COMP41670 Software Engineering

14. System Modelling including UML

Dr Chris Bleakley

Dr Liliana Pasquale

UCD School of Computer Science. **Scoil na Ríomheolaíochta UCD.**

Table of Contents

What is Systems Modelling?

Systems Modelling

- … is the process of developing abstract models of a system, with each model presenting a different view or perspective of the system.
- Nowadays this usually means diagrams in the Unified Modelling Language (UML). However, formal (mathematical) models of the system could be used.
- Formal models are good in that the code can be formally proven to meet the specification (formal verification). However, formal methods are slow to use. Hence they are typically only used for some safety critical systems.

Systems Modelling

- When building a new system, models are developed during requirements engineering and systems design.
- When working with an old system, models are developed to describe what it does.
- For languages like Java, there are tools available to reverse engineer UML diagrams from code.
- In a model-driven software engineering processing, it is possible to generate a complete or partial system implementation from the system model.

Model Perspectives

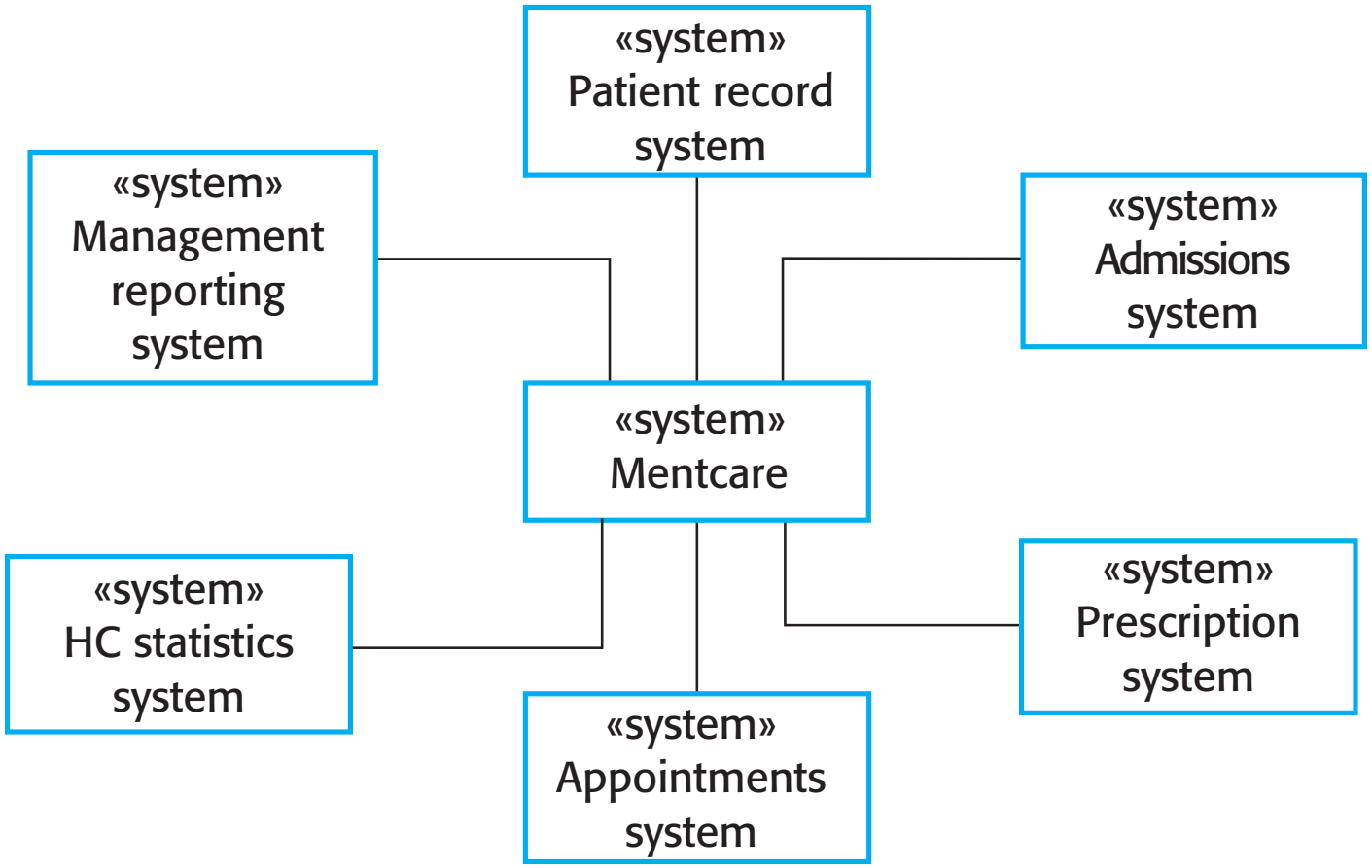
- Content Models describe the environment of the system and how systems work together.
  - Context Models and Process Models
- Interaction Models describe the how users work with the system and how the system works with other systems.
  - Use Cases and Sequence Diagrams
- Structural Models describe the architecture of the system, e.g. the components and how they relate to each other.
  - Class Diagrams
- Behavioural Models describe how the system components react to inputs and events.
  - Activity Models and State Diagrams

Context Models

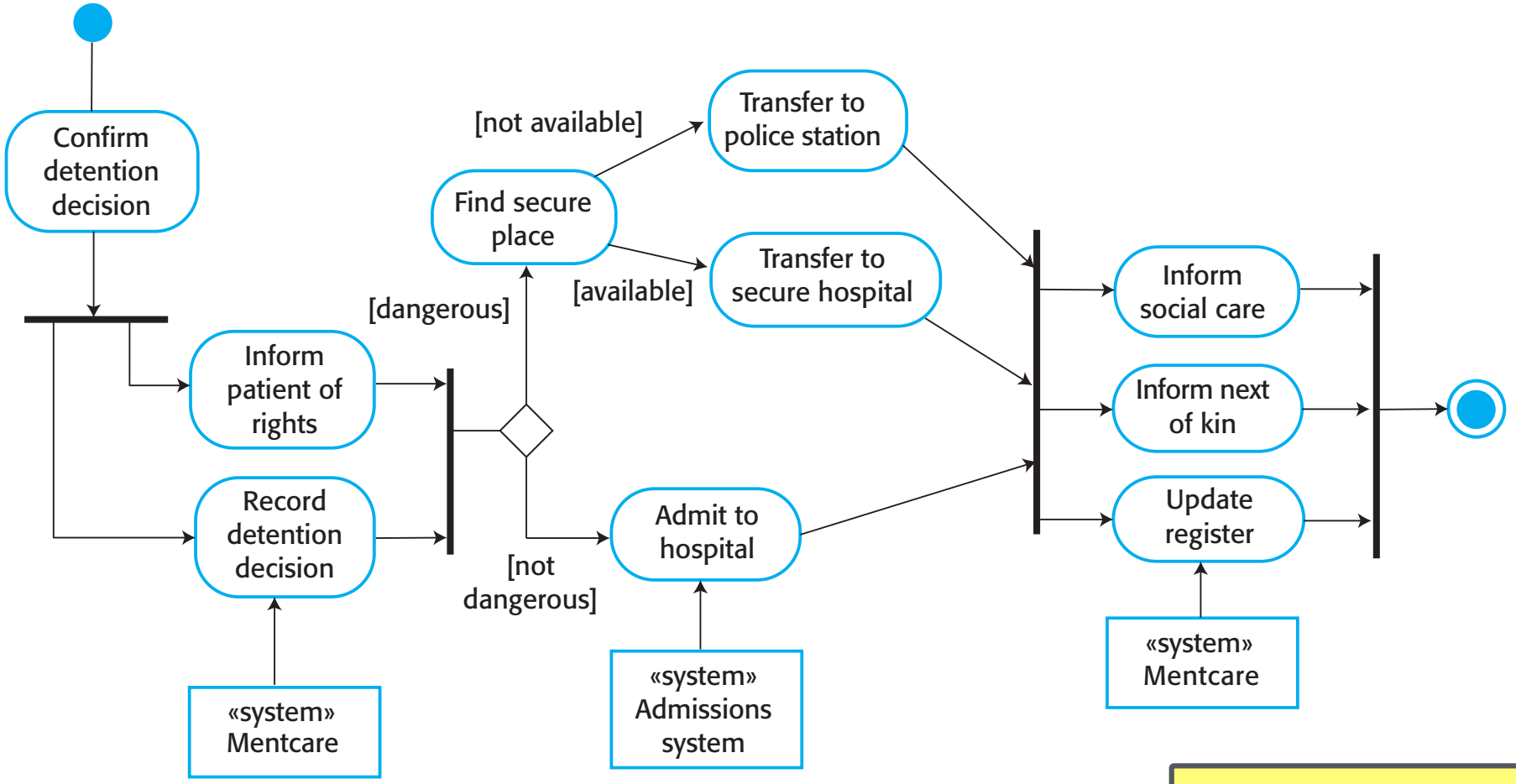
Context Models

- … are used to illustrate the relationship of a system to other systems.
- They help to define the boundary of a system, i.e. what is included and what is not.

Mentcare Example: Context Model



Mentcare Example: Process model for involuntary hospital admittance (detention)



Aka system level activity diagram

Unified Modelling Language

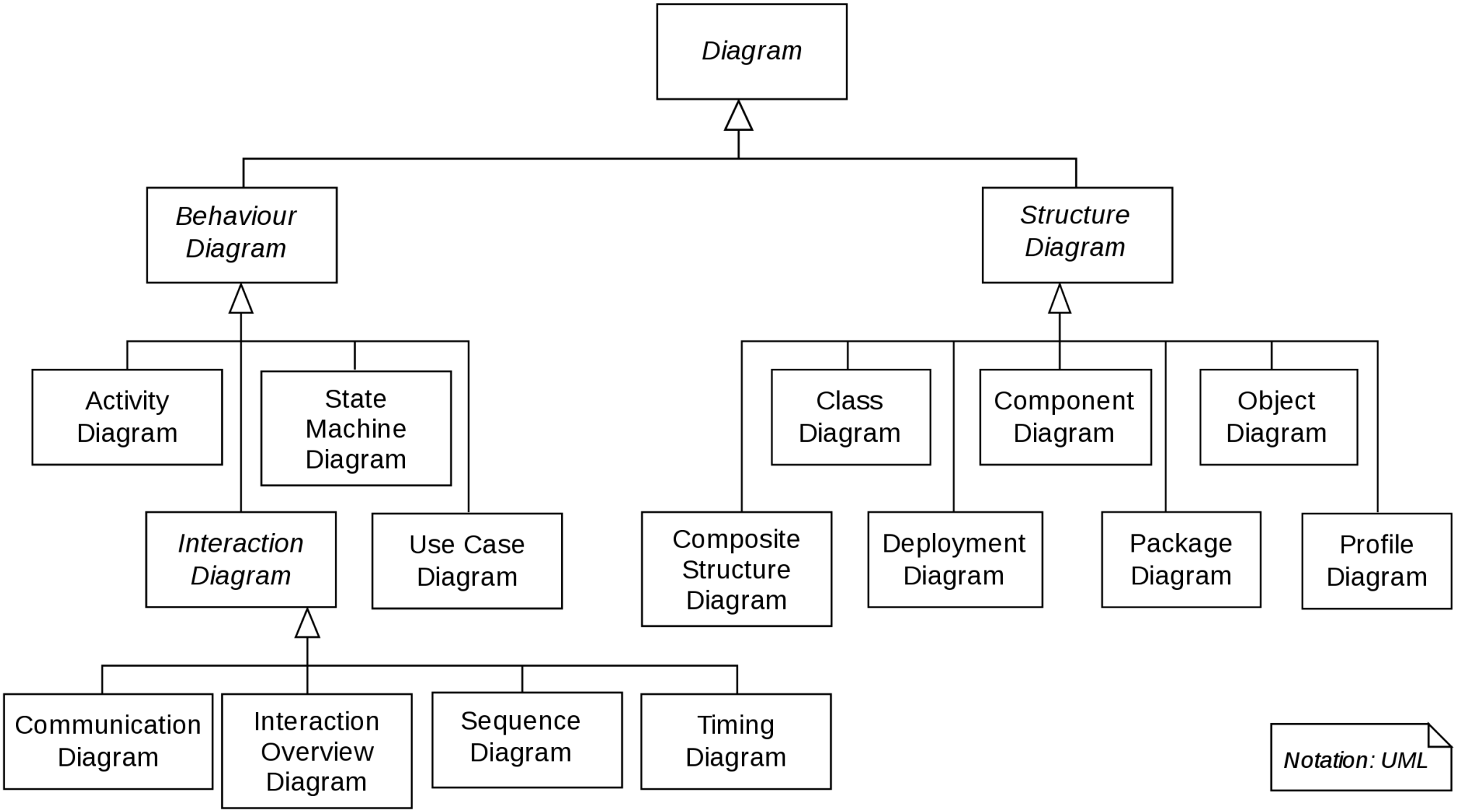
UML History

- Grady Booch, Ivar Jacobson, and James Rumbaugh created the Unified Modeling Language in 1995 while working at Rational software.
- In 1997, the Object Management Group adopted UML as a standard for its members, including Hewlett-Packard, IBM, and Apple Computer.
- In 2005, the language was published by the International Organization for Standardisation (ISO) and has since been revised and reviewed to keep it up to date.
- UML 2.5.1 was released in 2017.

https://www.omg.org/spec/UML

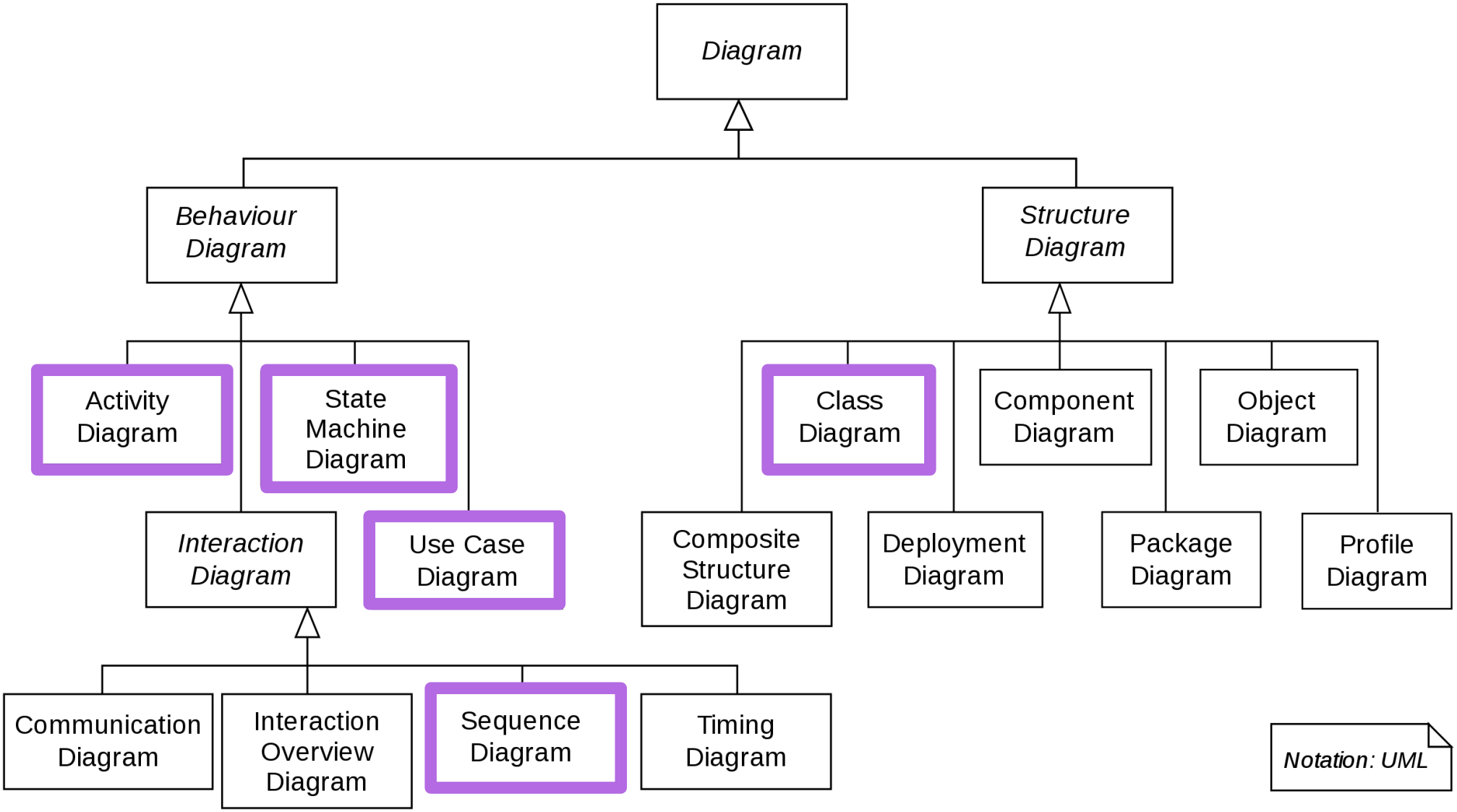
https://www.omg.org/spec/UML/2.5.1/PDF

UML Diagram Hierarchy



https://commons.wikimedia.org/wiki/File:UML_

UML Diagram Hierarchy



https://commons.wikimedia.org/wiki/File:UML_diagrams_overview.svg

Important Types of UML Diagrams

- Class diagrams: show the object classes in the system and the associations between these classes.
- Use case diagrams: show the interactions between actors and the system.
- Activity diagrams: show the activities involved in a process.
- State diagrams: show how the system moves through states in response to internal and external events.
- Sequence diagrams: show sequences of interactions between actors, system and objects over time.

https://circle.visual-paradigm.com/category/uml-diagrams/

UML Diagrams

- No diagram type is mandatory for a project.
- There is no set order in which diagrams are developed on a project.
- UML is a set of tools.
- Projects should use whatever is useful for communicating between team members and stakeholders.

https://circle.visual-paradigm.com/category/uml-diagrams/

Behavioural Models

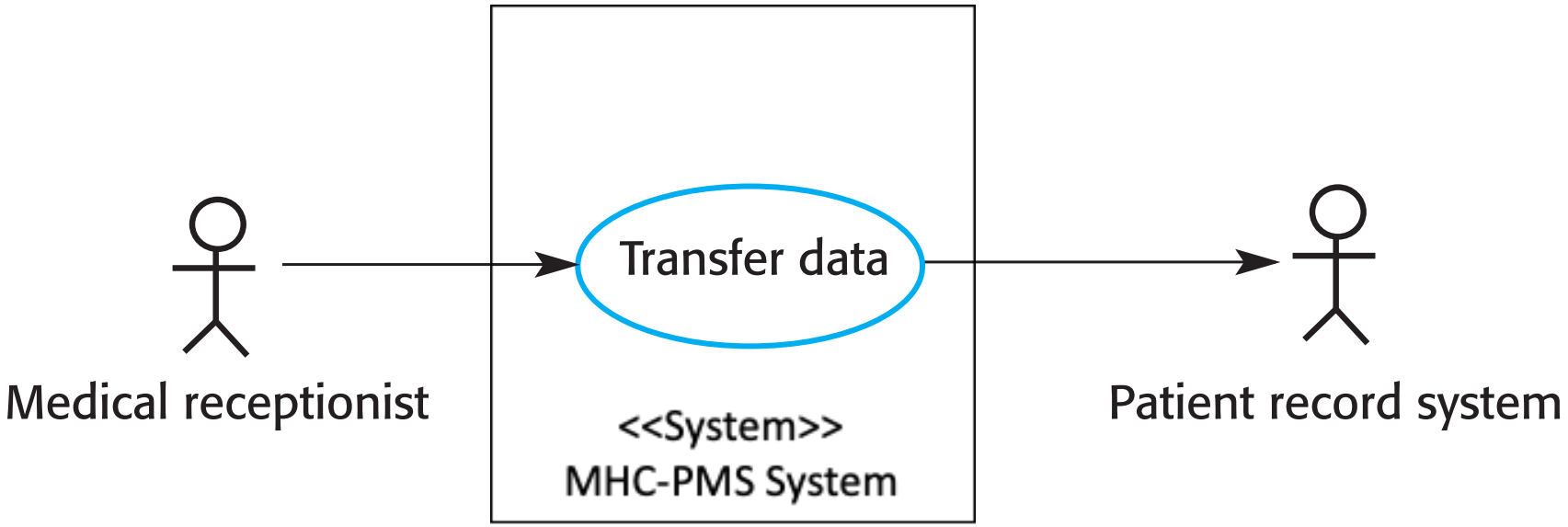
Behavioural Models

- … are models of the dynamic behaviour of a system as it is executing.
- They show what happens (or should happen) when a system responses to a stimulus. There are 2 types of stimuli: data and events (e.g. hardware interrupt).
  - Use case diagrams: describe the tasks that actors can initiate.
  - Activity diagrams: describe the flow of functions that the system does.
  - State diagrams: describe the system states and transitions between states triggered by events.
- Activity diagrams and state diagrams are similar to flow charts.
- An activity diagram is a flow of functions without trigger (event) mechanisms, a state machine model consists of triggered states.

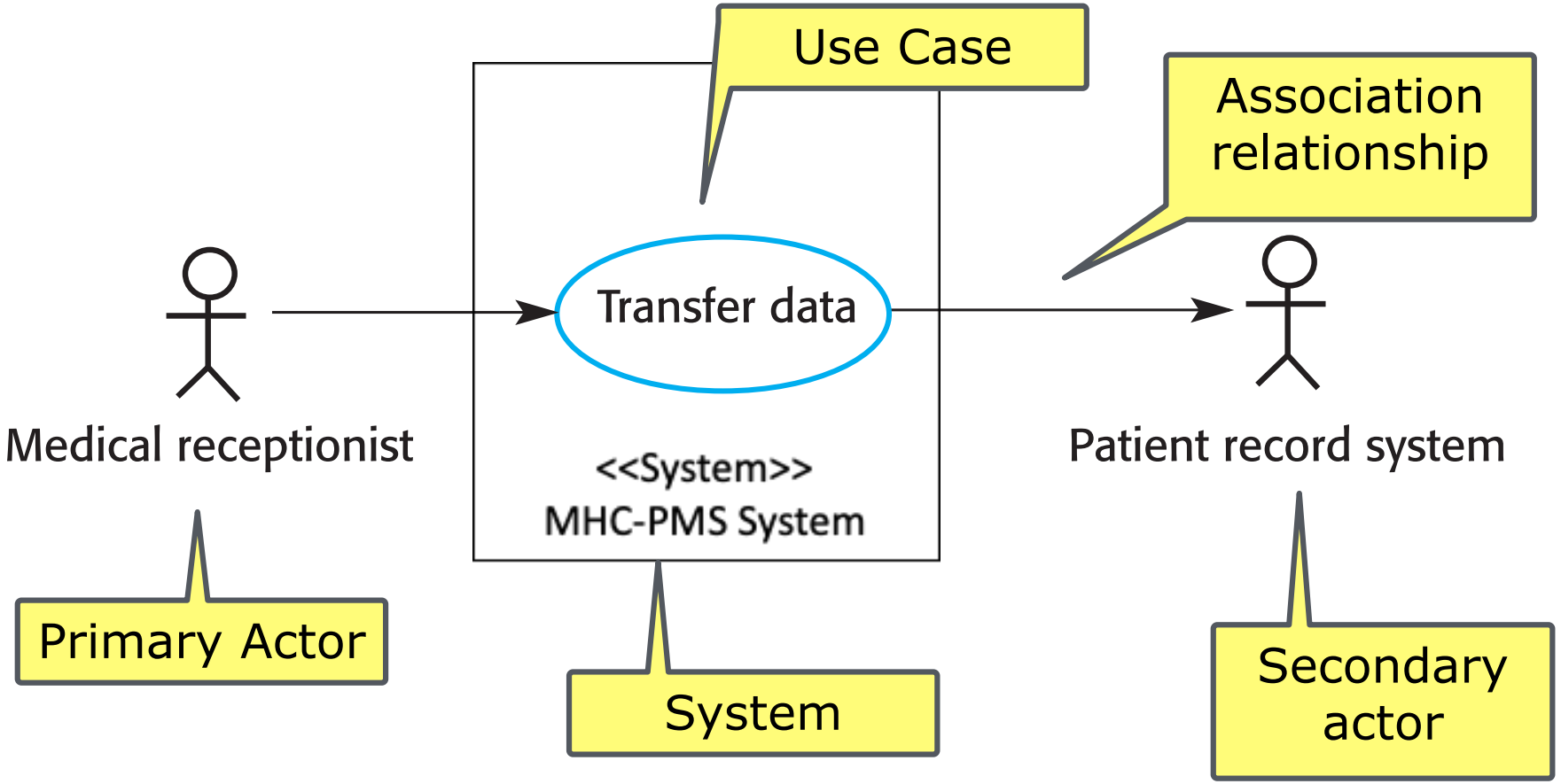
Use Case Diagrams

- Each use case represents a discrete task that involves an actor.
- An actor in a use case may be a person or an external system.

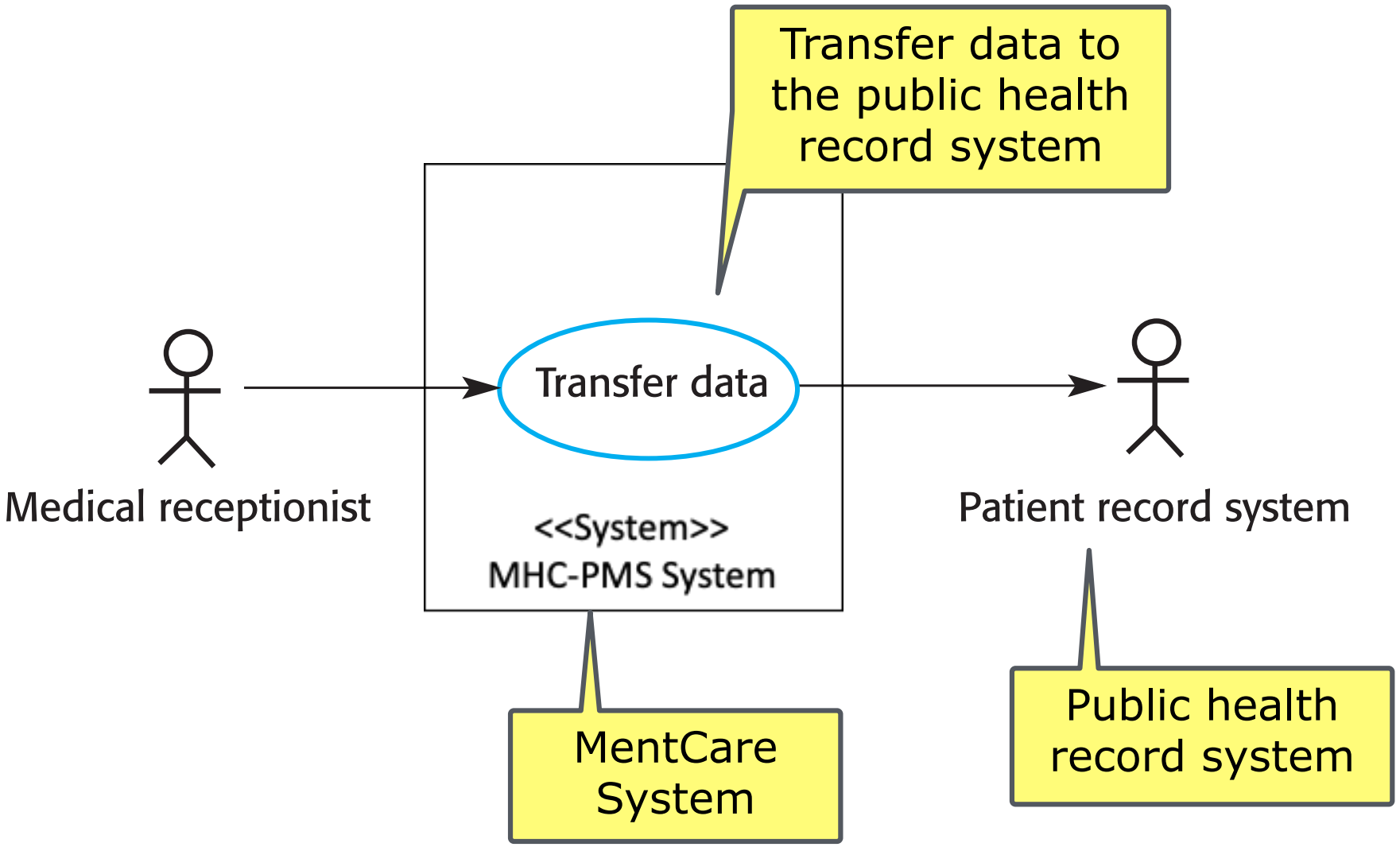
Mentcare Example: Transfer data use case diagram



Mentcare Example: Transfer data use case diagram



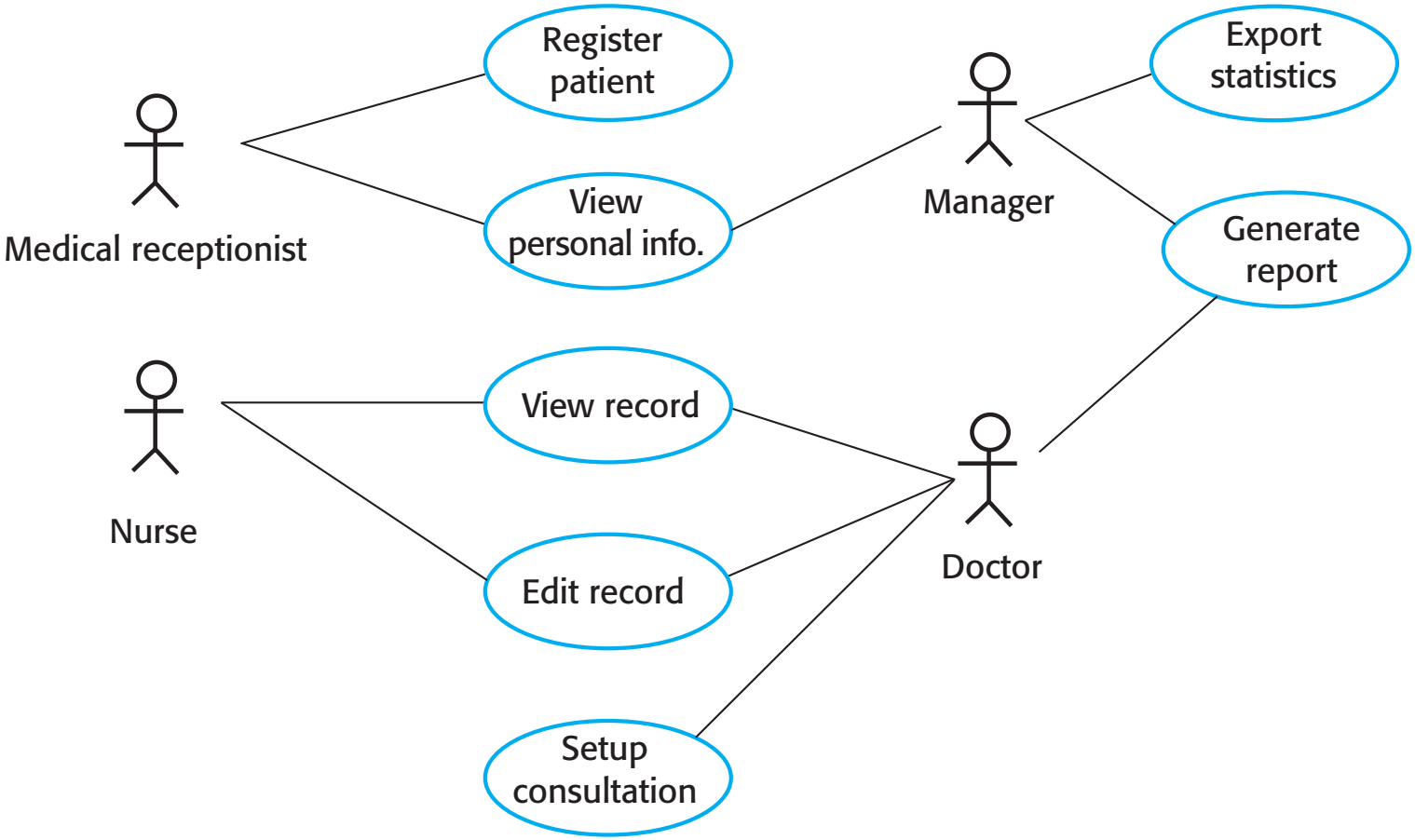
Mentcare Example: Transfer data use case diagram



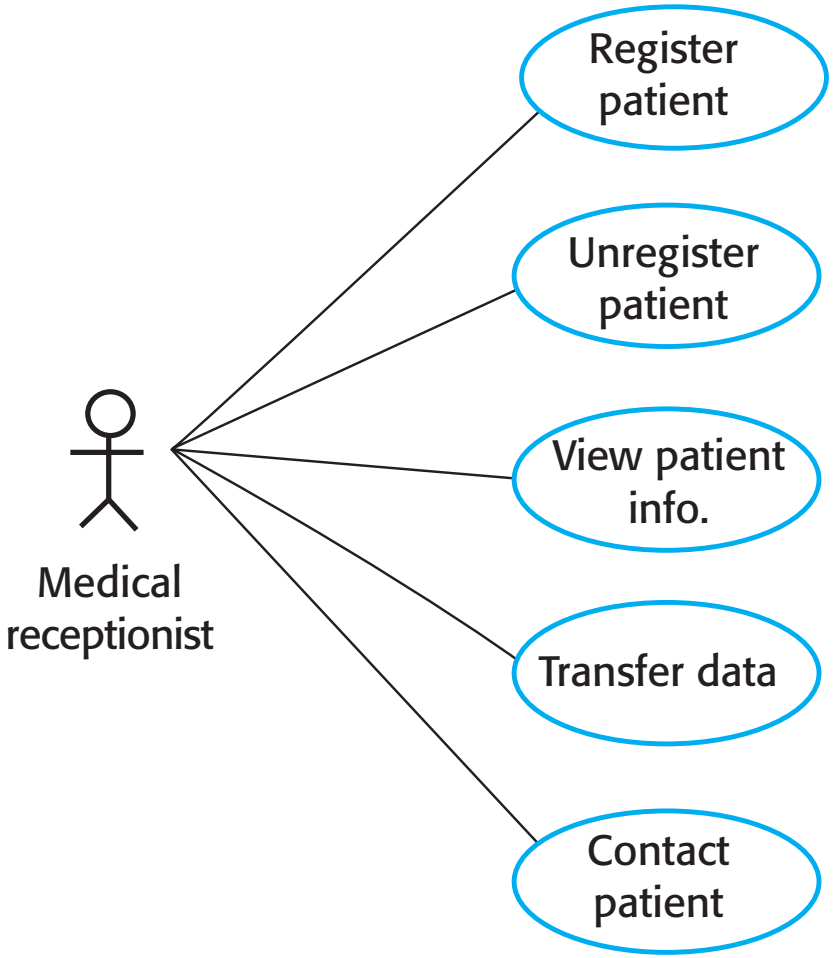
Mentcare Example: Transfer data use case text

MHC-PMS: 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 either be updated personal information (address, phone number, etc.) 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.

Mentcare Example: Use case diagram

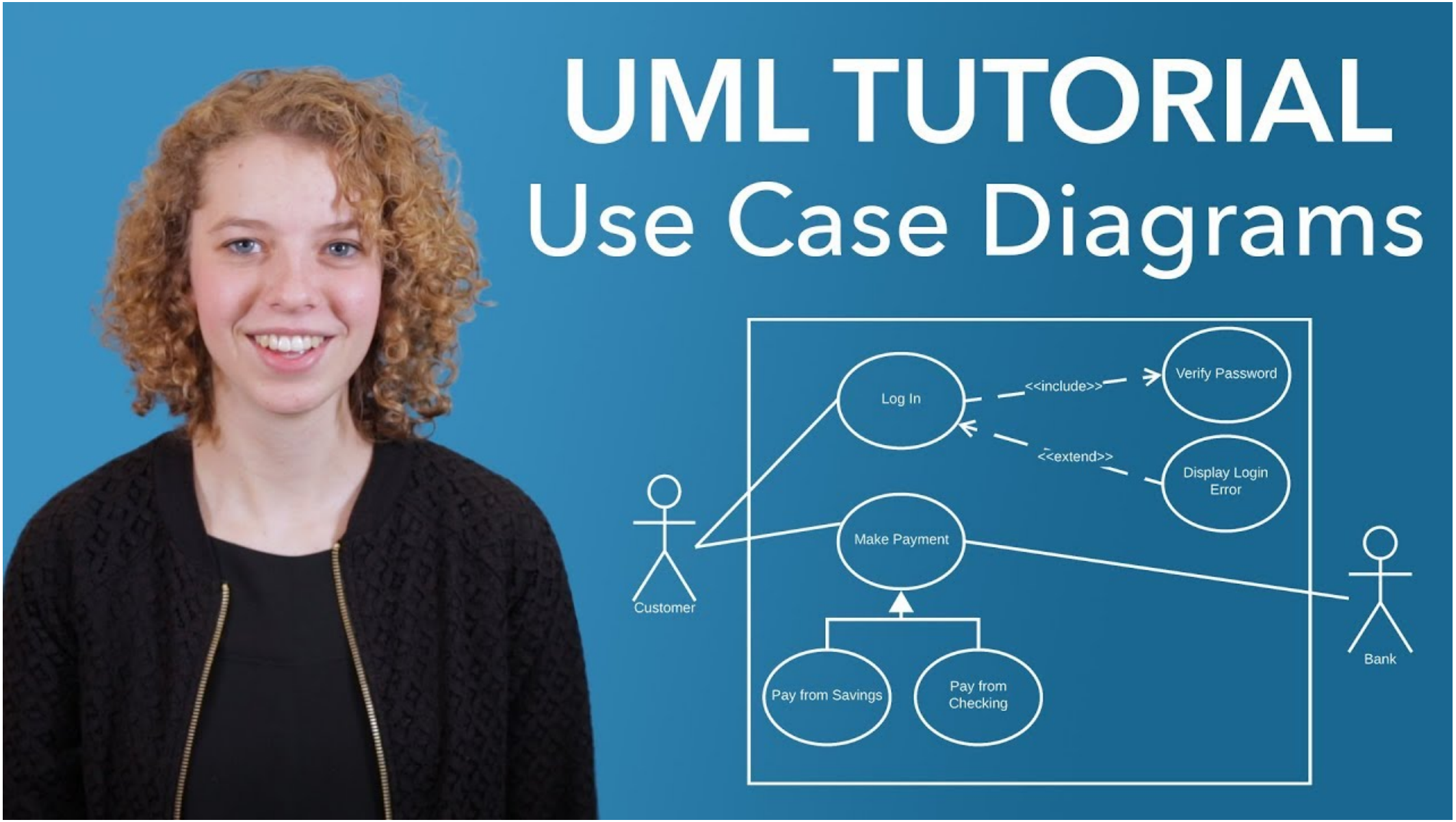


Mentcare Example: Use cases involving the role "medical receptionist"



Composite role view

UML Use Case Diagram Tutorial, Lucid Software, YouTube

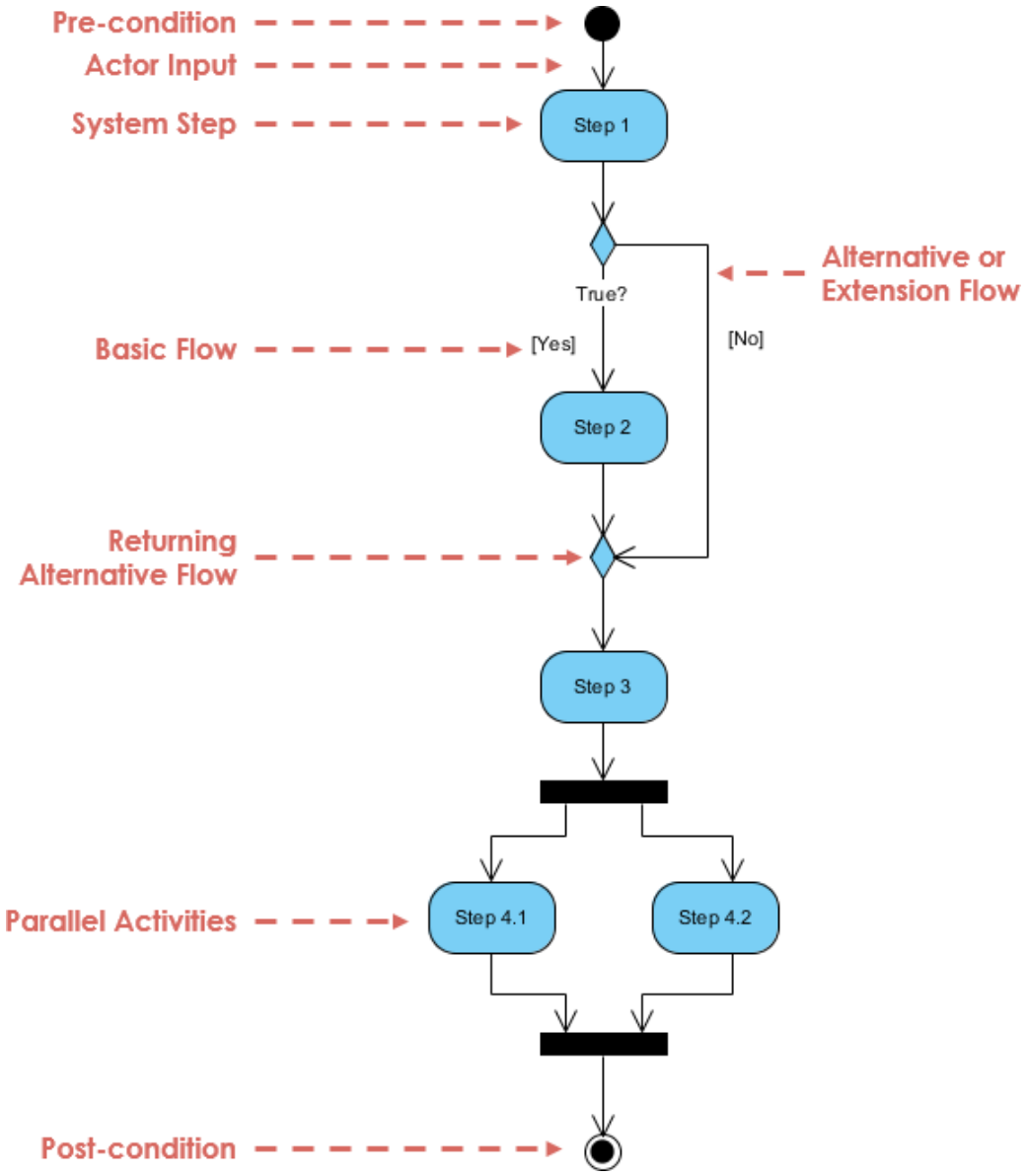


Explains includes, extends, and generalisation (inheritance) relationships

Activity Diagrams

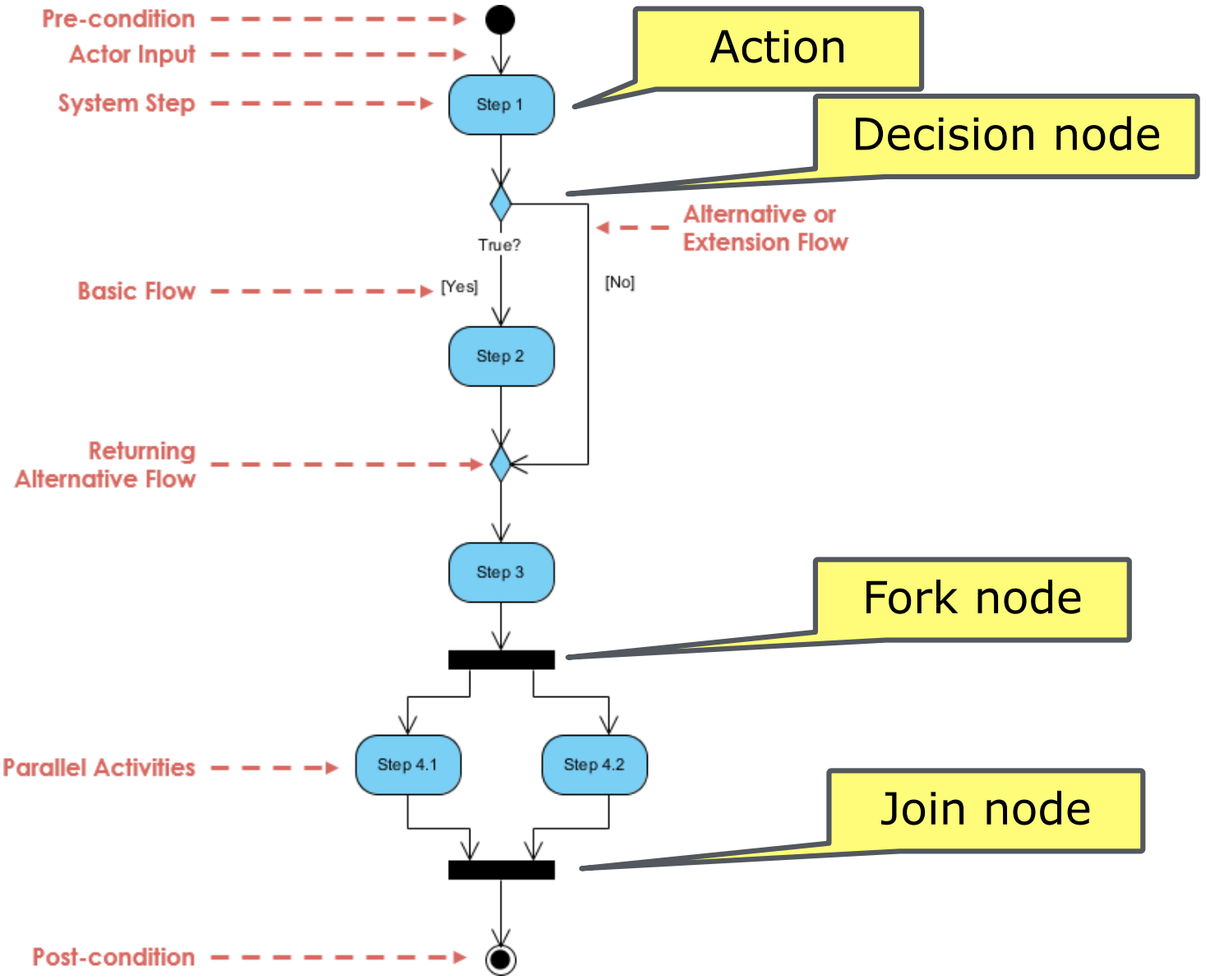
- A use case task is broken down into sequences of activities.
- An activity is a sequence of actions.

Insulin Pump Example: Activity Diagram



https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-activity-diagram/

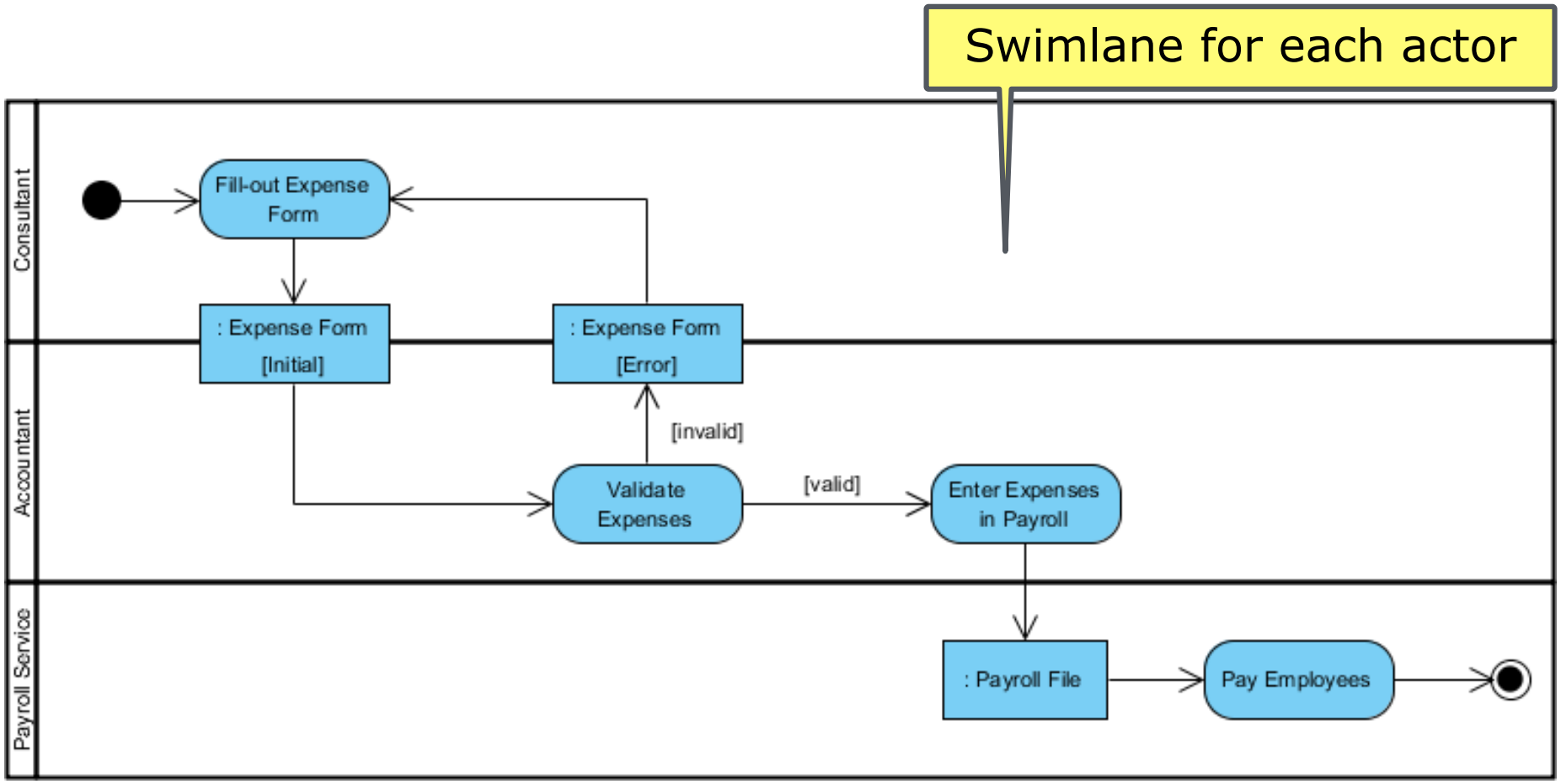
Insulin Pump Example: Activity Diagram



An activity is a set of actions

https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-activity-diagram/

Accountancy Example: Activity diagram

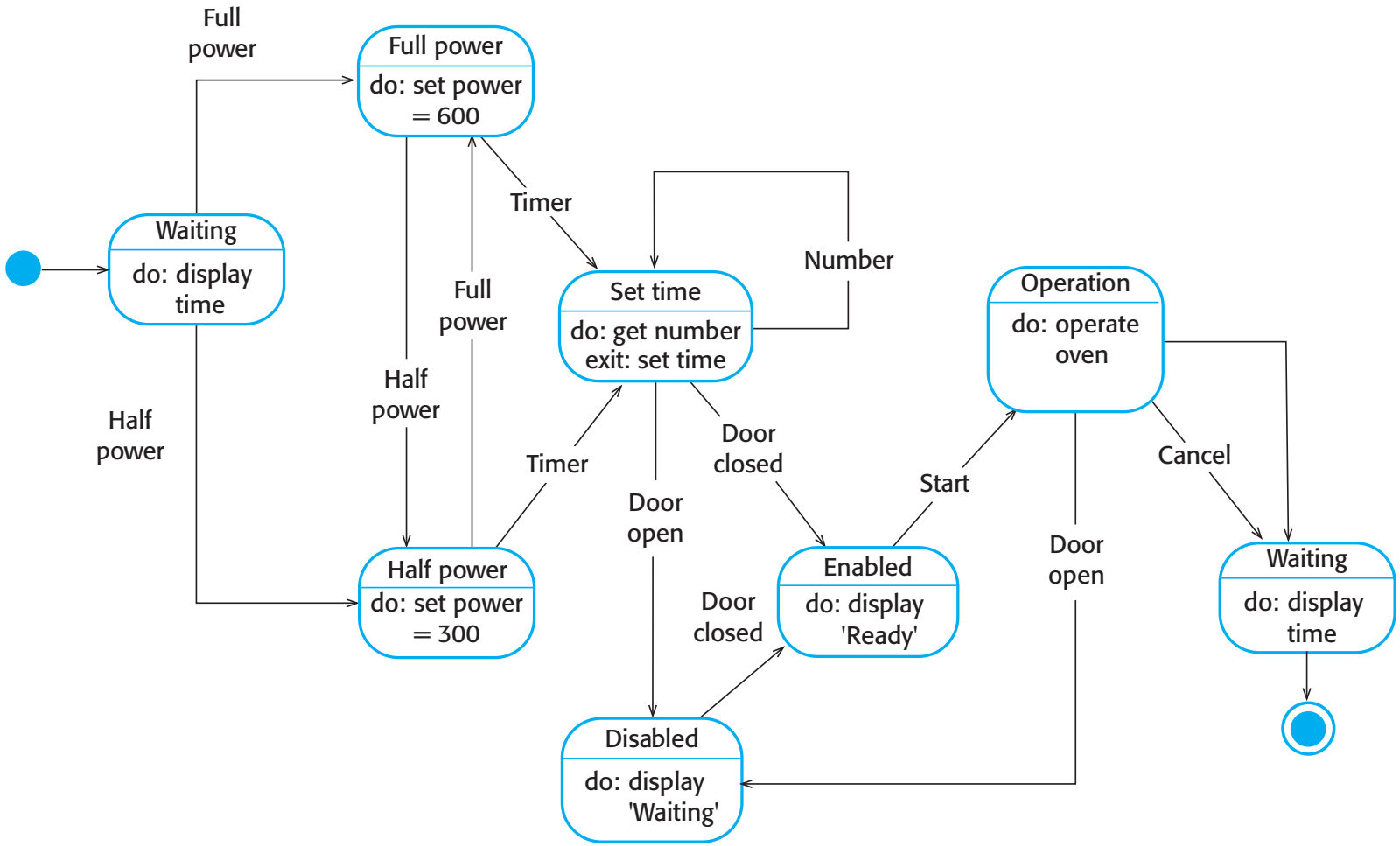


https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-activity-diagram/

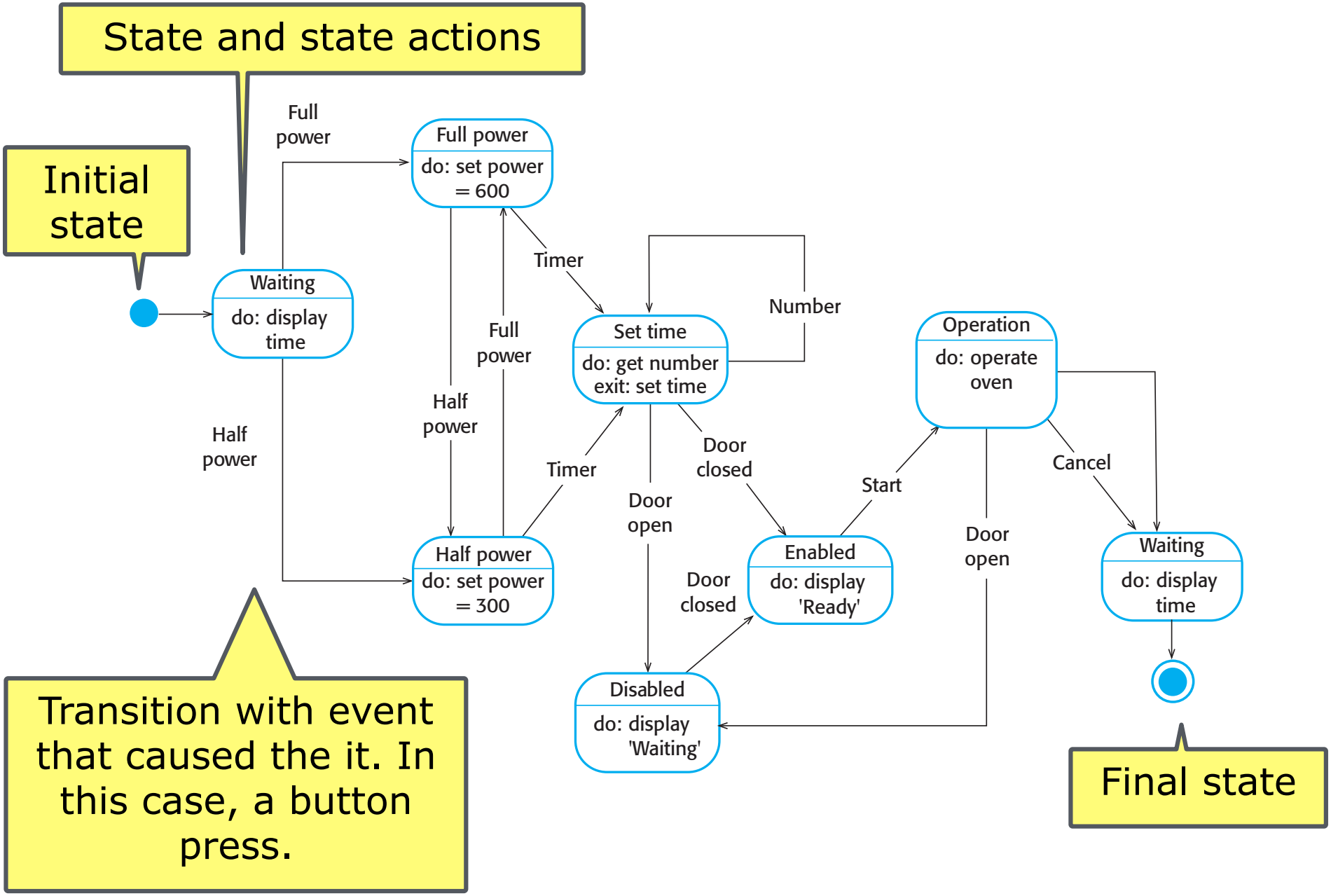
State Diagrams

- An activity diagram can be refined to create a state diagram.
- The state diagram shows system states (nodes) and transitions between states triggered by events.

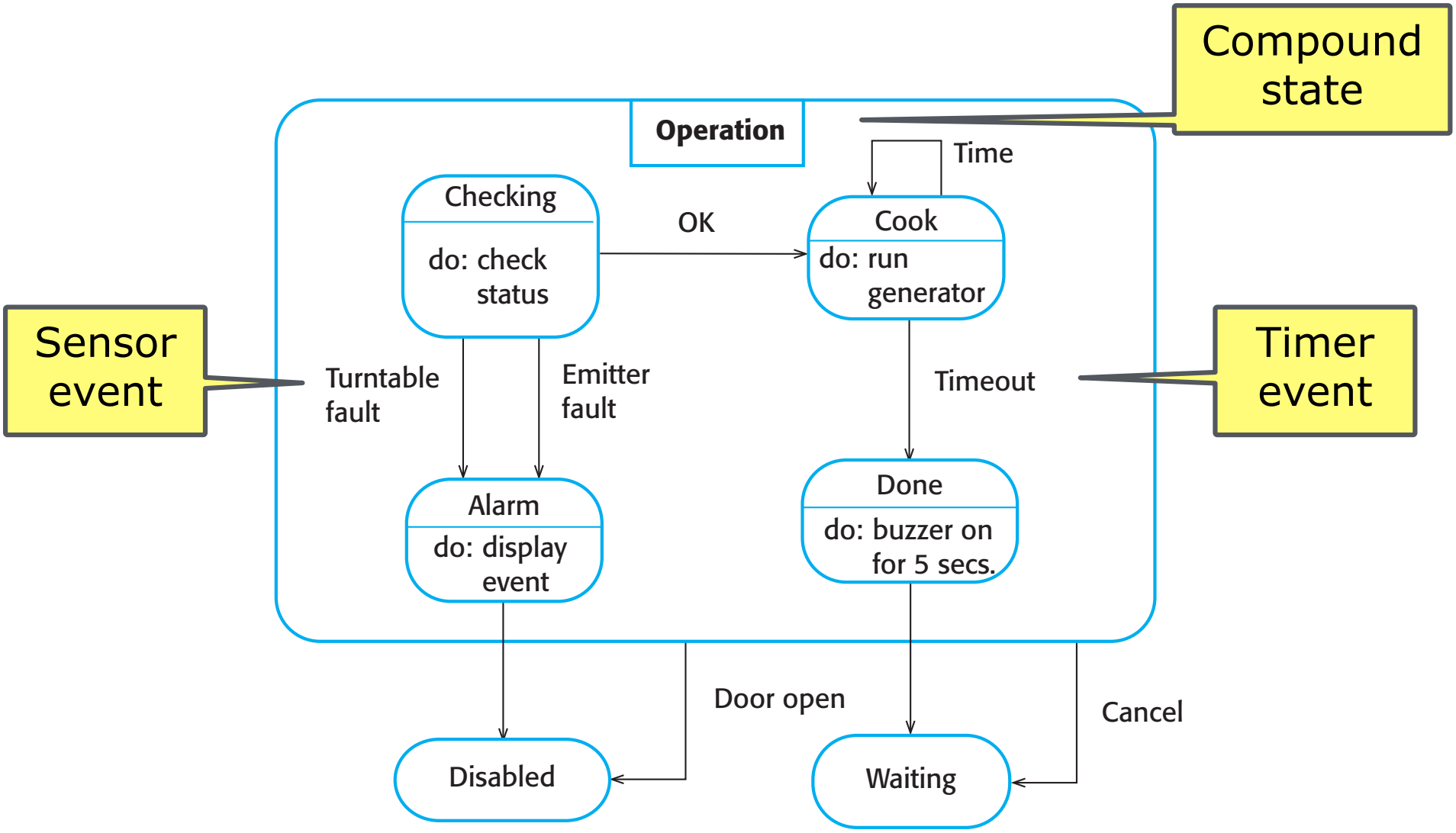
Microwave Oven Example: State diagram



Microwave Oven Example: State diagram



Microwave Oven Example: State diagram for Operation compound state



Microwave Oven Example: States and events 1/2

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

Microwave Oven Example: States and events 2/2

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.

Structural Models

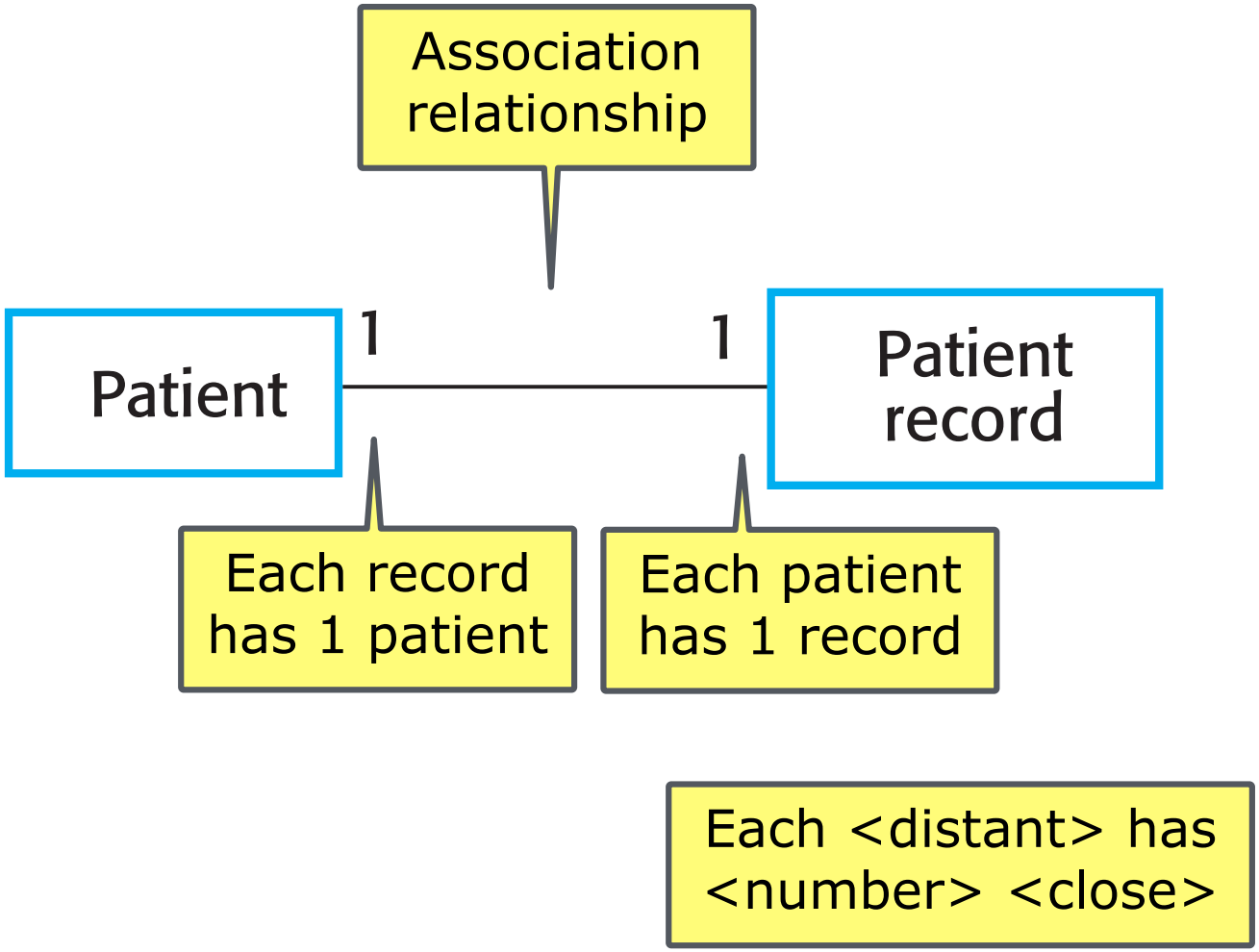
Structural Models

- … show the organisation of a system in terms of the components that make up the system and their relationships.
- Structural models describe the system architecture.
- Static models show the structure of the system design.
- Dynamic models show the organisation of the system when it is executing. Details sequence models can be used for this.

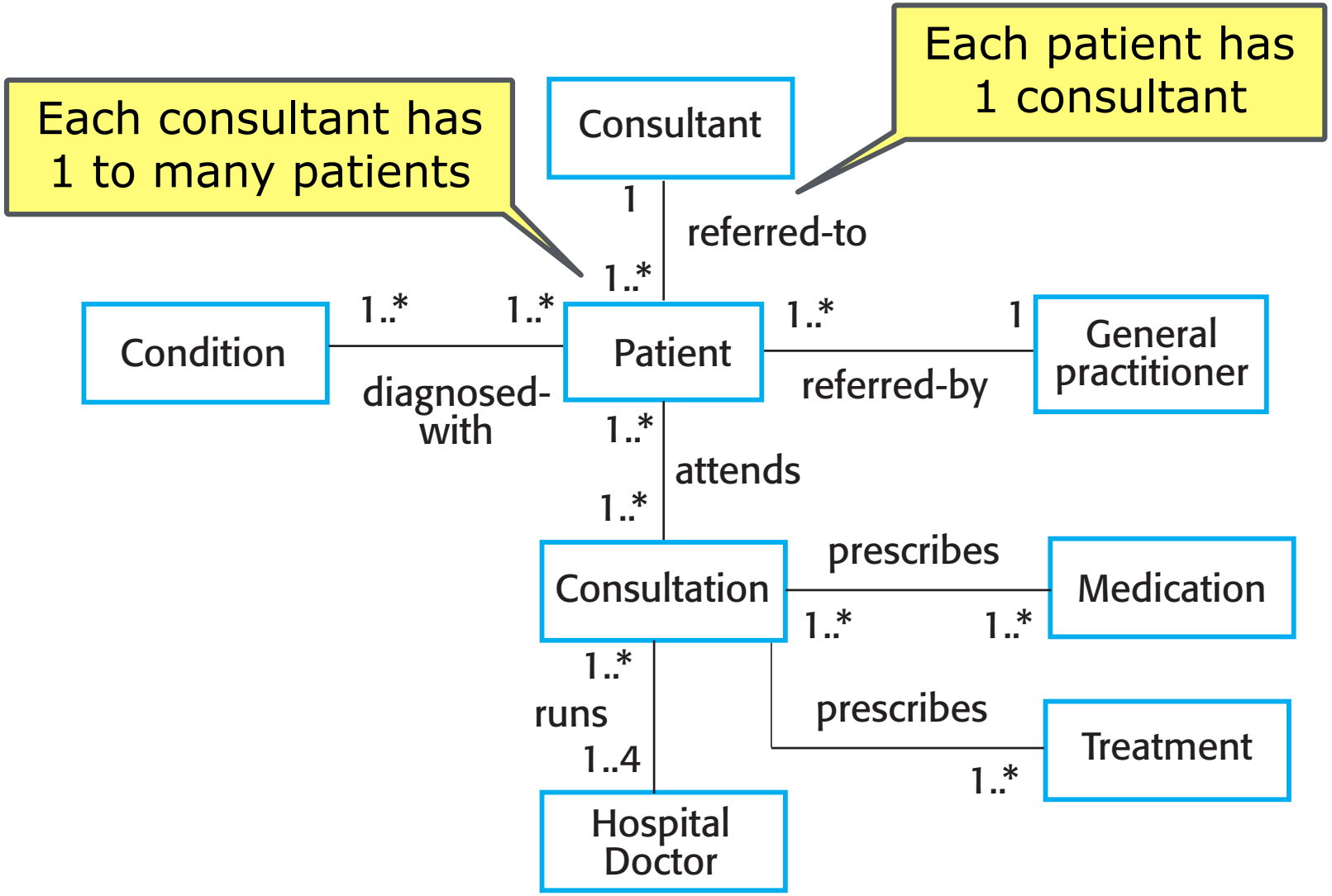
Class Diagrams

- … show the classes in a system and the associations between them.

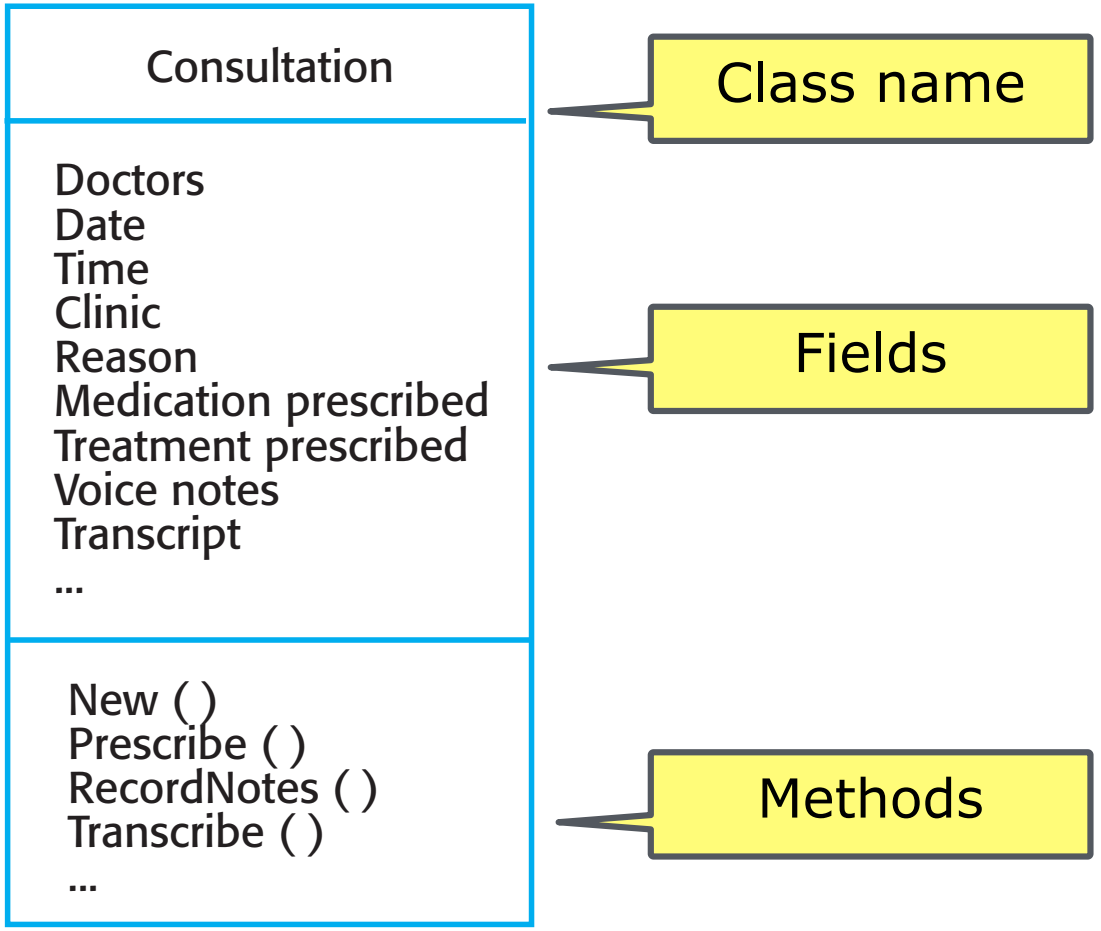
Mentcare Example: Class Diagram



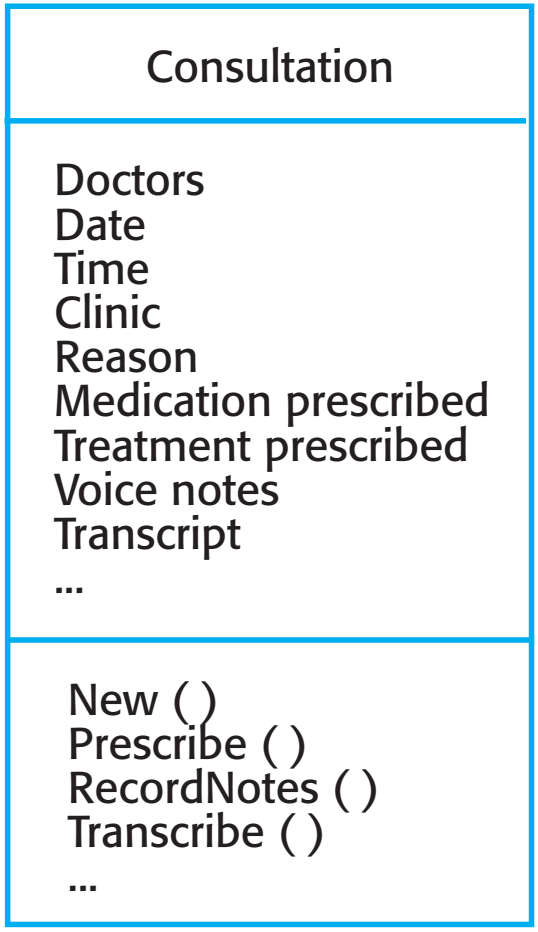
Mentcare Example: Class Diagram



Mentcare Example: Consultant Class

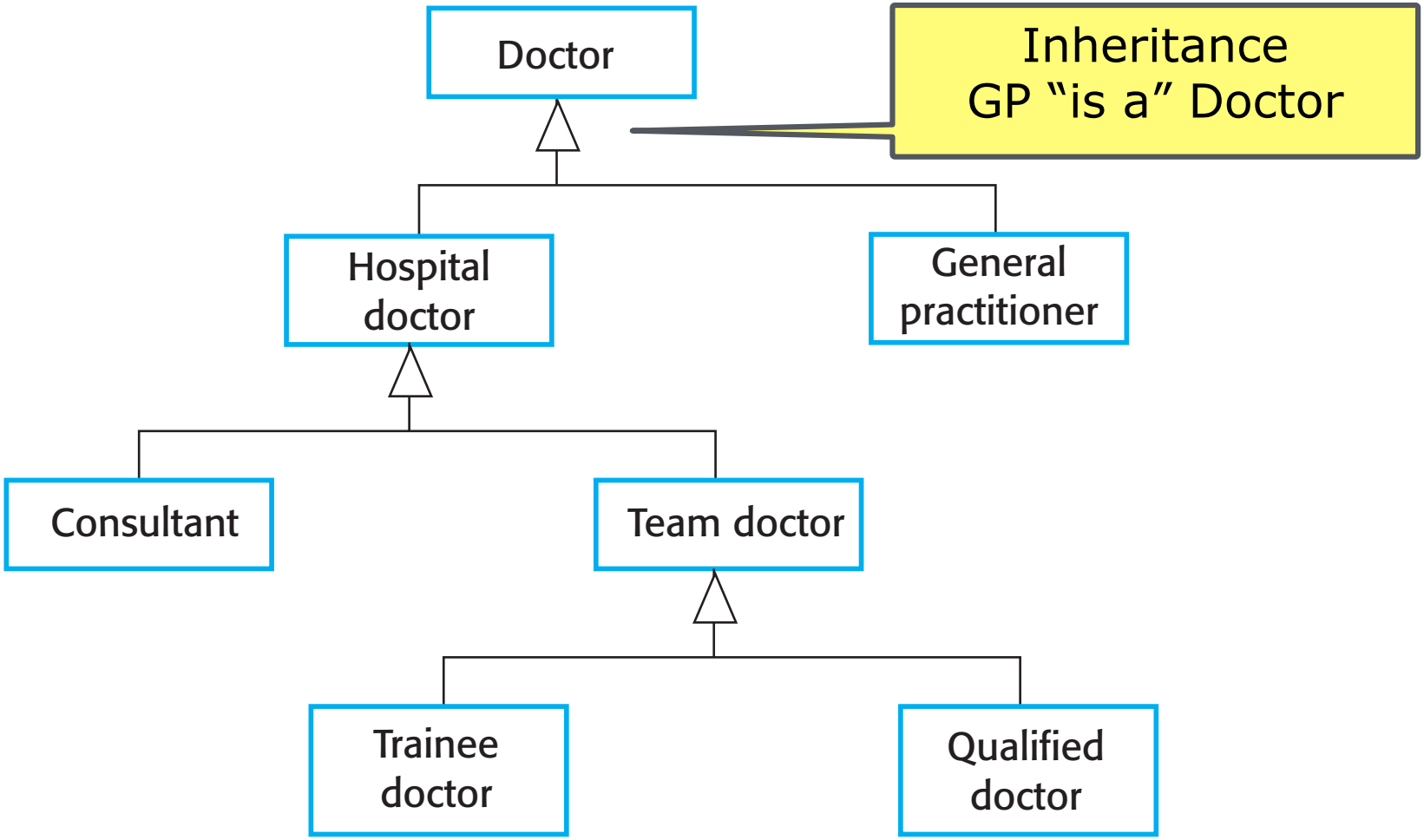


Mentcare Example: Consultant Class

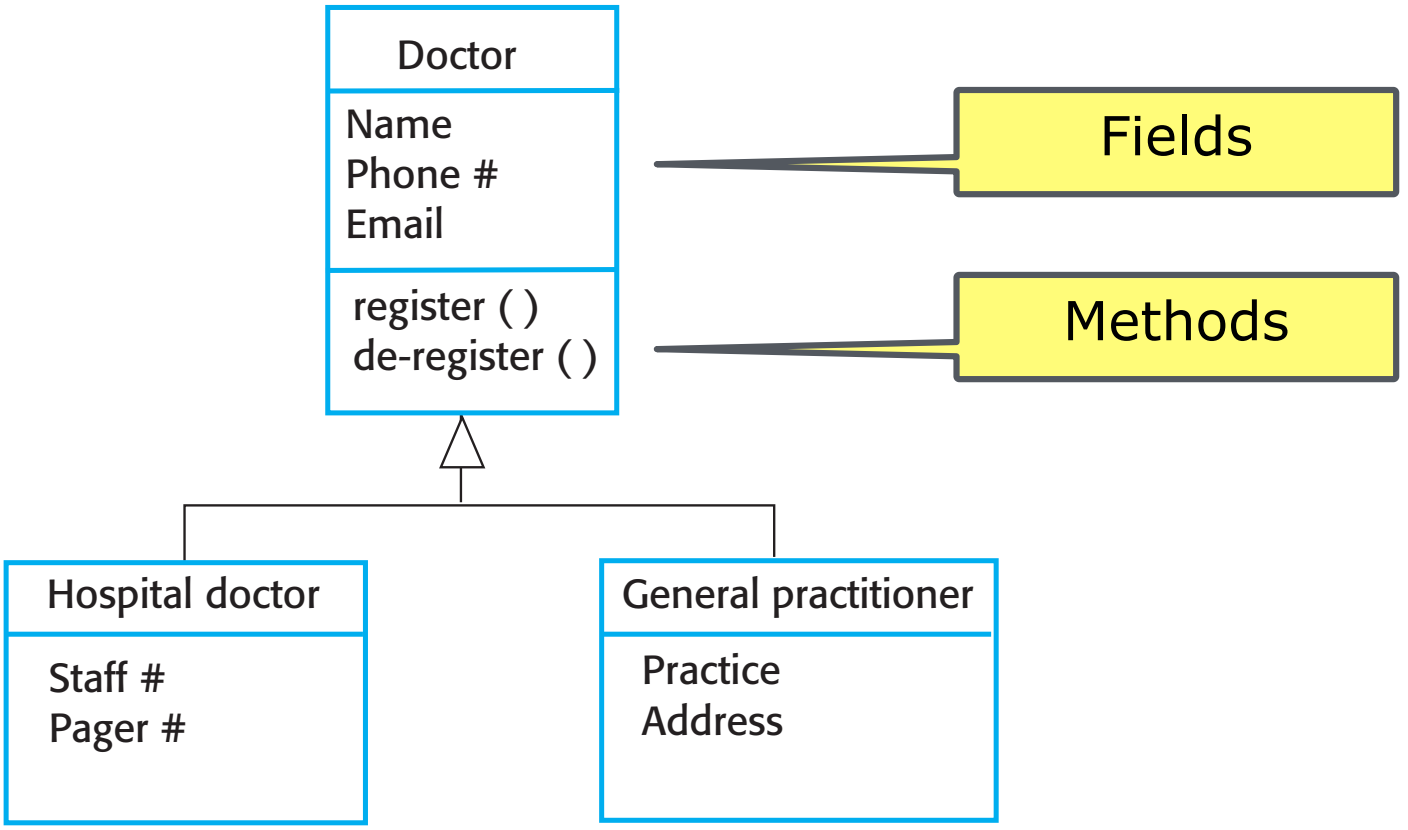


Visibility prefixes:
- private
+ public
# protected (same class or subclass only)
~ package

Mentcare Example: Inheritance (aka generalisation) hierarchy

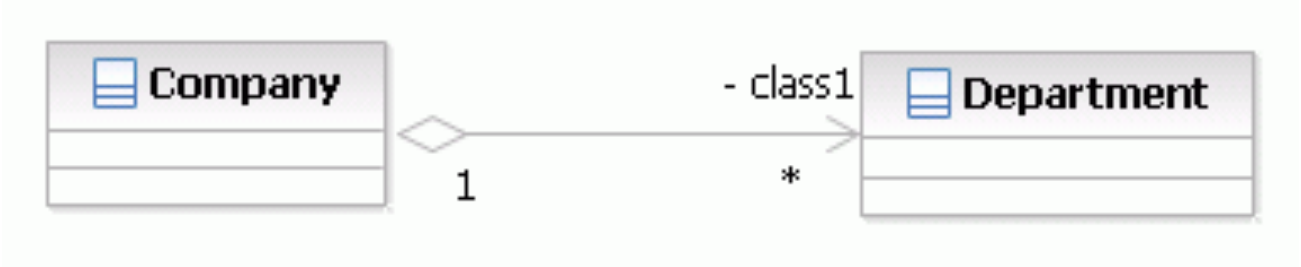


Mentcare Example: Inheritance (aka generalisation) hierarchy with detail



Aggregation

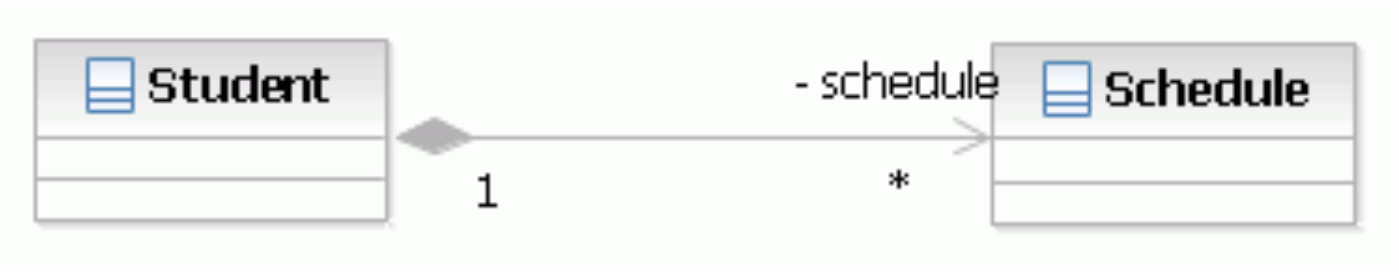
Aggregation relationship.
The aggregate groups an assembly of objects.
E.g. a company is an aggregate of many departments.



https://www.ibm.com/docs/en/rsm/7.5.0?topic=diagrams-aggregation-relationships

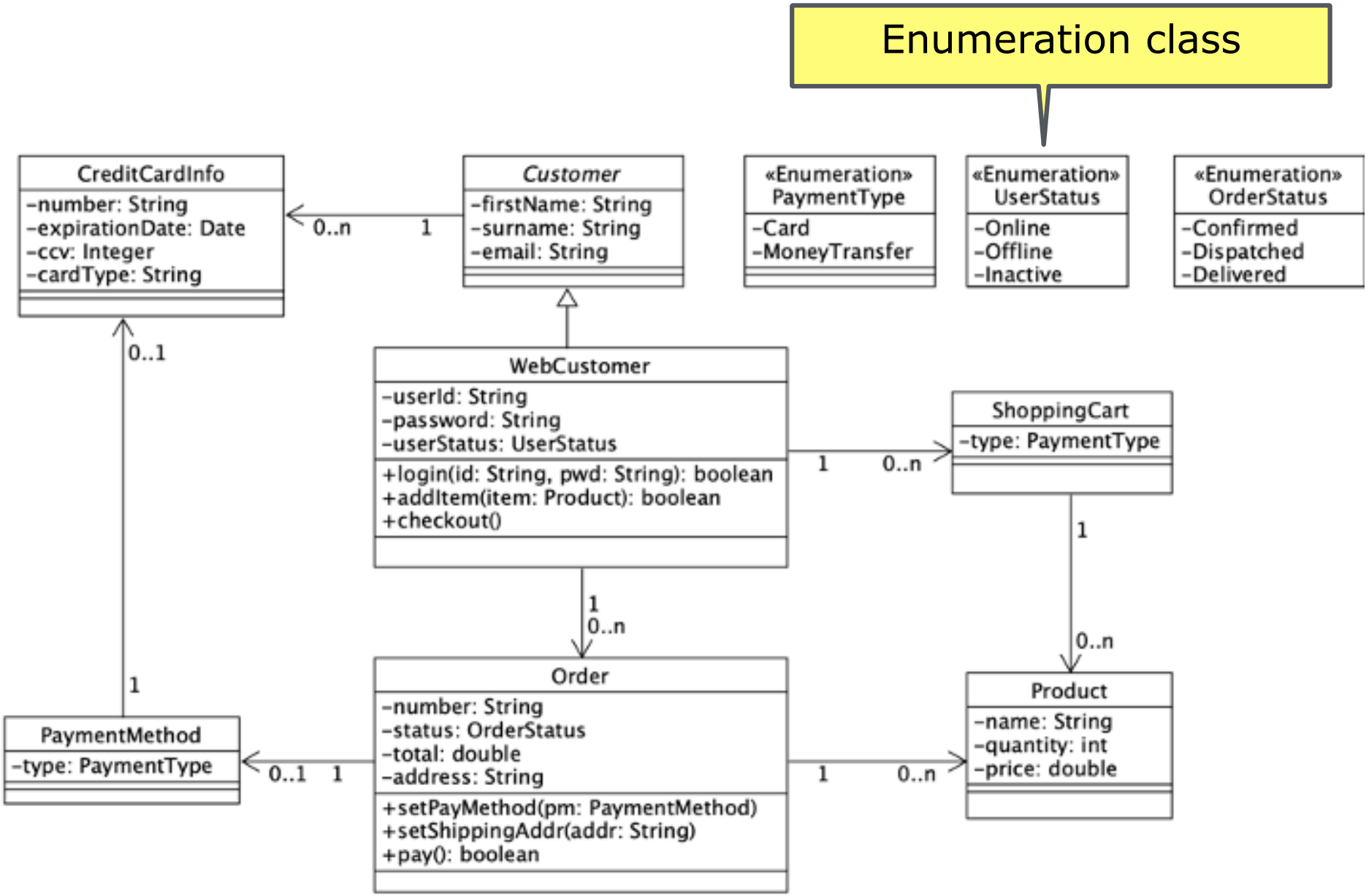
Composition

Composition relationship.
This is a form of aggregation in which the part class's lifetime is dependent on the lifetime of the whole classifier.

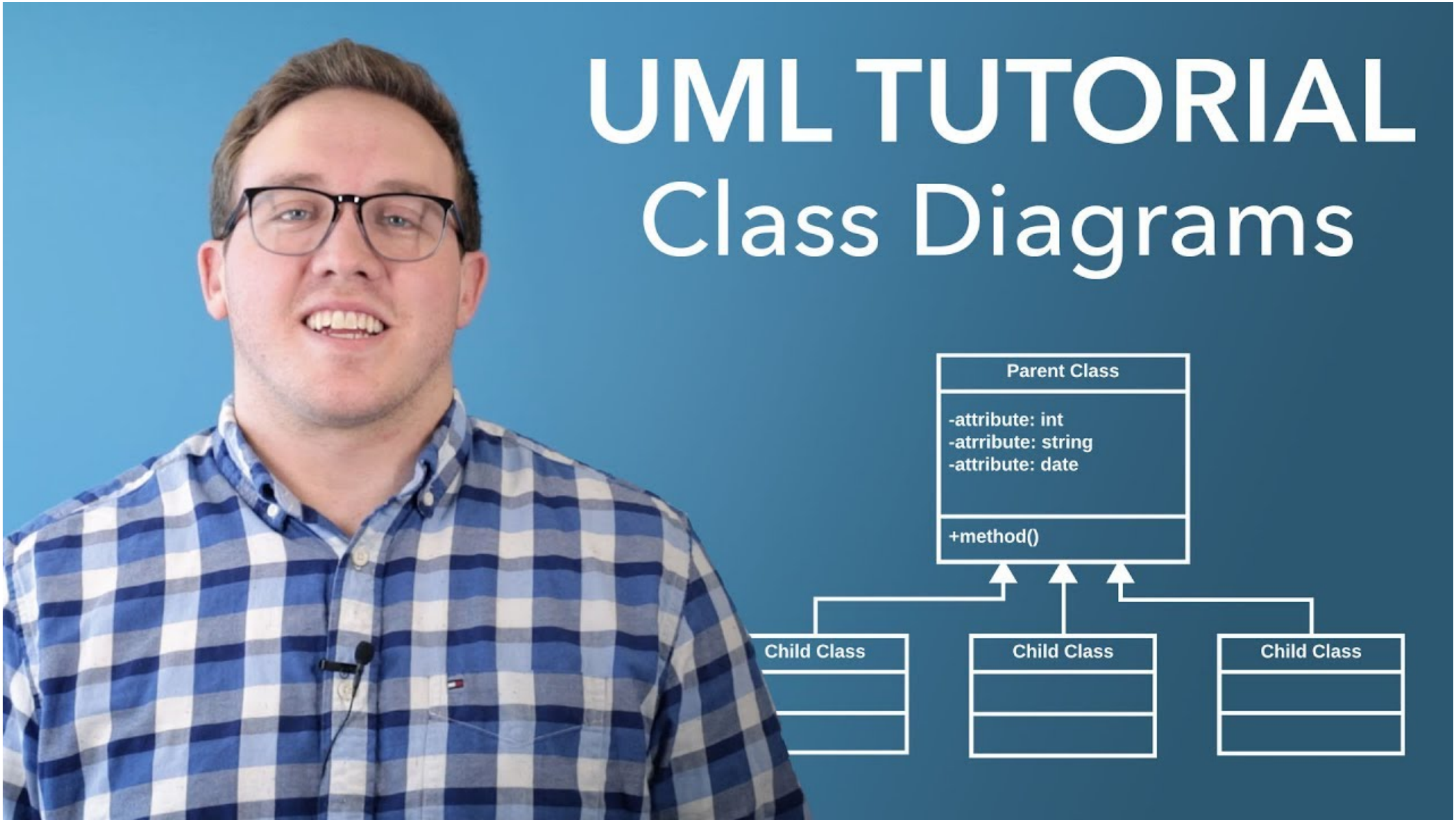


https://www.ibm.com/docs/en/rsm/7.5.0?topic=diagrams-composition-association-relationships

Class Diagram Example



UML Class Diagram Tutorial, Lucid Software, YouTube



Interaction Models

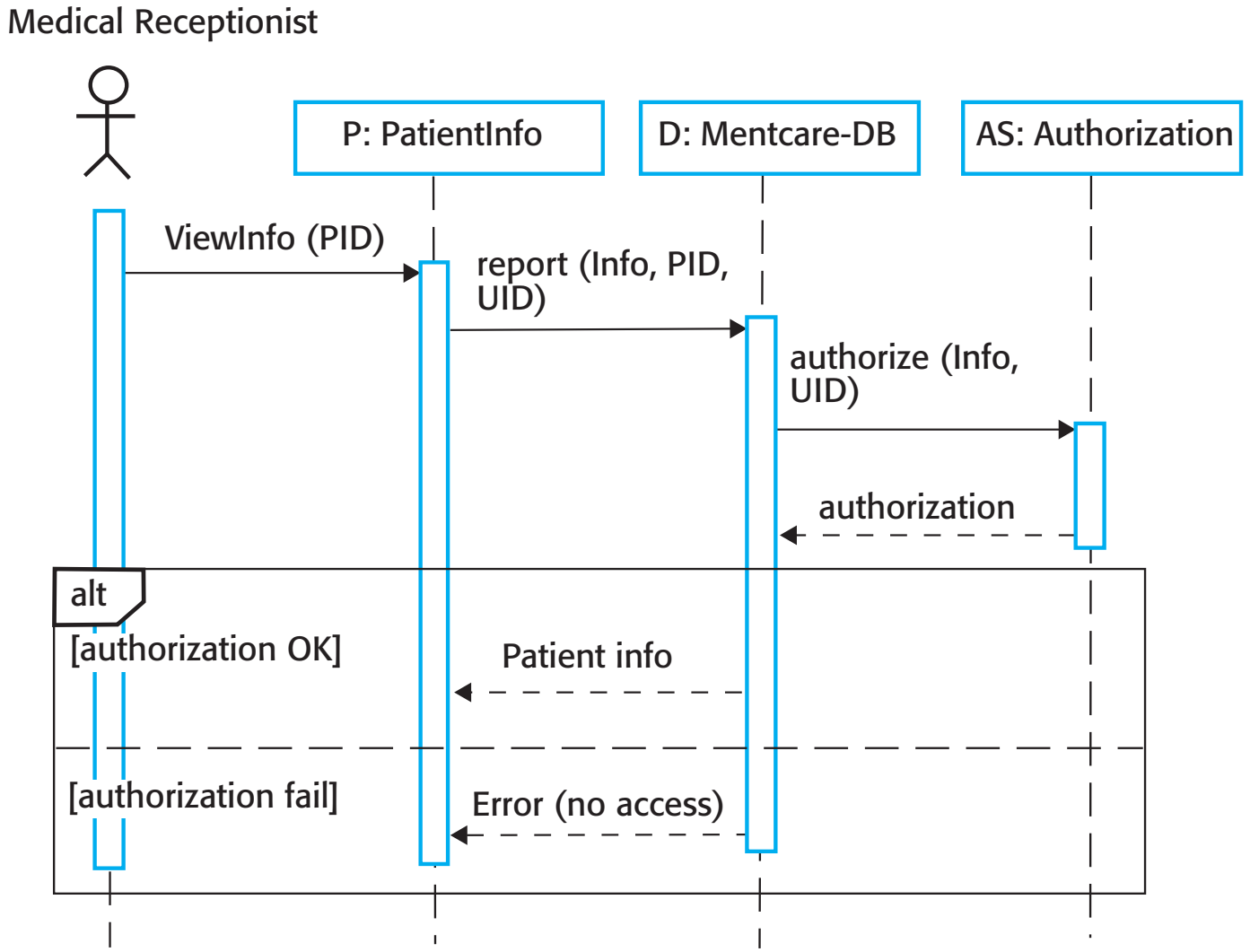
Interaction Models

- … show how users work with the system and how systems work together.
  - Sequence diagrams

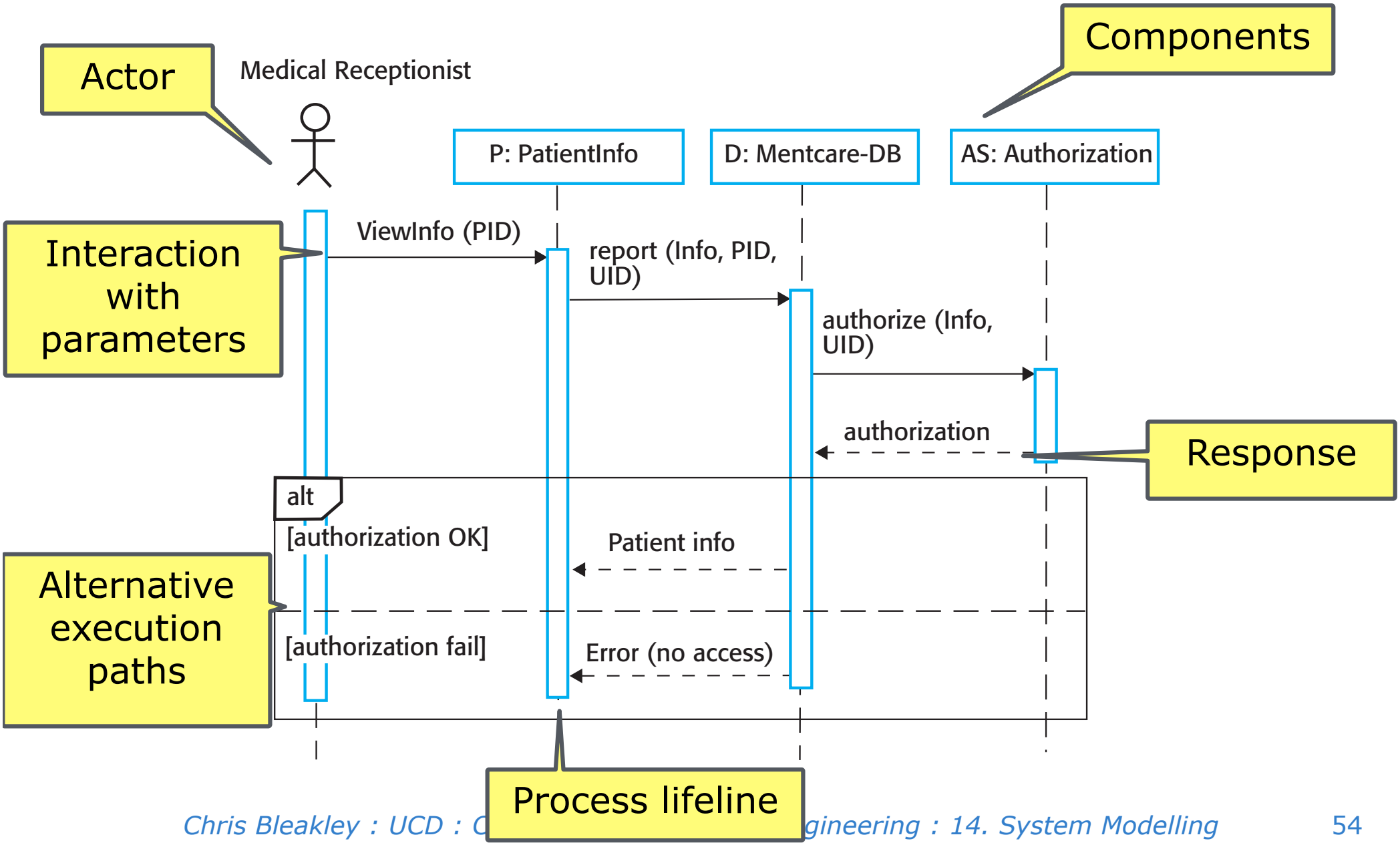
Sequence Diagrams

- … model the sequence of interactions over time between actors, components or objects in the system.
  - High level: interactions between systems and actors.
  - Medium level: interactions between components.
  - Low level: interactions between objects.
- A sequence diagram shows the sequence of interactions that take place during a particular use case or use case instance.

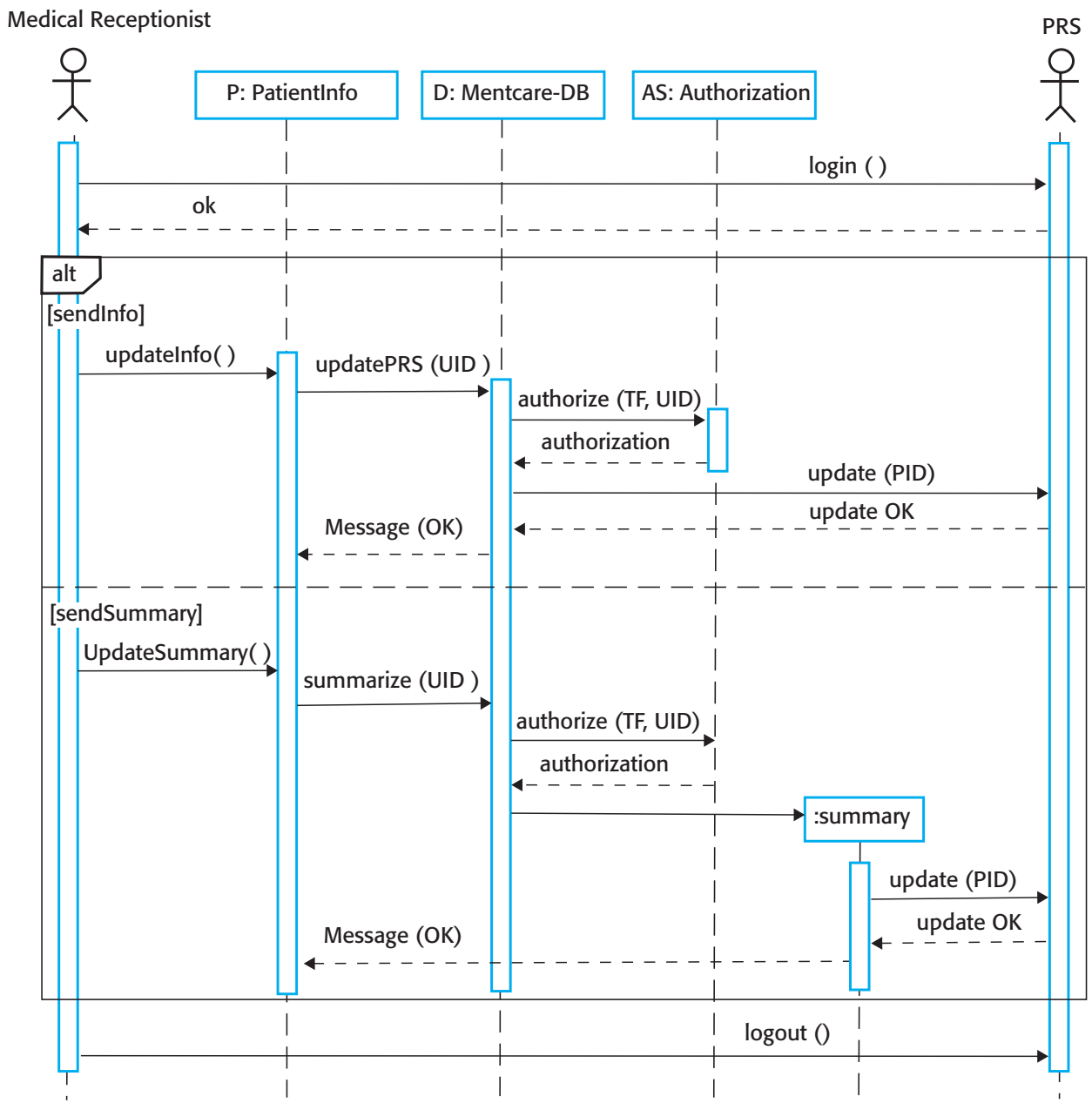
Mentcare Example: Sequence diagram for view patient information



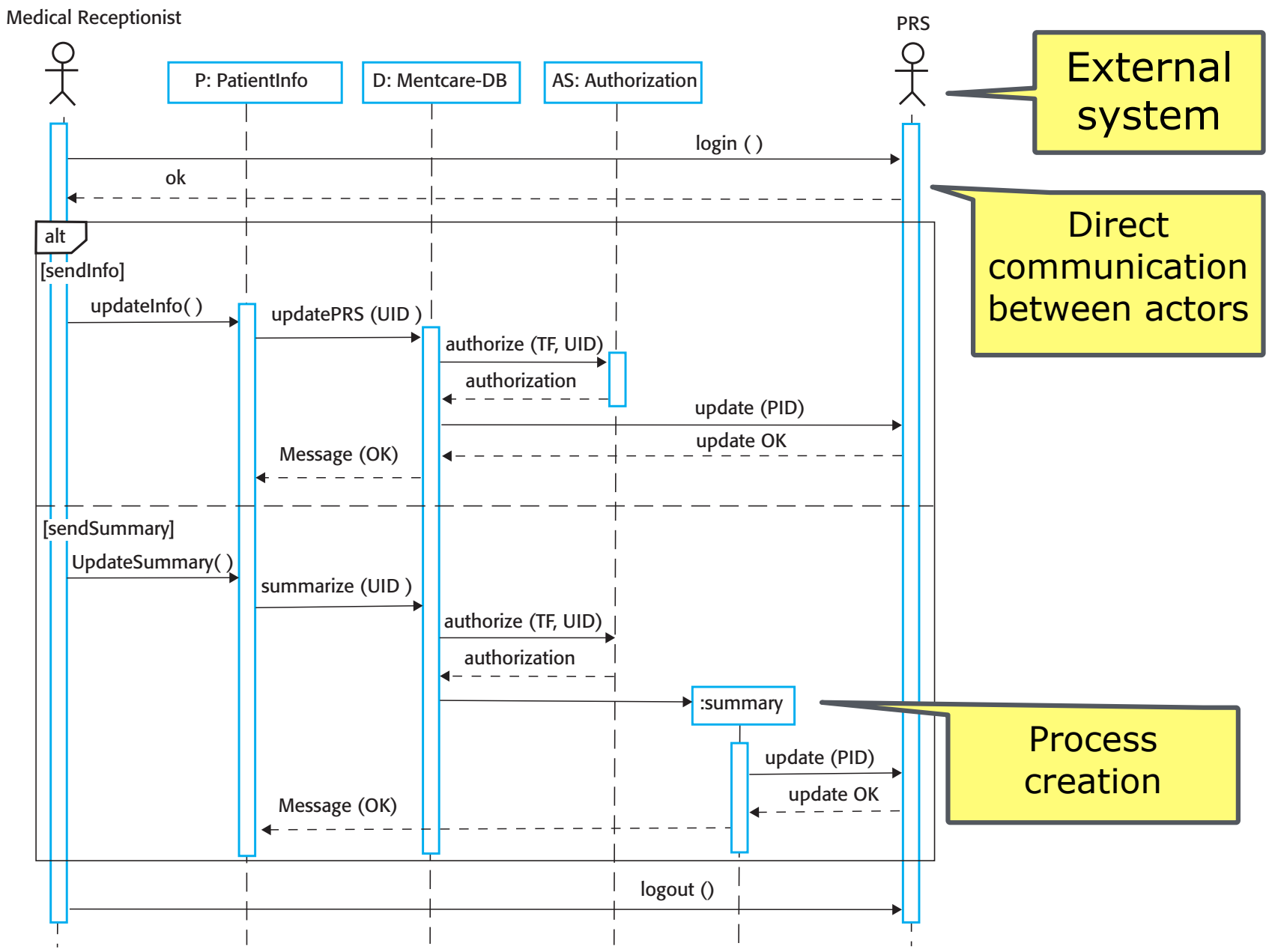
Mentcare Example: Sequence diagram for view patient information



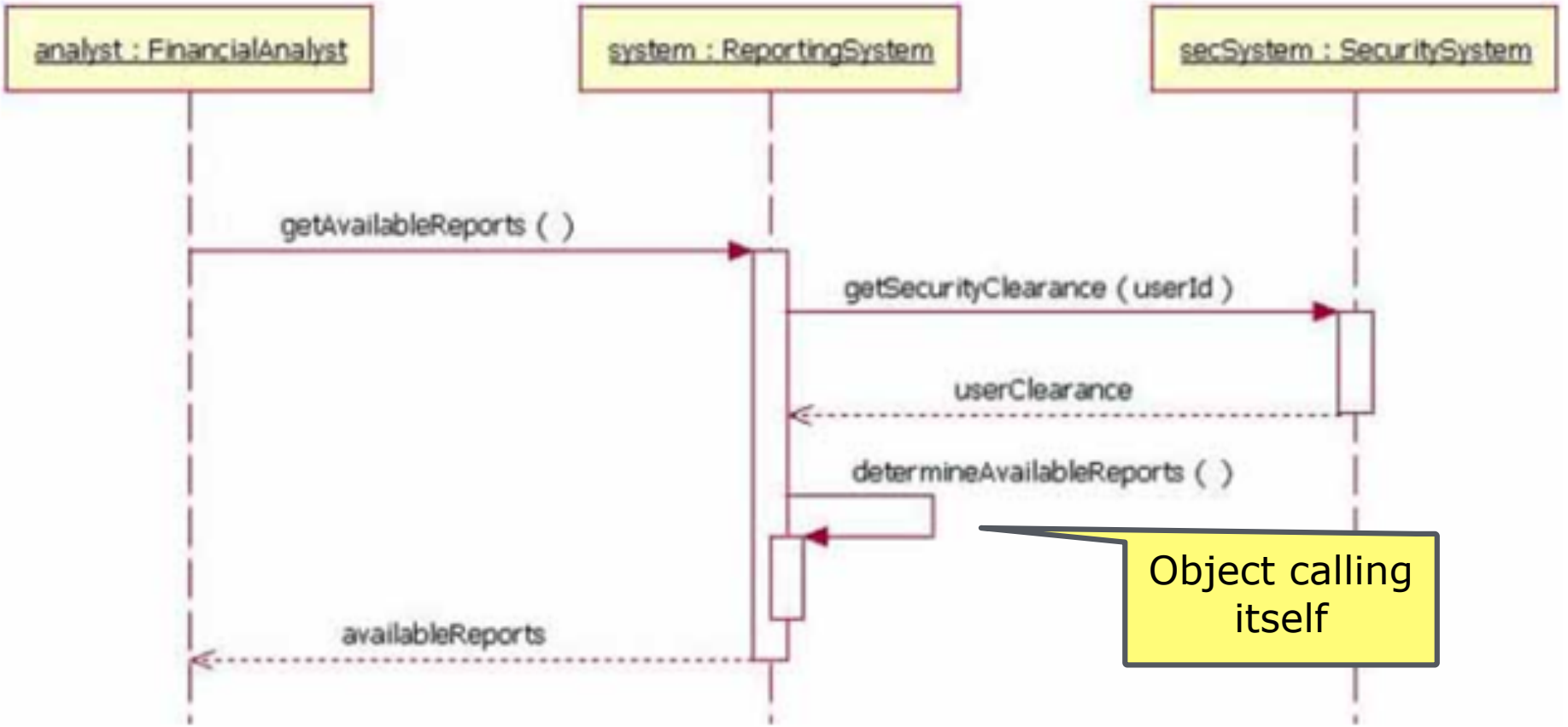
Mentcare Example:Sequence diagram for transfer data



Mentcare Example:Sequence diagram for transfer data

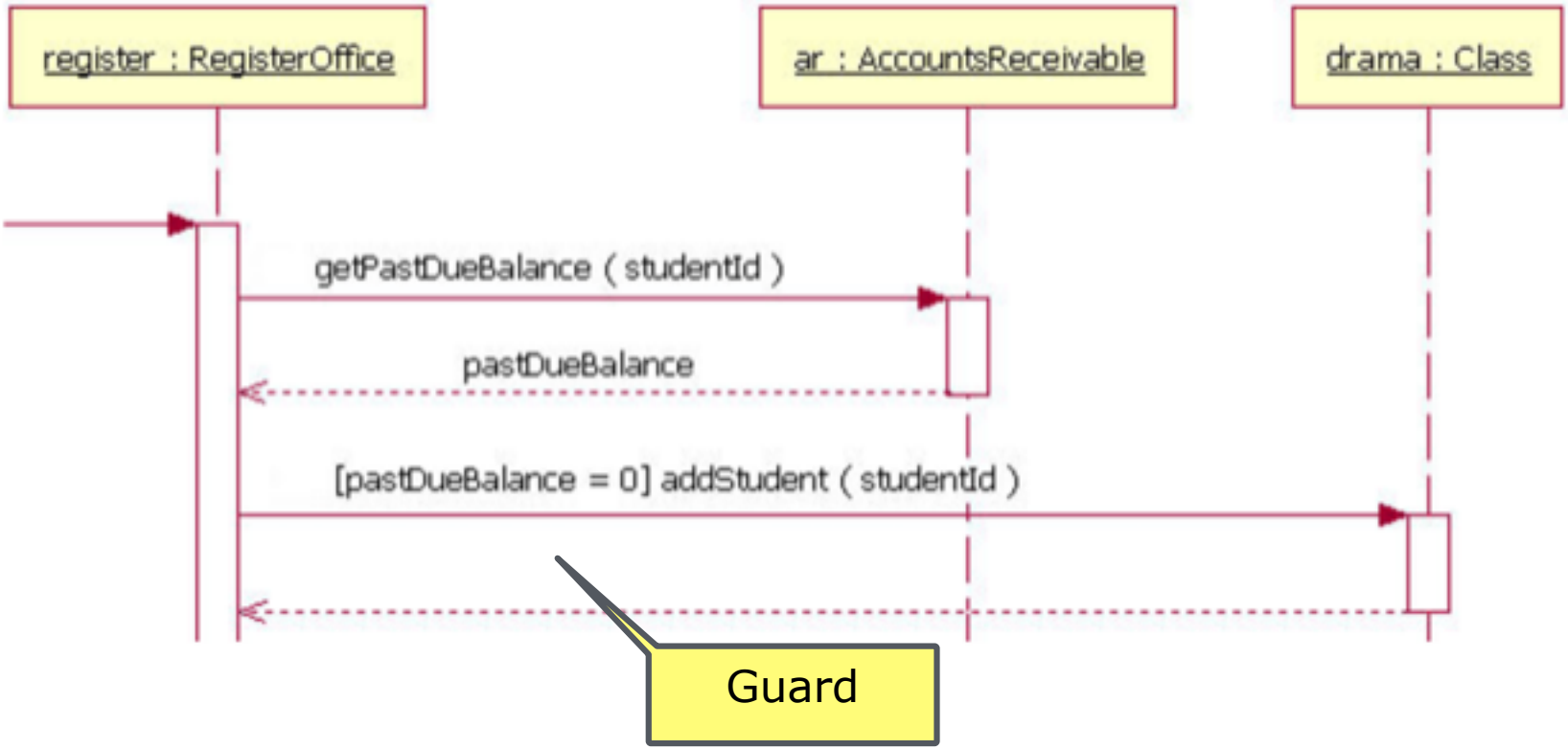


Object Calling Its Own Method



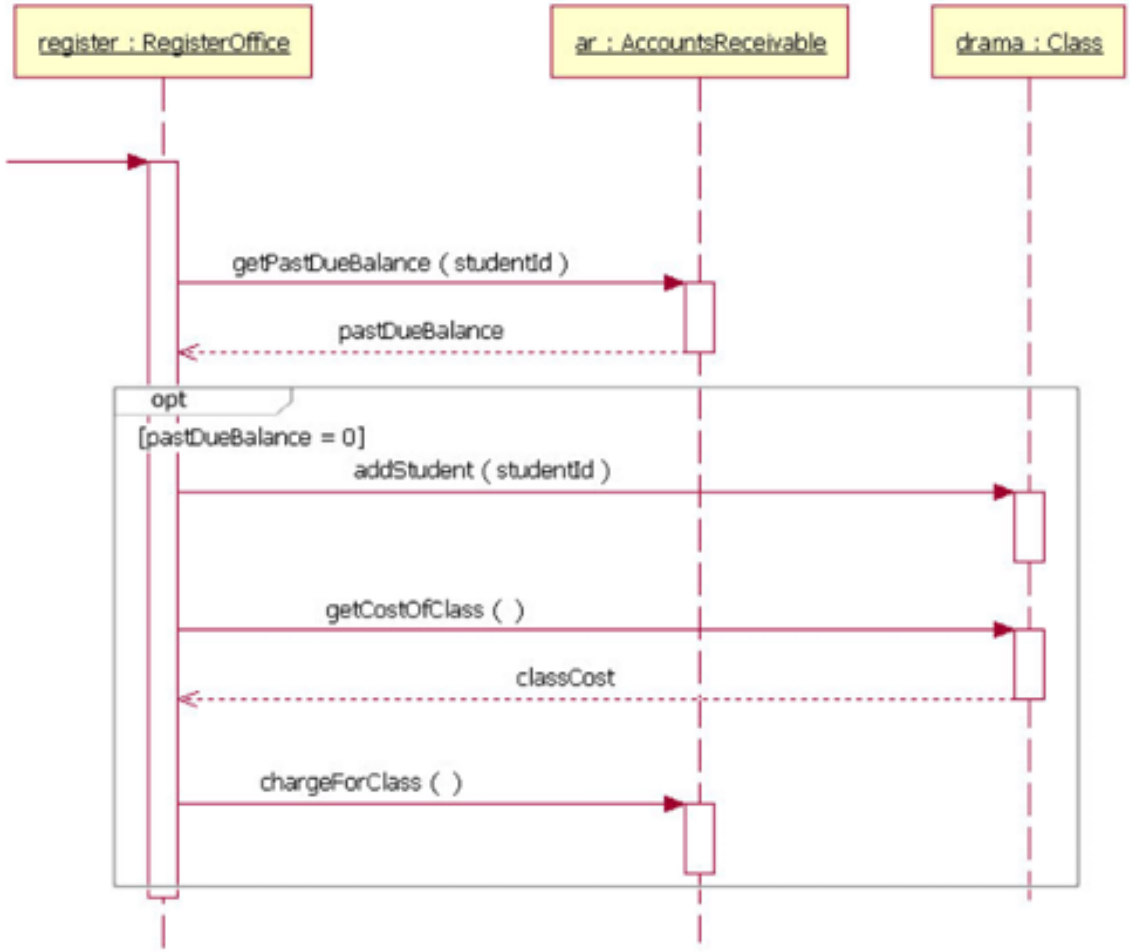
https://www.ibm.com/developerworks/rational/library/3101.html

Precondition for an Interaction



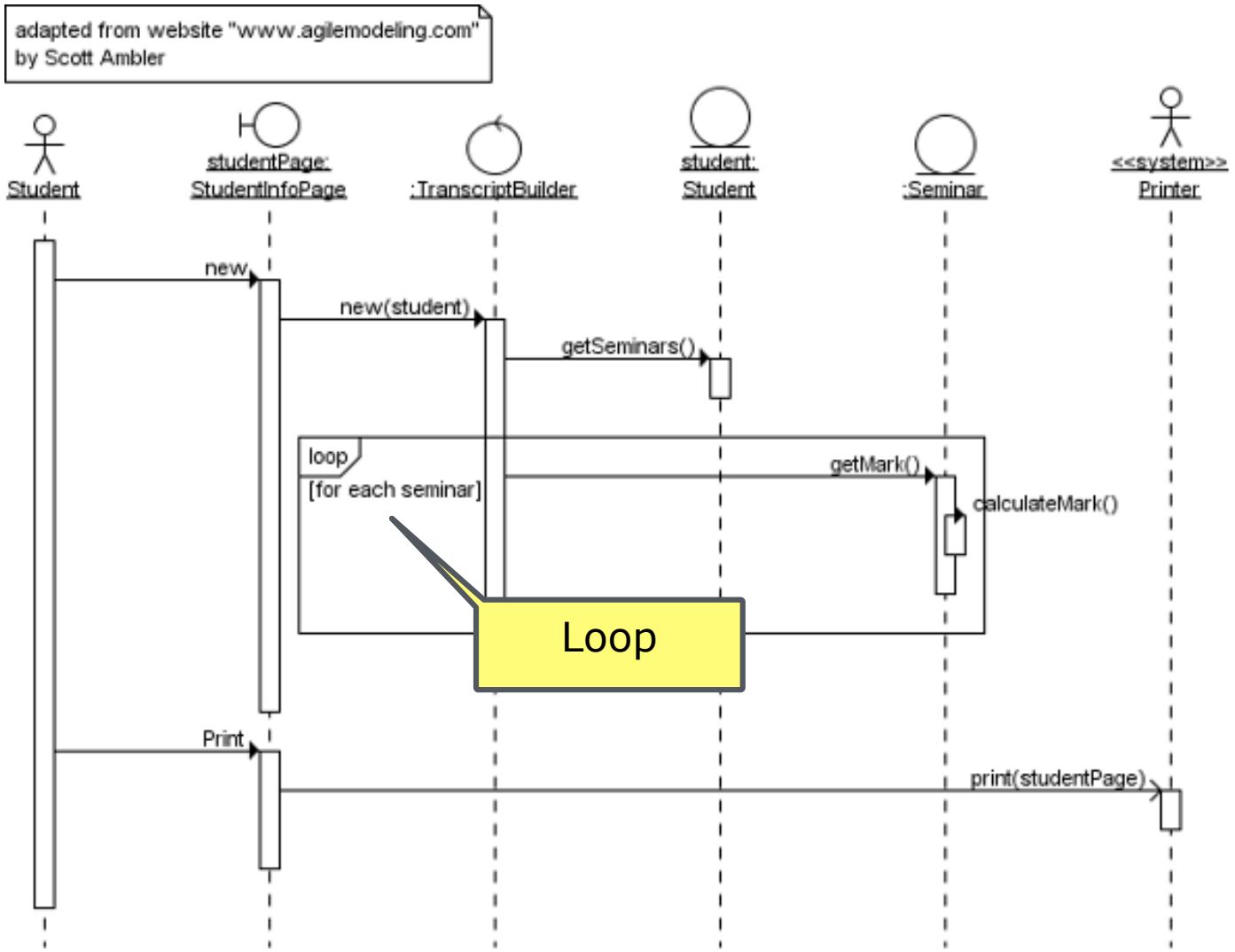
https://www.ibm.com/developerworks/rational/library/3101.html

Preconditions for Multiple Interactions

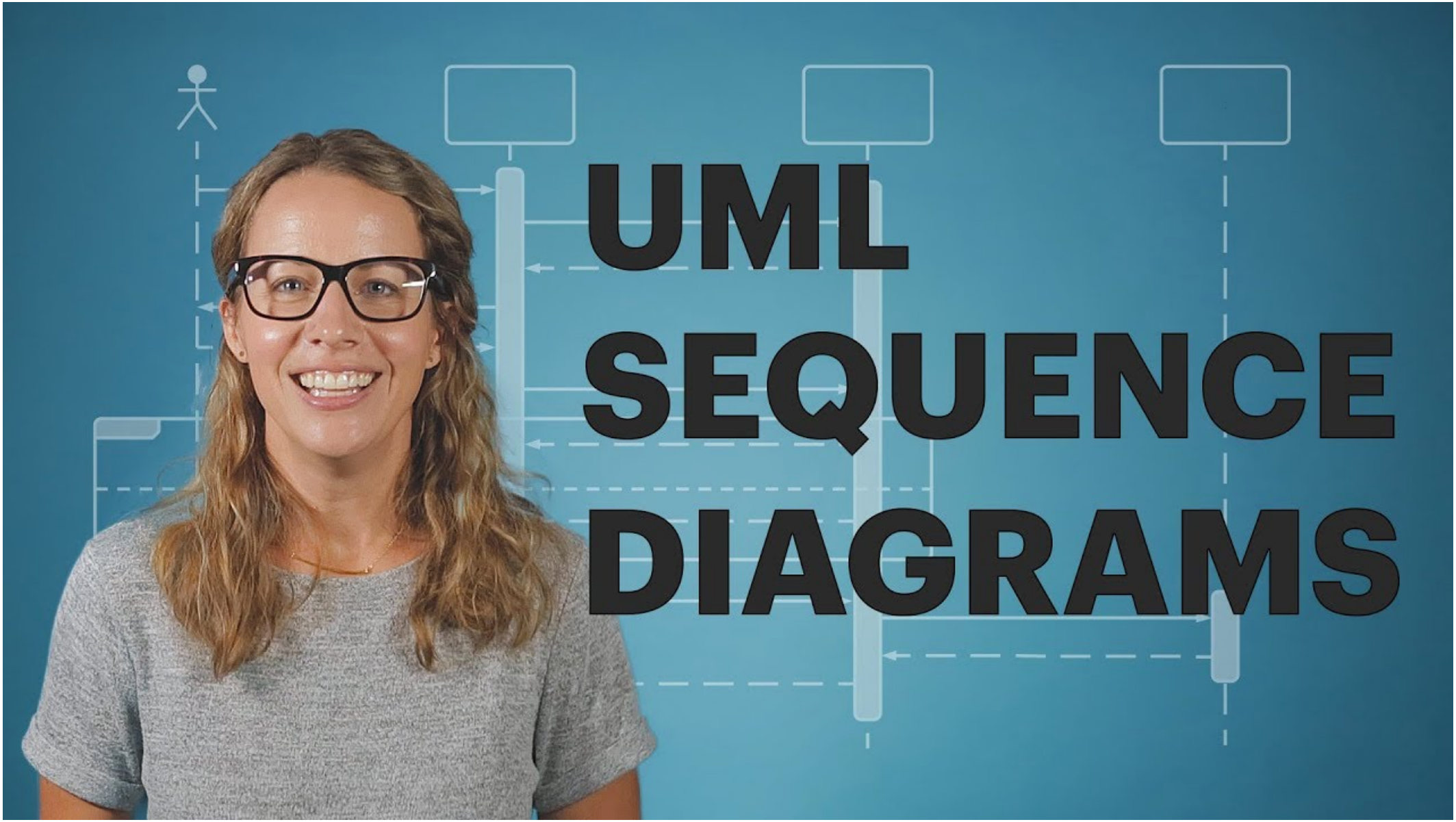


https://www.ibm.com/developerworks/rational/library/3101.html

Loops


adapted from website "www.agilemodeling.com"
by Scott Ambler
Student
studentPage:
StudentInfoPage
:TranscriptBuilder
student:
Student
:Seminar
<<system>>
Printer
new
new(student)
getSeminars()
loop
[for each seminar]
getMark()
calculateMark()
Loop
Print
print(studentPage)

How to Make a UML Sequence Diagram, Lucid Software, YouTube



Model-Driven Engineering

Model-Driven Engineering

- The concept is that designer enters the model and the tools automatically generate the code.
- Useful in constrained domains where there are libraries of commonly used components.
- In more general domains, the model-driven approach is sometimes used are the high level (architectural) only. Normal coding is used to fill in the details.

Model Based Design with Simulink, MATLAB, YouTube

Pop Quiz

Pop Quiz

- Create the following for the Individual Project Solution.
  - a Class Diagram
  - a Sequence Diagram
  - A State Diagram
- http://www.umletino.com

Summary

- **Context Models** illustrate the relationship of a system to other systems.
- Unified Modelling Language
- **Behavioural Models** show the dynamic behaviour of a system as it is executing.
  - Use case diagram
  - Activity diagram
  - State diagram
- **Structural Models** show the organisation of a system in terms of the components that make up the system and their relationships.
  - Class diagram
- **Interaction Models** show how users work with the system and how systems work together.
- Model-Driven Engineering