Software Engineering

Lecture 9

Mr. Noor Ul Arfeen

System Modeling

System modeling is the process of developing abstract models of a system, with each model presenting a different view or perspective of that system

System modelling helps the analyst to understand the functionality of the system and models are used to communicate with customers

System modeling has now come to mean representing a system using some kind of graphical notation, which is now almost always based on notations in the Unified Modeling Language (UML)

System Perspectives

An **external perspective**, where you model the context or environment of the system

An **interaction perspective**, where you model the interactions between a system and its environment, or between the components of a system

A **structural perspective**, where you model the organization of a system or the structure of the data that is processed by the system

A **behavioral perspective**, where you model the dynamic behavior of the system and how it responds to events

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

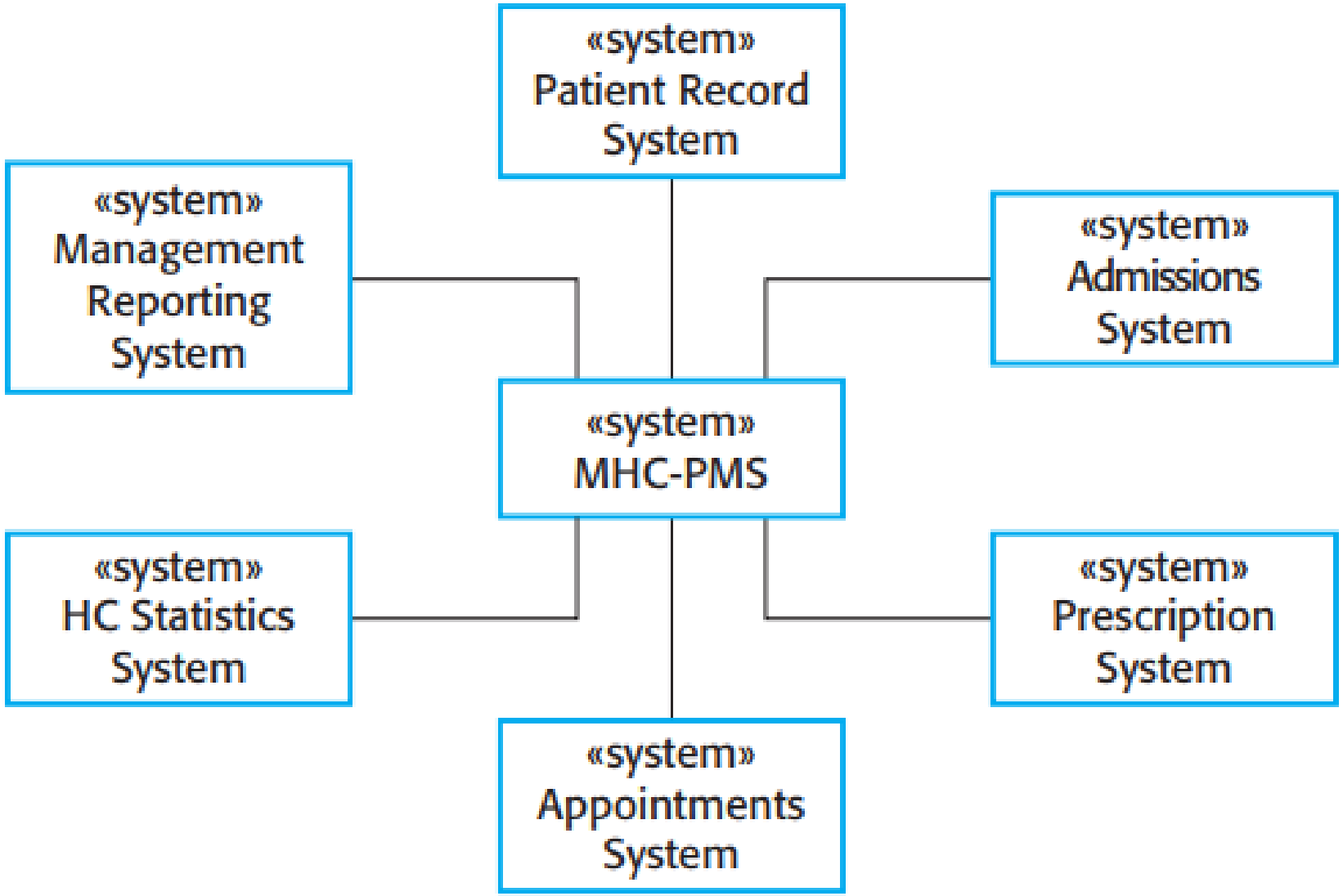
Architectural models show the system and its relationship with other systems

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

The context of the MHC-PMS



Interaction Models

Types of interactions that can be represented in a model:

Modeling **user interaction** is important as it helps to identify user requirements

Modeling **system-to-system interaction** highlights 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 case modeling, is used to model interactions between a system and external actors (users or other systems)

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

Use cases can be represented using a *UML use case diagram* and in a more detailed *textual/tabular format*

Transfer-data use case

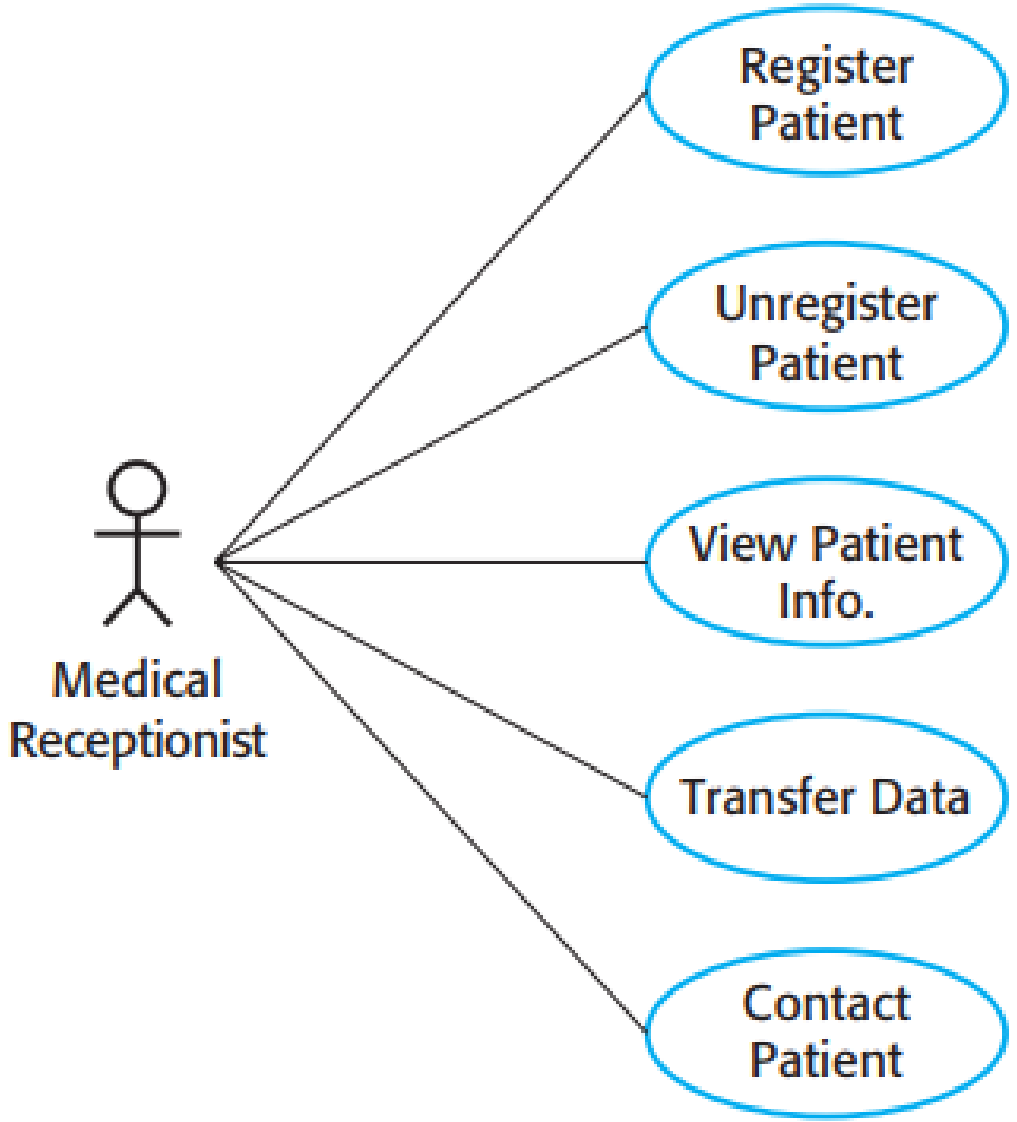
A use case from the MHC-PMS that represents the task of uploading data from the MHC-PMS to a more general patient record system

There are two actors in this use case: the operator who is transferring the data and the patient record system

Tabular description of the 'Transfer data' use case

MHC-PMS: Transfer data	
Actors	Medical receptionist, patient records system (PRS)
Description	A receptionist may transfer data from the MHC-PMS 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.

Use cases involving the role ‘medical receptionist’



Sequence Diagrams

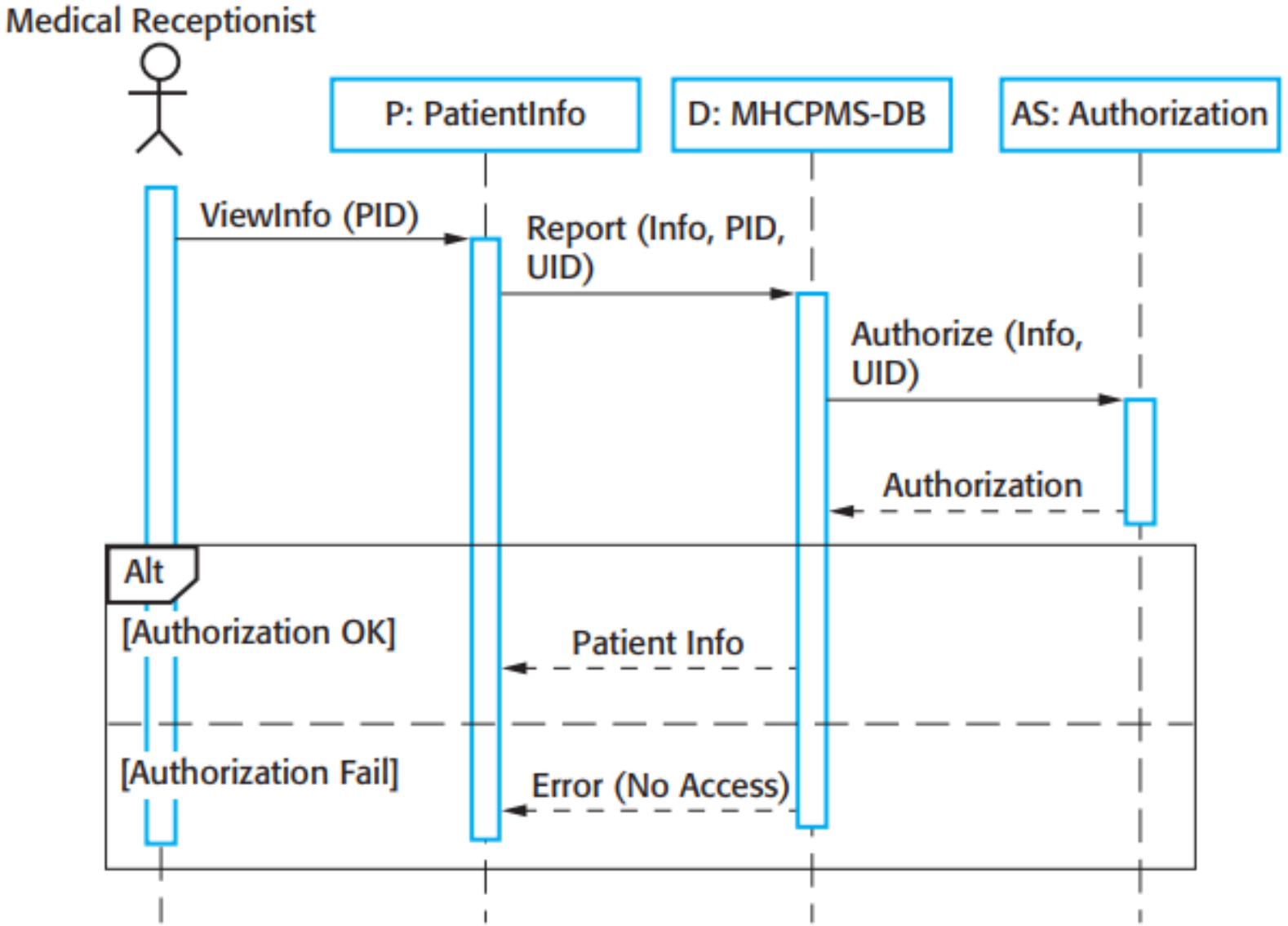
Sequence diagrams are part of the UML and **are used to model the interactions** between the actors and the objects within a system

A sequence diagram **shows the sequence of interactions** that take place during a particular use case or use case instance

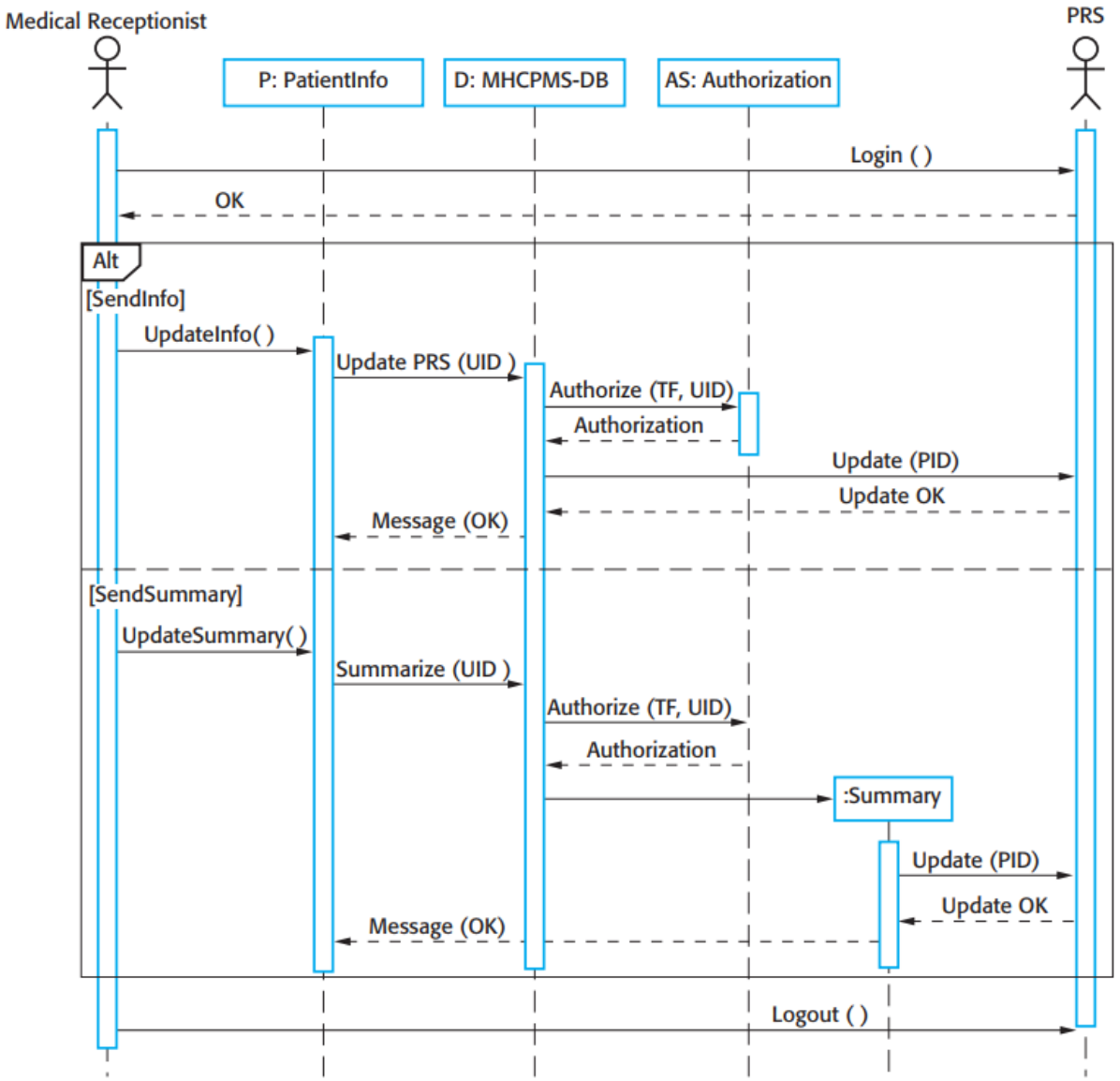
The **objects and actors involved are listed along the top of the diagram**, with a dotted line drawn vertically from these

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

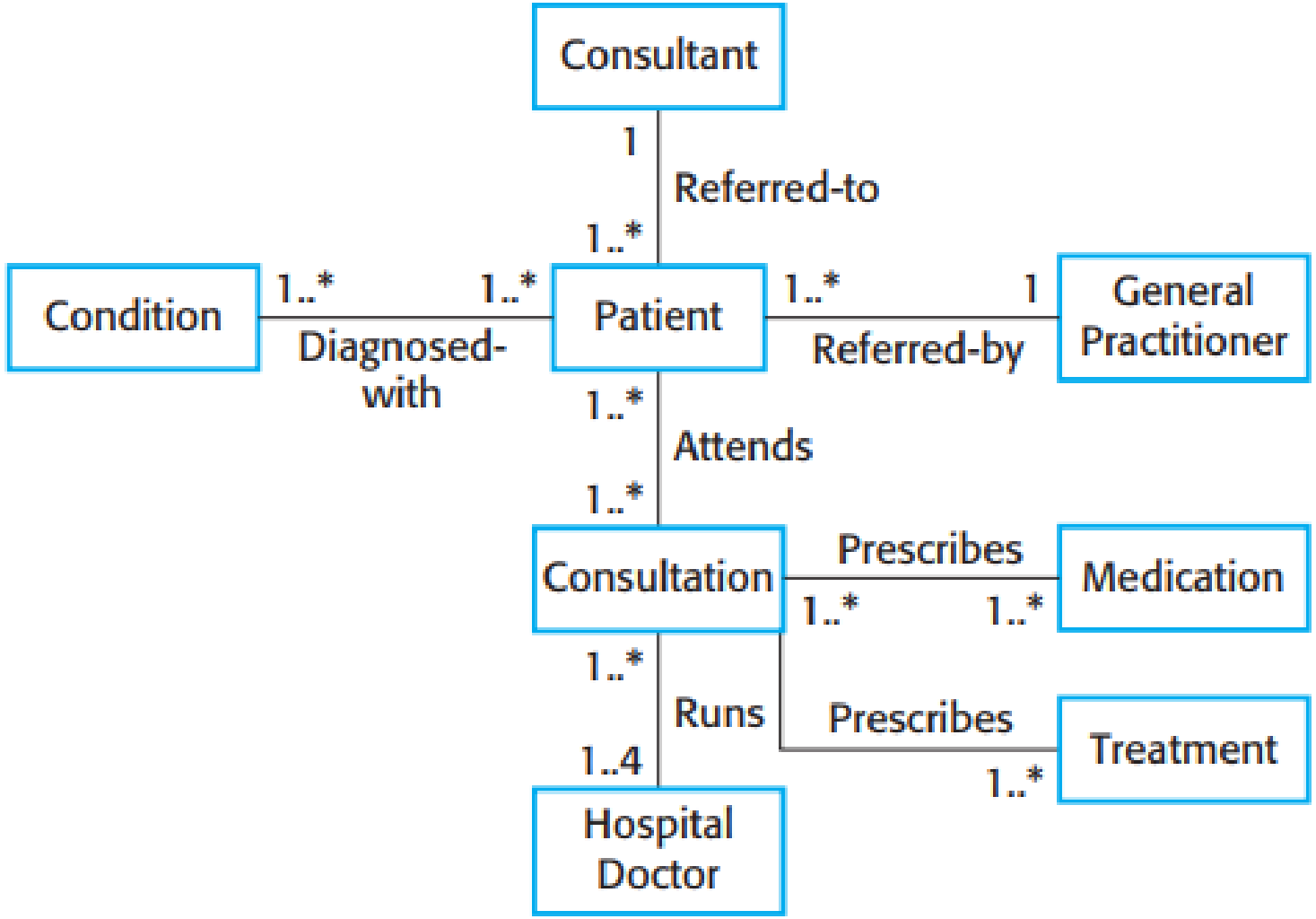
Class diagrams are used when developing an object-oriented system model to show the ***classes*** in a system and the ***associations*** between these classes

An 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

Class Diagrams

UML classes and association



Classes and associations in MHC-PMS

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

Behavioral Models

Data-driven models show the sequence of actions involved in processing input data and generating an associated output

Event-driven modeling shows how a system responds to external and internal events