

**Today:**

**Getting from Requirements Document to Use Cases**

**Representing Use Cases with UML**

**Introduction to Crystal Clear**

**Introduction to GIT**

**Project – Class Exercise – creating use cases**

**Next Steps -**

**Setting up your project on GIT**

**<http://msysgit.github.io/>**

Many slides in this weeks presentation provided by Ian Sommerville,  
Software Engineering, 9<sup>th</sup> Edition

# Chapter 5 – System Modeling

## Lecture 1

---

## Three Amigos -

**UML** was created at Rational ~1994-1996, by Grady Booch, Ivar Jacobson, and James Rumbaugh, adopted by the OMG in 1997 (Think BJR – Better Java Required)

## **Some other names to know**

---

Kent Beck - XP

Ward Cunningham – Wiki, design patterns, XP

Martin Fowler - Thoughtworks

Jim Highsmith – Cutter Consortium

Ron Jeffries - XP

Robert C. Martin – XP

Ed Yourdon – Cutter Consortium

Adrian Cockburn – Crystal

# UML Diagrams -

---

## Structural Diagrams

- Class
- Component
- Composite structure
- Deployment
- Object
- Package
- Profile

# UML Diagrams

---

Behavioral diagrams include

- Activity
- State
- Use Case
- Plus Four kinds of interaction diagrams:
  - Sequence
  - Communication
  - Interaction overview
  - Timing

## UML diagram types

---

- ◆ Activity diagrams, which show the activities involved in a process or in data processing .
- ◆ 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.
- ◆ State diagrams, which show how the system reacts to internal and external events.

## 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.



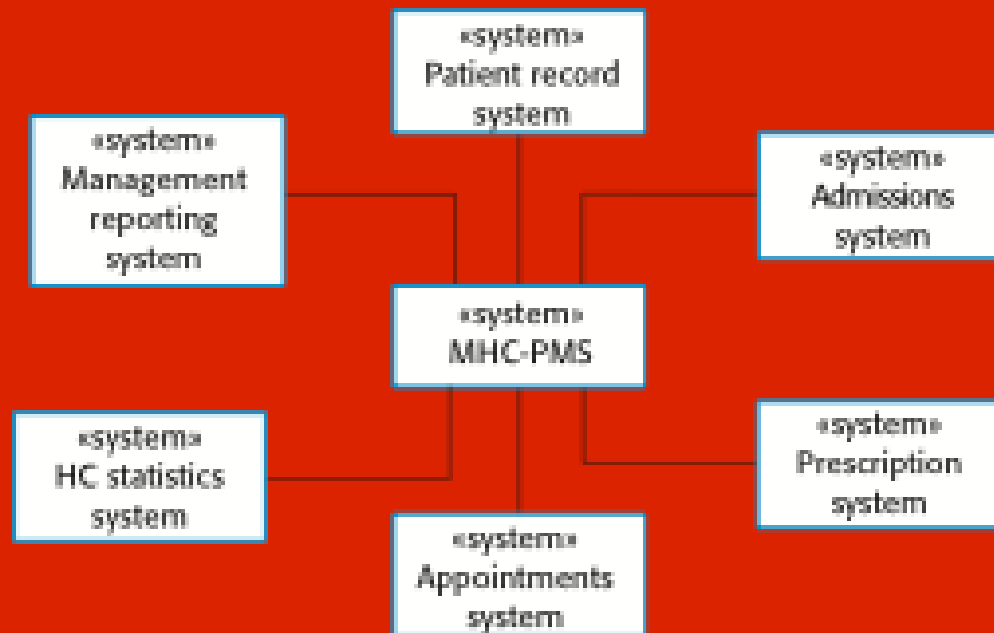
# System boundaries

---

- ◇ 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 MHC-PMS

---

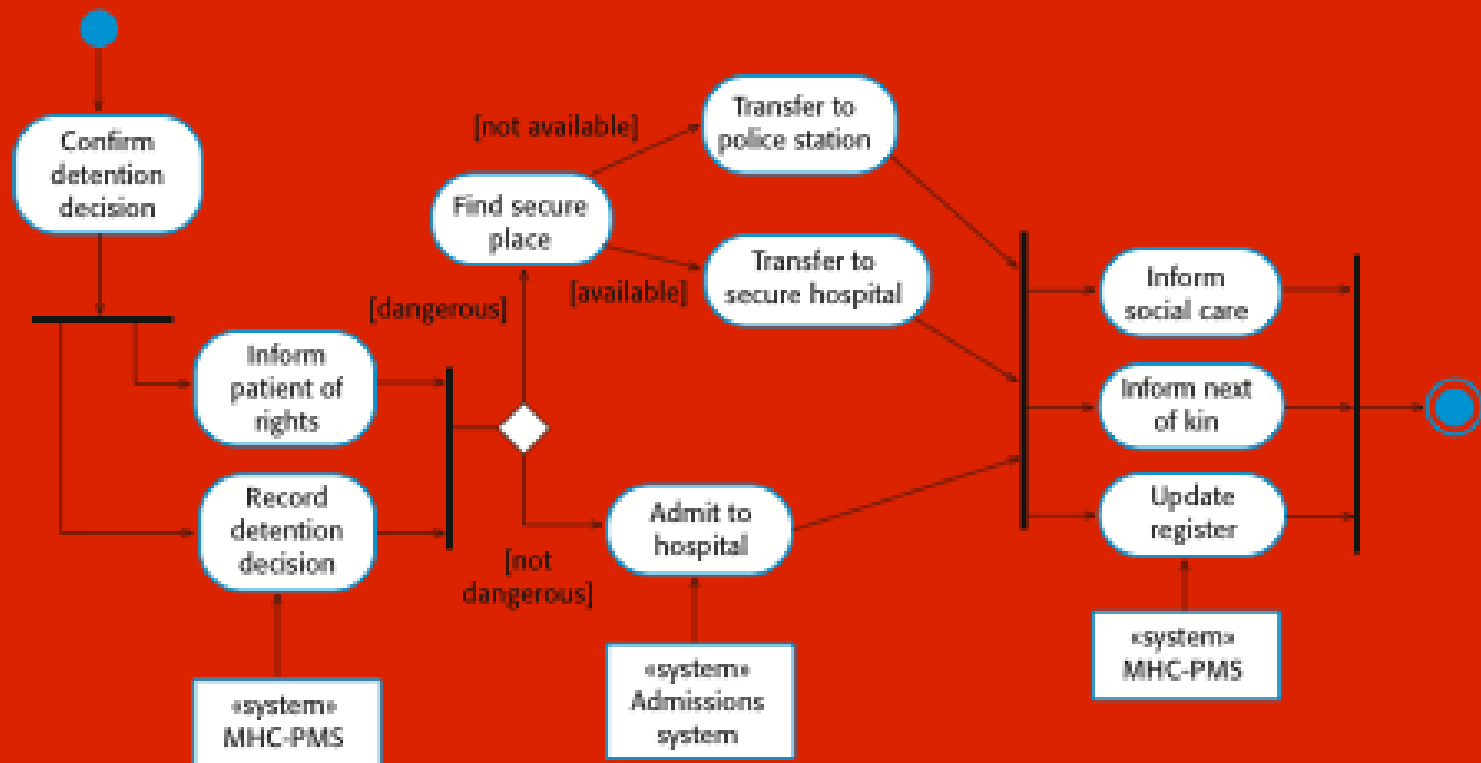


## Process perspective

---

- ◆ 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



## 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.
- ◆ Use case diagrams and sequence diagrams may be used for interaction modeling.

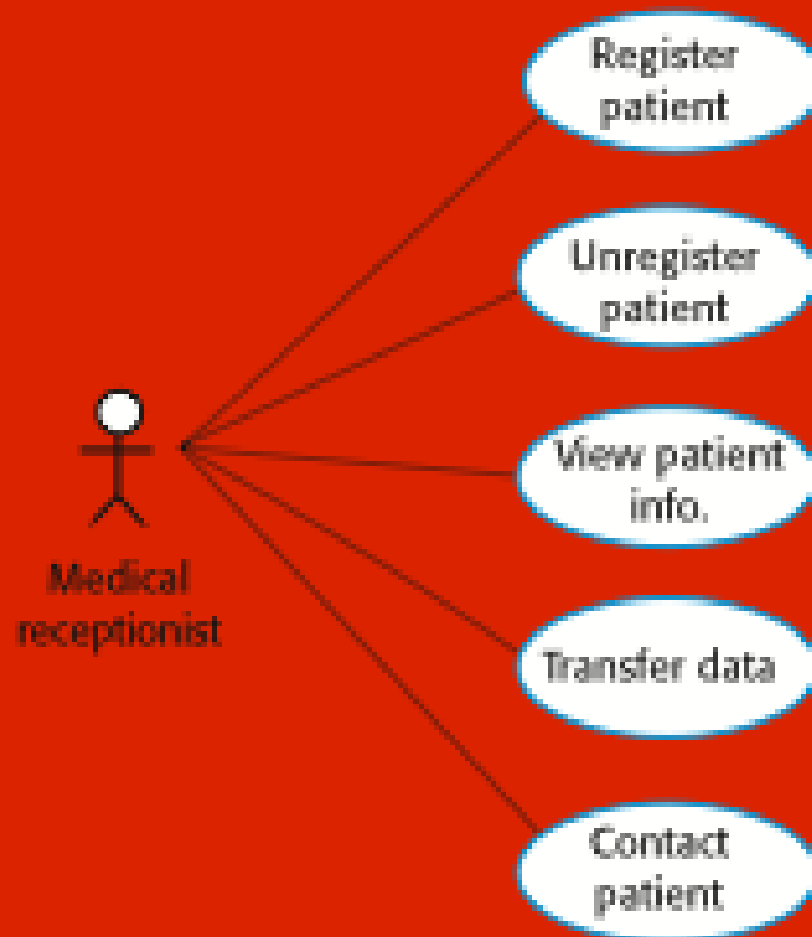
## Use case modeling

---

- ◆ 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.

## Use cases in the MHC-PMS involving the role 'Medical Receptionist'

---



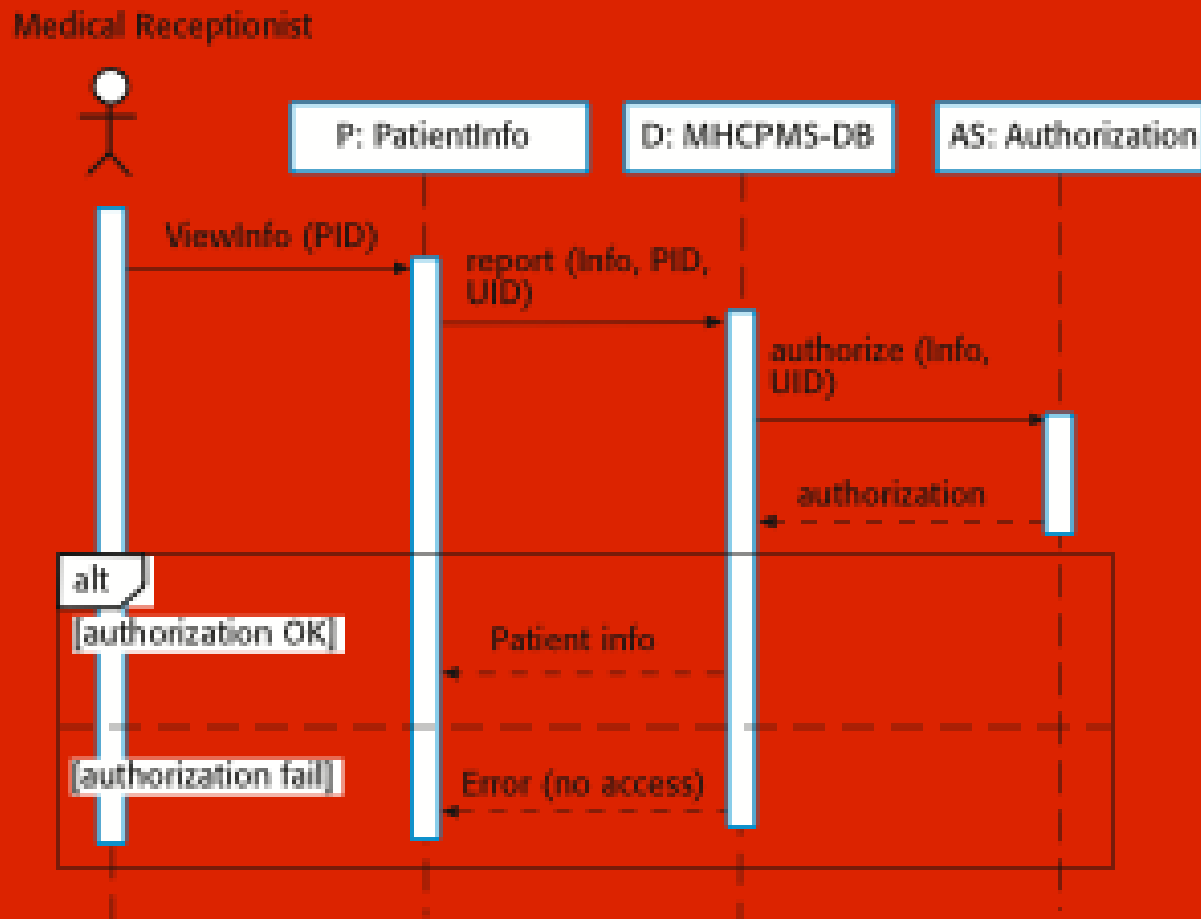
# Sequence diagrams

---

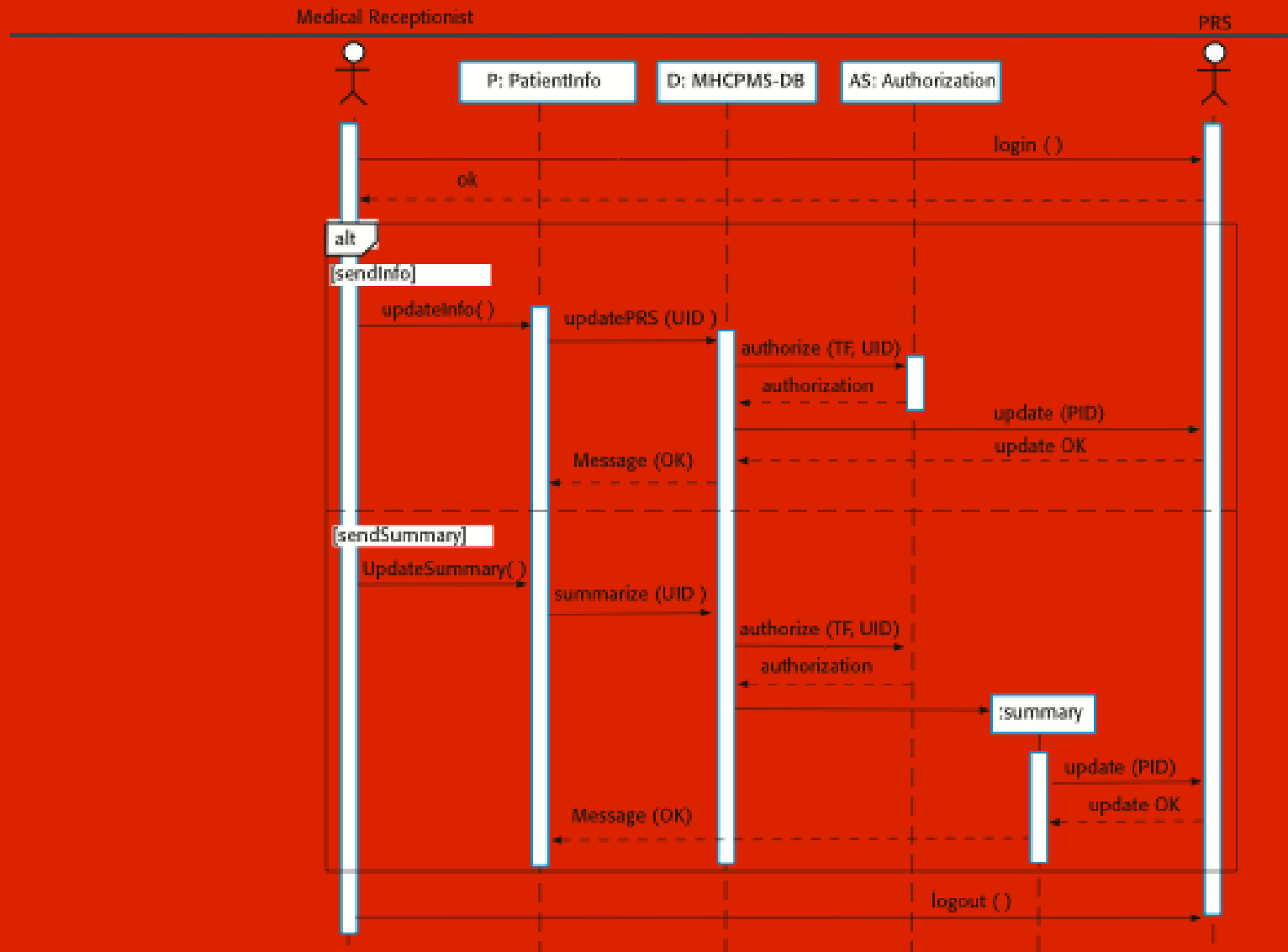
- ◆ Sequence diagrams are part of the UML and are used to model the interactions between the actors and the objects within a system.
- ◆ A sequence diagram shows the sequence of interactions that take place during a particular use case or use case instance.
- ◆ The objects and actors involved are listed 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



## Structural models

---

- ◇ Structural models of software display the organization of a system in terms of the components that make up that system and their relationships.
- ◇ Structural models may be static models, which show the structure of the system design, or dynamic models, which show the organization of the system when it is executing.
- ◇ You create structural models of a system when you are discussing and designing the system architecture.

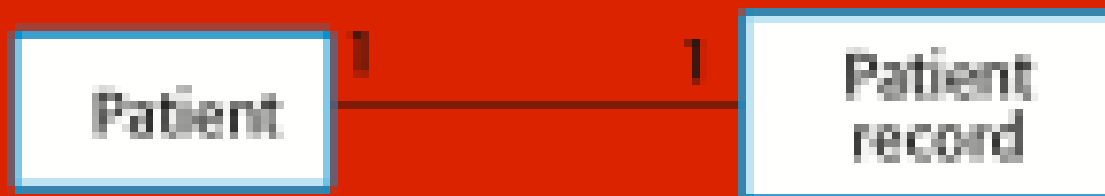
# Class diagrams

---

- ❖ Class diagrams are 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 that indicates that there is some relationship between these classes.
- ❖ When you are developing models during the early stages of the software engineering process, objects represent something in the real world, such as a patient, a prescription, doctor, etc.

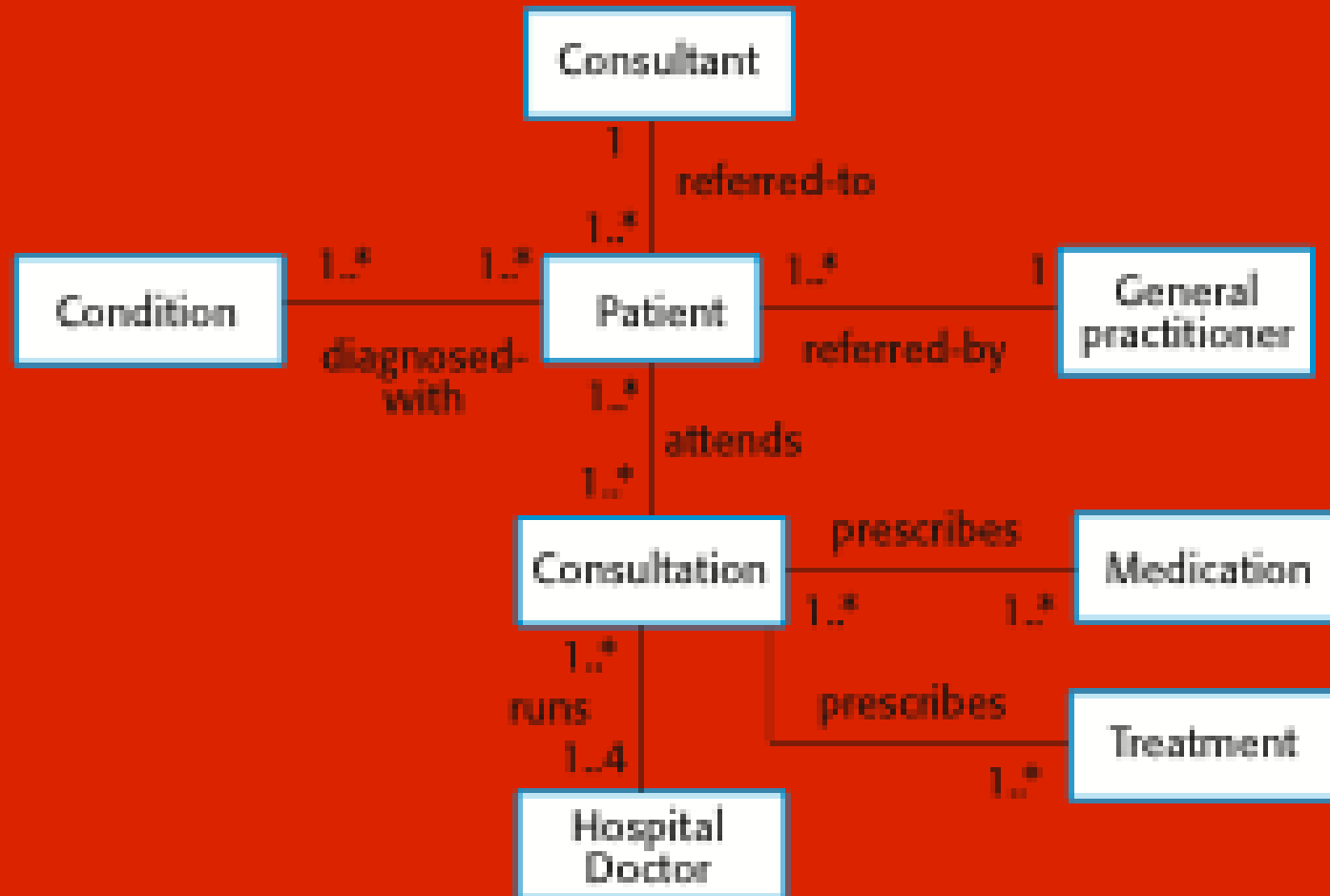
# UML classes and association

---



# Classes and associations in the MHC-PMS

---



## The Consultation class

---

### Consultation

Doctors  
Date  
Time  
Clinic  
Reason  
Medication prescribed  
Treatment prescribed  
Voice notes  
Transcript  
...

New ()  
Prescribe ()  
RecordNotes ()  
Transcribe ()  
...

## Key points

---

- ◆ A model is an abstract view of a system that ignores system details. Complementary system models can be developed to show the system's context, interactions, structure and behavior.
- ◆ Context models show how a system that is being modeled is positioned in an environment with other systems and processes.
- ◆ Use case diagrams and sequence diagrams are used to describe the interactions between users and systems in the system being designed. Use cases describe interactions between a system and external actors; sequence diagrams add more information to these by showing interactions between system objects.
- ◆ Structural models show the organization and architecture of a system. Class diagrams are used to define the static structure of classes in a system and their associations.



---

## Chapter 5 – System Modeling

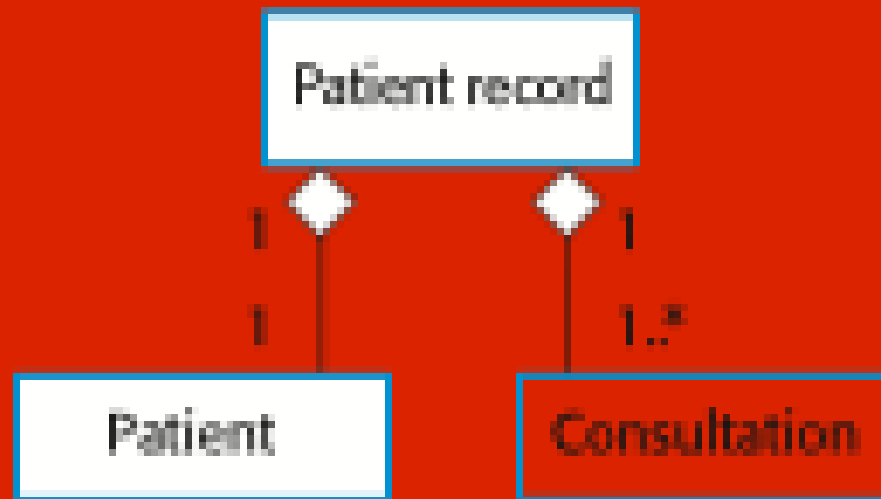
## Object class aggregation models

---

- ◆ An aggregation model shows how classes that are collections are composed of other classes.
- ◆ Aggregation models are similar to the part-of relationship in semantic data models.

## The aggregation association

---



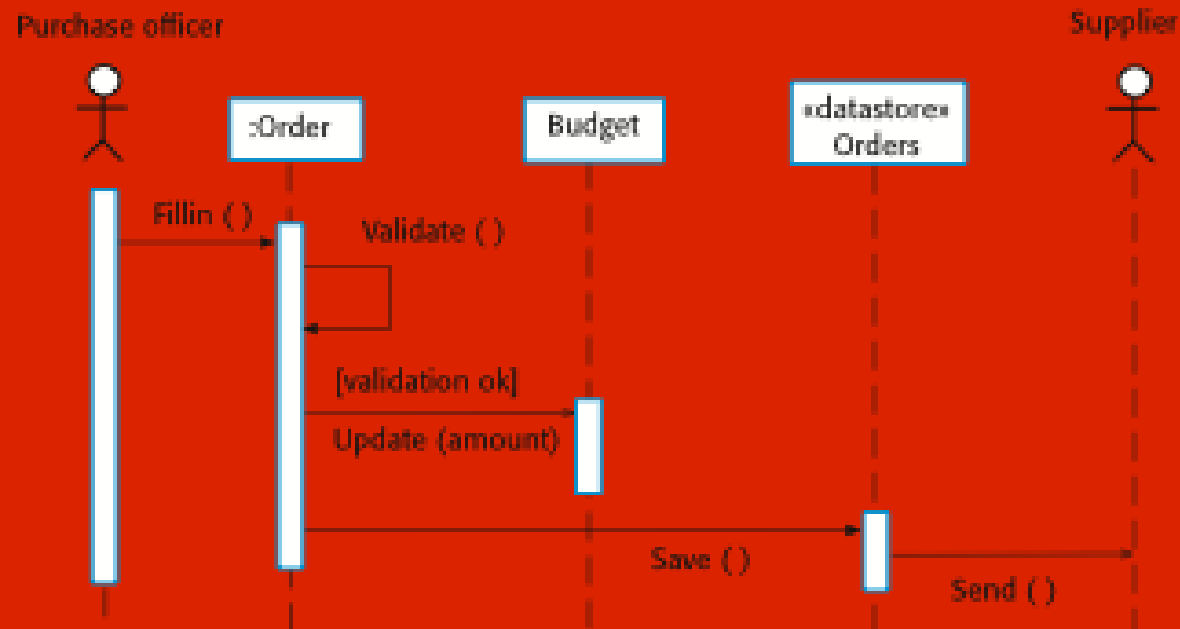
## Behavioral models

---

- ◇ Behavioral models are 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.

# Order processing

---



## Event-driven modeling

---

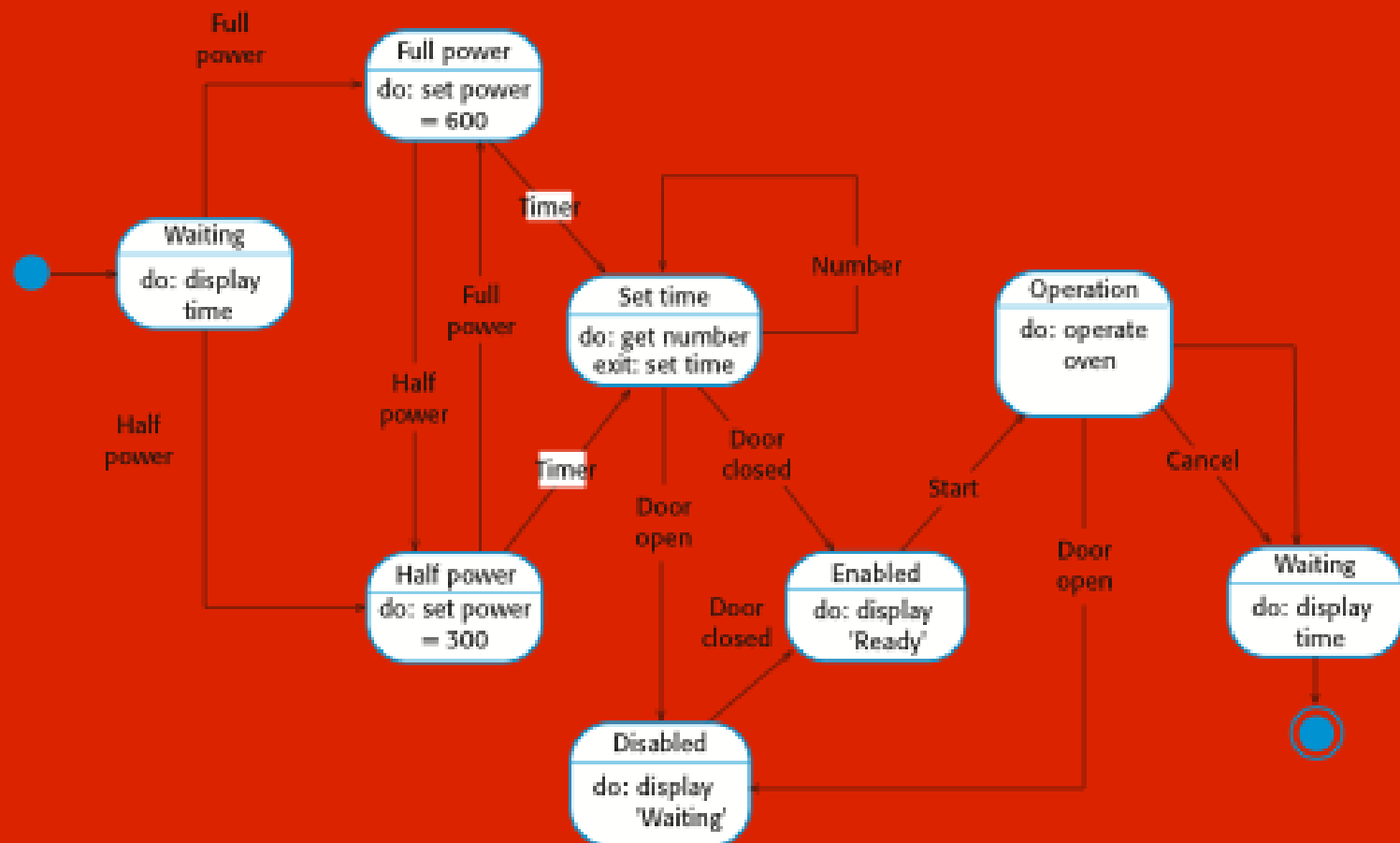
- ◆ Real-time systems are often event-driven, with minimal data processing. For example, a landline phone switching system responds to events such as 'receiver off hook' by generating a dial tone.
- ◆ Event-driven modeling shows how a system responds to external and internal events.
- ◆ It is based on the assumption that a system has a finite number of states and that 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.
- ◆ Statecharts are an integral part of the UML and are used to represent state machine models.

# State diagram of a microwave oven





## States and stimuli for the microwave oven (a)

---

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.

# GIT

---

- ◇
- ◇ Create a github account
- ◇ Install and configure GIT on your local machine
- ◇ Appoint a configuration manager for your group
- ◇ Create your project repository
- ◇ Add collaborators (the accounts for everyone in your group)
- ◇ **<http://msysgit.github.io/>**
- ◇ **<https://help.github.com/articles/git-cheatsheet>**
- ◇