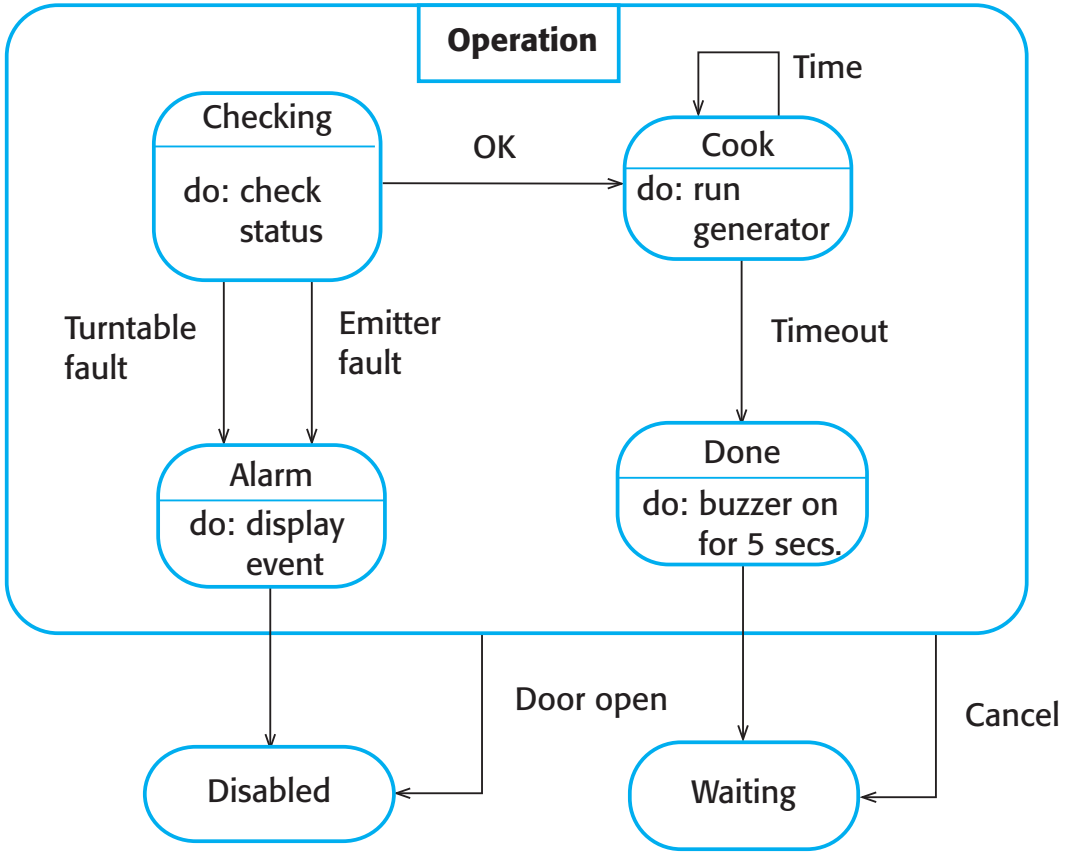
States for the microwave oven

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.

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.

Microwave oven operation



Questions?