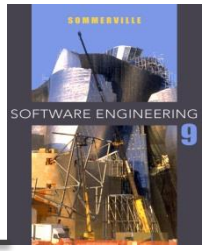
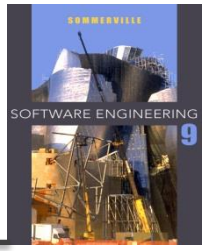


System Modeling

Introductory Slides: System modeling

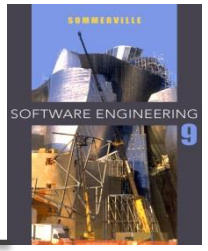


- System modeling is the process of developing abstract models of a system, with each model presenting a different **view** or perspective of that system.

- System 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 modeling helps the analyst
  - to **understand** the functionality of the system and
  - models are used to **communicate** with customers.

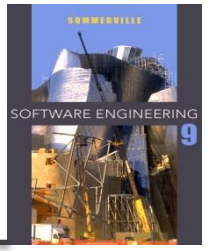
Existing and Planned System Models



- Models of the **existing system** are used during requirements engineering.
  - They help clarify what the existing system does and can be
  - used as a basis for discussing its strengths and weaknesses.
  - These then lead to requirements for the new system.
- Models of the **new system** are used during requirements engineering to help explain the **proposed requirements** to other system stakeholders.
  - Engineers use these models to discuss design proposals and
  - to document the system for implementation.

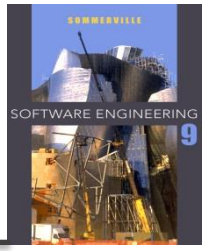
- In a model-driven engineering process, it is possible to generate a complete or partial system implementation from the **system model.**

System Perspectives



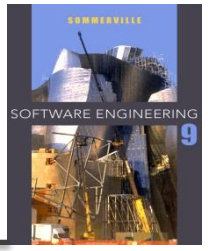
- An **external perspective**, where you model the context or environment of the system.
- An **interaction perspective**, where you model the interactions between a system and its environment, **or** between the components of a system.
- A **structural perspective**, where you model the organization 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



- The System Perspectives (last slide) are modeled with diagrams
- **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.

1. Context Models

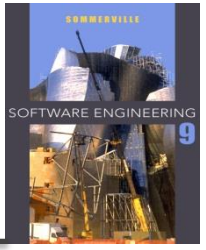


- Context models are used to illustrate the **operational context** of a system -
- They show what lies outside the system boundaries.

- Social and organizational concerns may affect the decision on where to position system boundaries.

- Architectural models show the system and its relationship with other systems.

System Boundaries

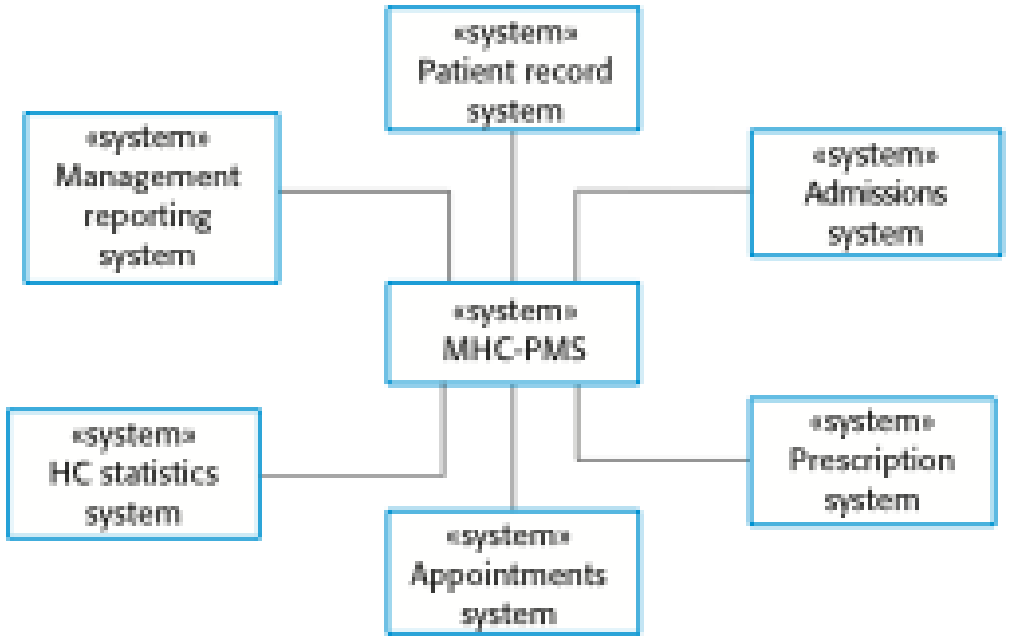


- System boundaries are established to define **what is inside and what is outside the system.**
  - They show other systems that are used or depend on the system being developed.

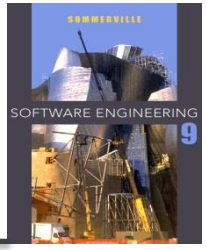
- The position of the system boundary has a profound effect on the system requirements.

- 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

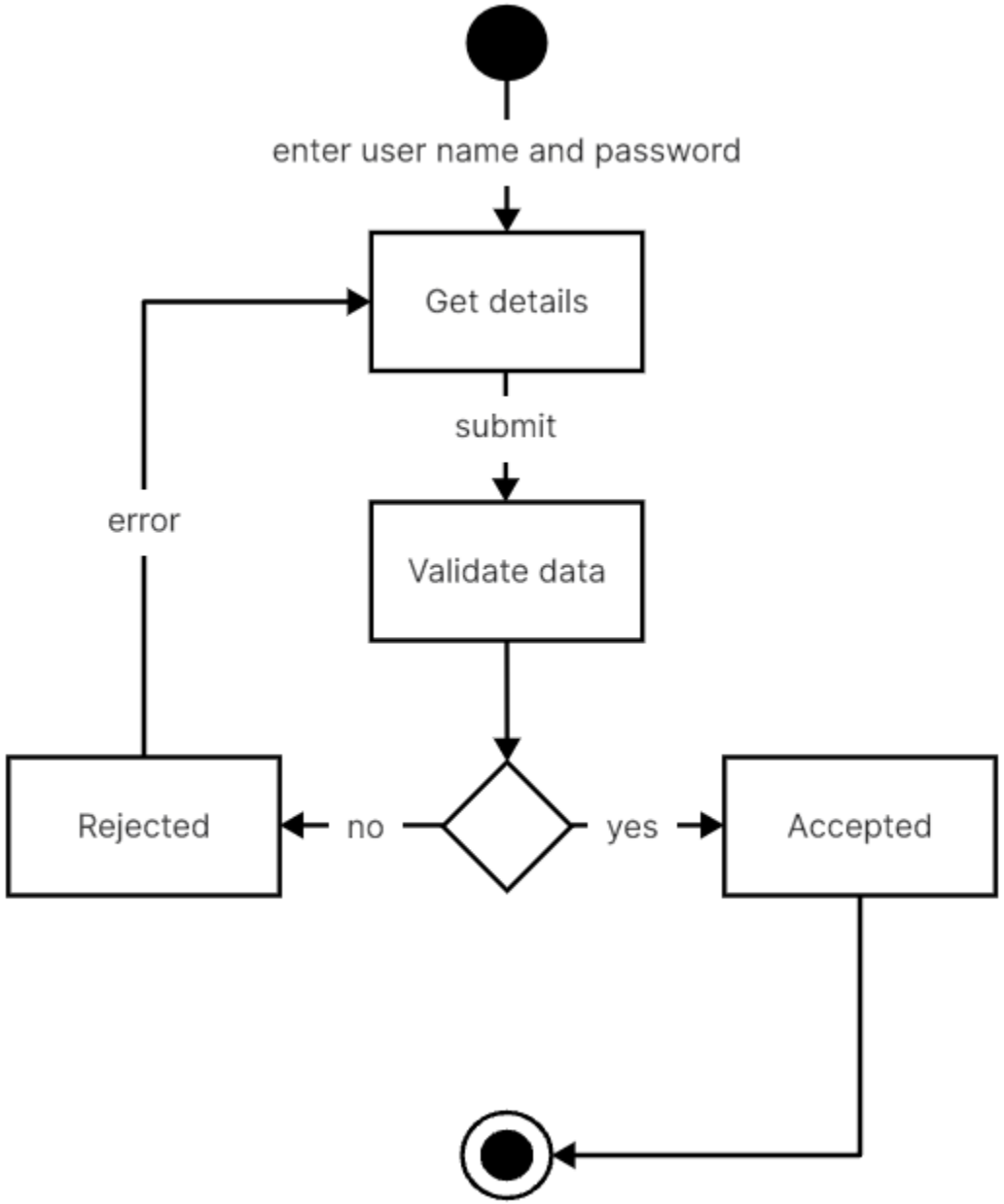


- **Context models** simply **show** the other systems in the environment, not **how** the system being developed is used in that environment.

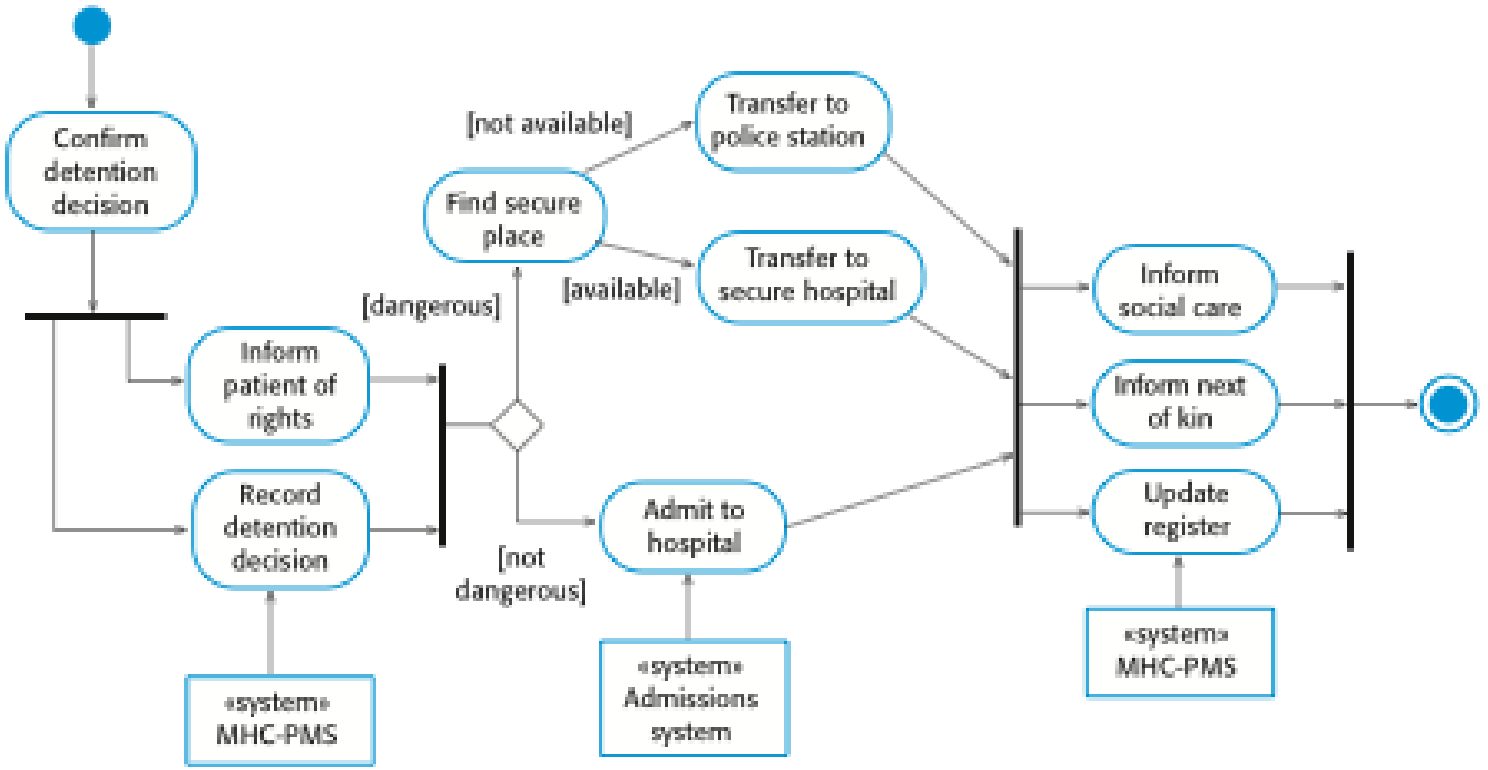
- **Process models** reveal how the system being developed is used in broader business processes.
- How it 'works' Detailed.

- UML activity diagrams may be used to define business process models.

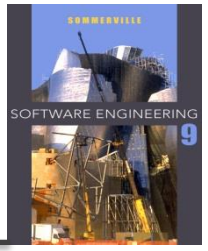
Example


enter user name and password
Get details
submit
Validate data
error
Rejected
no
yes
Accepted

Example


Confirm detention decision
Inform patient of rights
Record detention decision
«system» MHC-PMS
[dangerous]
[not dangerous]
Find secure place
[not available]
Transfer to police station
[available]
Transfer to secure hospital
Admit to hospital
«system» Admissions system
Inform social care
Inform next of kin
Update register
«system» MHC-PMS

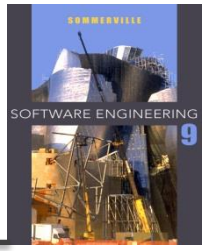
2. Interaction Models



- Modeling user interaction is used to identify user requirements.
- We see **structural** connections and **dynamic** (**behavioral**) interactions.
- We do this with graphical models.

- **Use case diagrams and sequence diagrams may be used for interaction modeling.**
- **These are the most popular modeling mechanisms**

Use Case Modeling (Interaction Model)



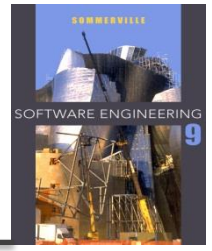
- Use cases were developed originally to support **requirements elicitation** and now incorporated into the UML.

- Each use case represents a **discrete task** that involves **external interaction** with a system.

- **Actors** in a use case may be people, devices, or other systems.

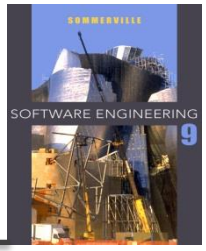
- Represented diagramatically to provide an overview of the use case and in a more detailed textual form.

Transfer-data Use Case Diagram (graphical model)



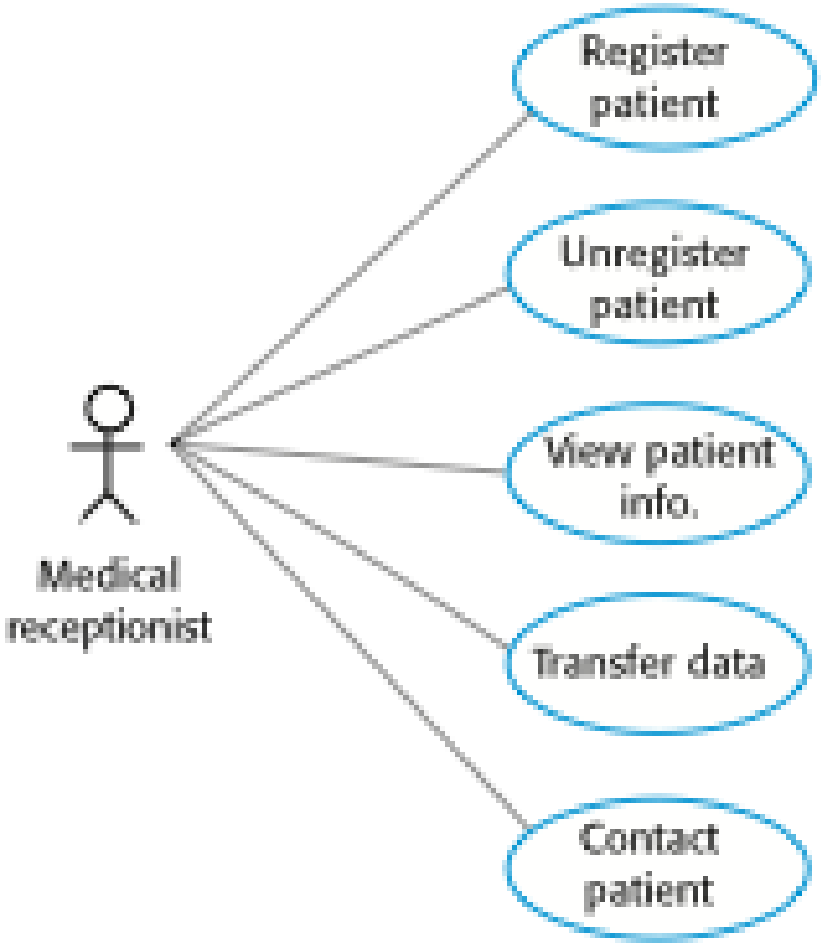
❿ A use case in the MHC-PMS

Tabular Description of the ‘Transfer data’ use-case

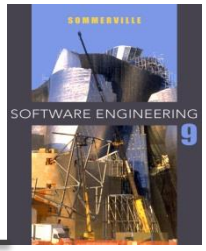


MHC-PMS: Transfer data	
Actors	Medical receptionist, patient records system (PRS)
Description	A receptionist may transfer data from the MHC-PMS to a general patient record database that is maintained by a health authority. The information transferred may either be updated personal information (address, 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.

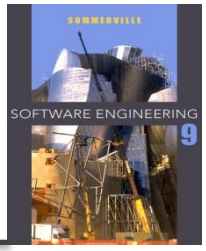
Use Cases in the MHC-PMS involving the role 'Medical Receptionist' (only showing one actor here)



Use case description in a tabular format:

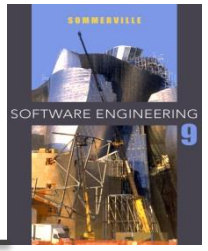


Use case title	Transfer data
Description	A receptionist may transfer data from the MHC-PMS to a general patient record database that is maintained by a health authority. The information transferred may either be updated personal information (address, phone number, etc.) or a summary of the patient's diagnosis and treatment.
Actor(s)	Medical receptionist, patient records system (PRS)



Preconditions	Patient data has been collected (personal information, treatment summary); The receptionist must have appropriate security permissions to access the patient information and the PRS.
Postconditions	PRS has been updated
Main success scenario	1. Receptionist selects the "Transfer data" option from the menu. 2. PRS verifies the security credentials of the receptionist. 3. Data is transferred. 4. PRS has been updated.
Extensions	2a. The receptionist does not have the necessary security credentials. 2a.1. An error message is displayed. 2a.2. The receptionist backs out of the use case.

Sequence Diagrams (Interaction Model)



- **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.

Example

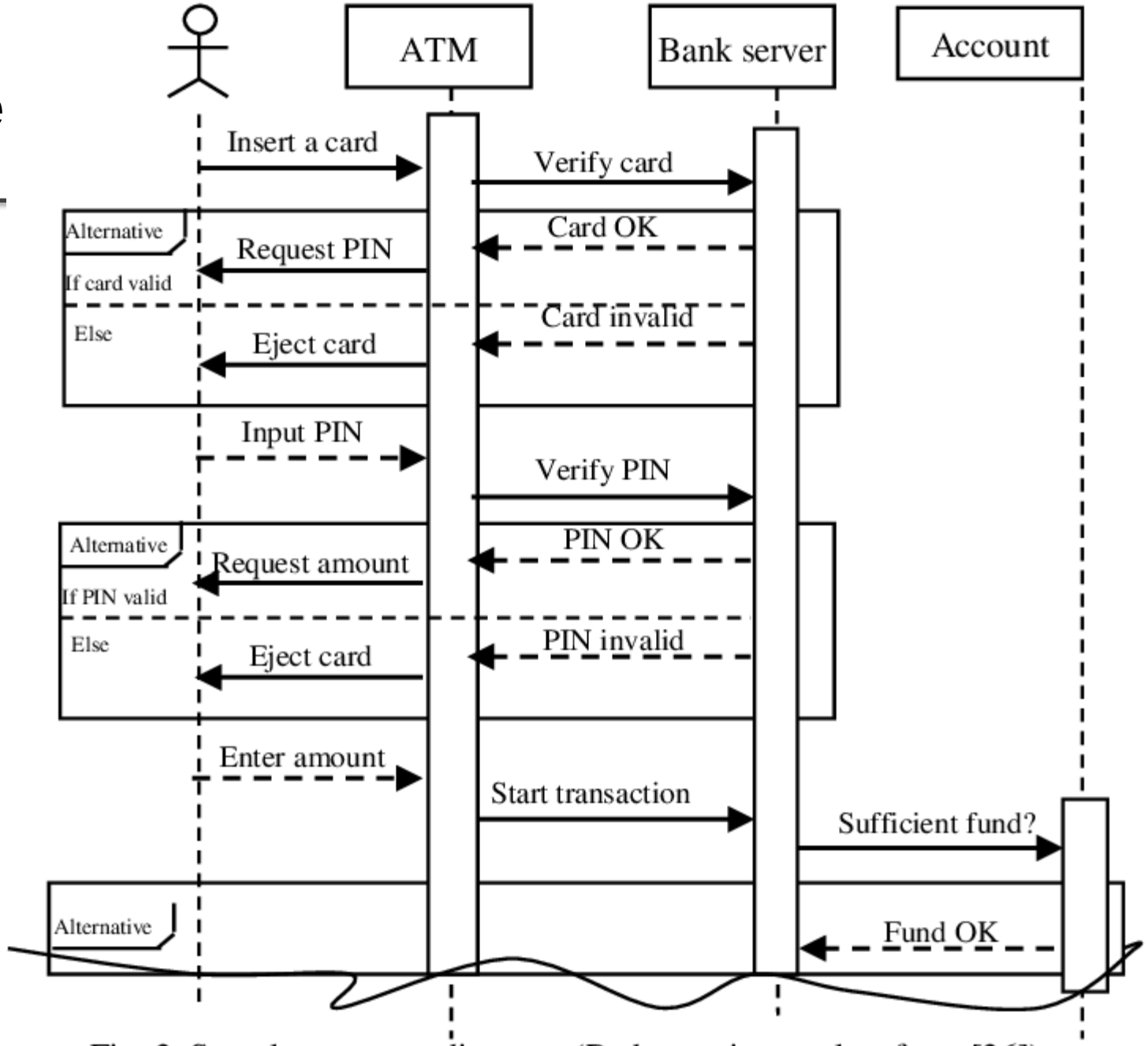
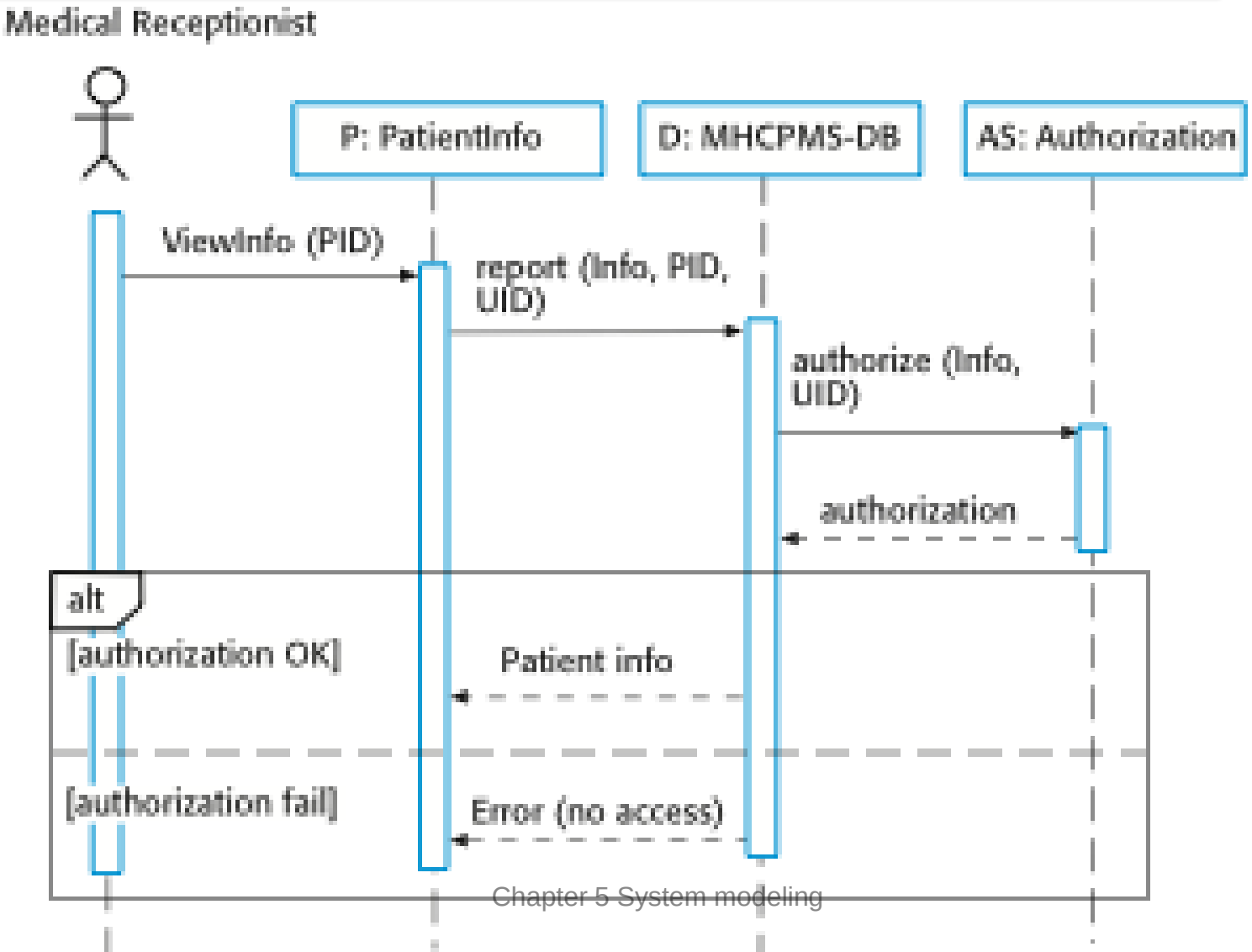
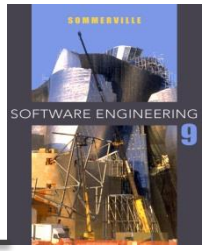


Fig. 2. Sample sequence diagram. (Redrawn, incomplete from [26])

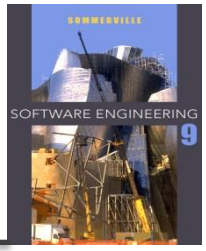
Sequence diagram for View Patient Information



Structural models

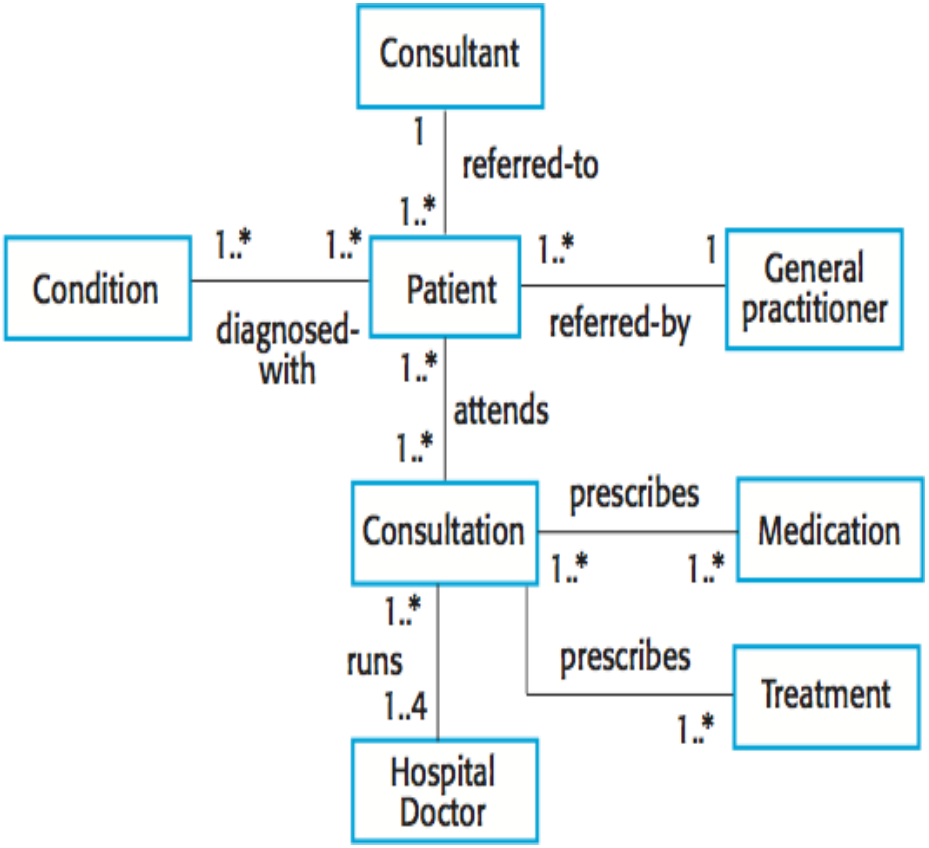


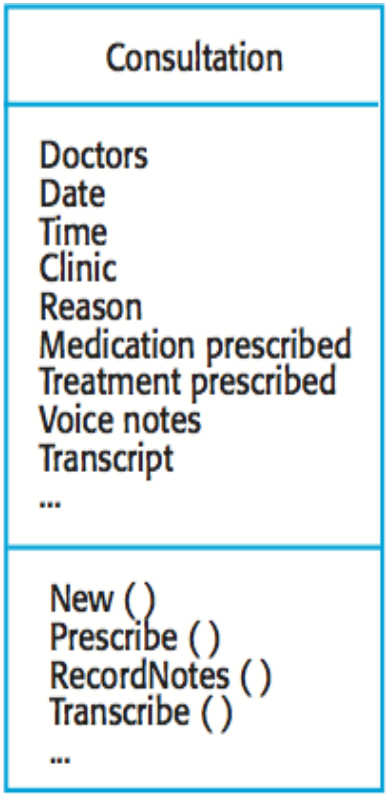
- Structural models of software display the organization of a system in terms of the components that make up that system and their relationships.
- Structural models may be static models, which show the structure of the system design, or dynamic models, which show the organization of the system when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



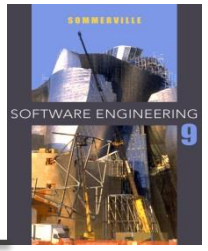
- UML class diagrams are used when developing an object-oriented system model to show the classes in a system and the associations between these classes.
- An object class can be thought of as a general definition of one kind of system object.
- An association is a link between classes that indicates 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,

ERD


Consultant
1
referred-to
1..*
Condition
1..*
1..*
Patient
1..*
1
General practitioner
diagnosed-with
referred-by
1..*
attends
1..*
Consultation
prescribes
1..*
1..*
Medication
1..*
runs
1..4
prescribes
1..*
Treatment
Hospital Doctor

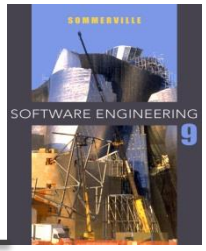

Consultation
Doctors
Date
Time
Clinic
Reason
Medication prescribed
Treatment prescribed
Voice notes
Transcript
...
New ( )
Prescribe ( )
RecordNotes ( )
Transcribe ( )
...

Generalization

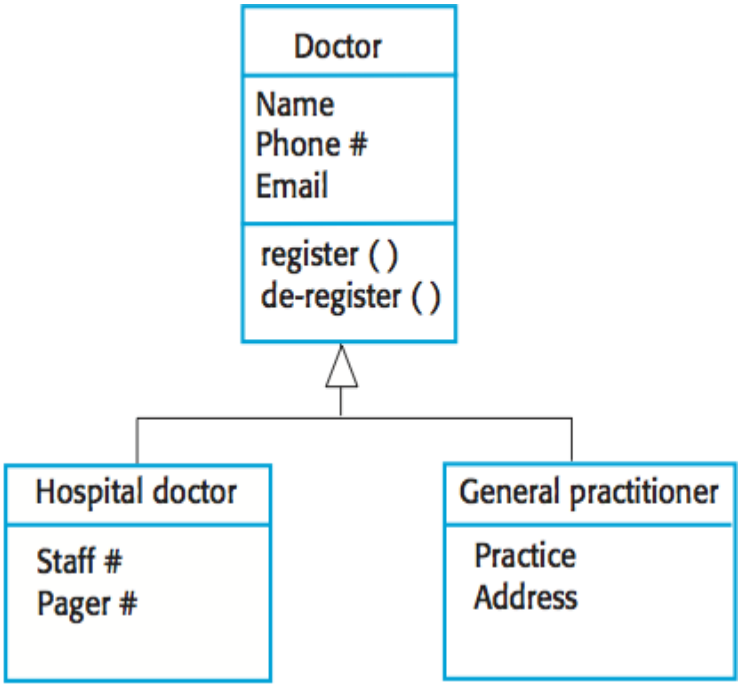


- Generalization is an everyday technique that we use to manage complexity.
- In modeling systems, it is often useful to examine the classes in a system to see if there is scope for generalization.
- In object-oriented languages, such as Java, generalization is implemented using the class inheritance mechanisms built into the language.

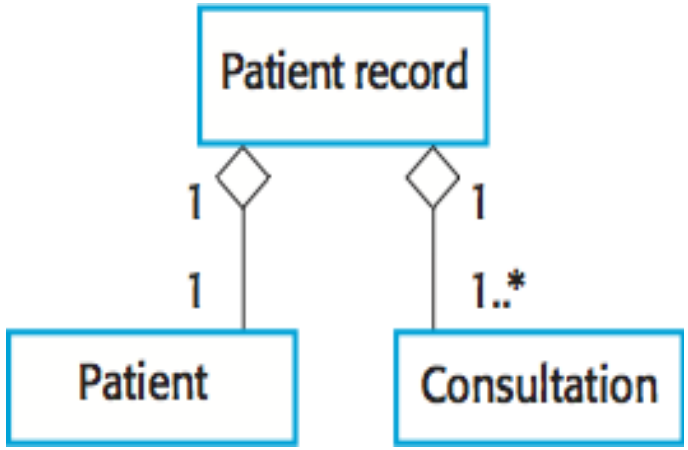
Generalization



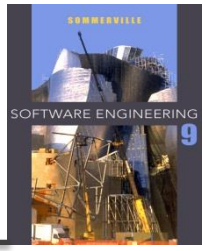
- In a generalization, the attributes and operations associated with higher-level classes are also associated with the lower-level classes.
- The lower-level classes are subclasses inherit the attributes and operations from their superclasses.
- These lower-level classes then add more specific attributes and operations.


Doctor
Name
Phone #
Email
register ( )
de-register ( )
Hospital doctor
Staff #
Pager #
General practitioner
Practice
Address

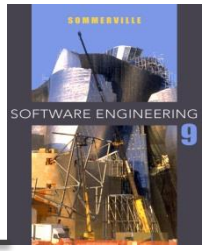
An aggregation model shows how classes that are collections are composed of other classes. Aggregation models are similar to the part-of relationship in semantic data models.



Behavioral models

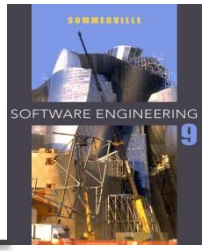


- ✧ Behavioral models are models of the dynamic behavior of a system as it is executing.
- ✧ They show what happens or what is supposed to happen when a system responds to a stimulus from its environment. Two types of stimuli:
  - Some data arrives that has to be processed by the system.
  - Some event happens that triggers system processing. Events may have associated data, although this is not always the case.

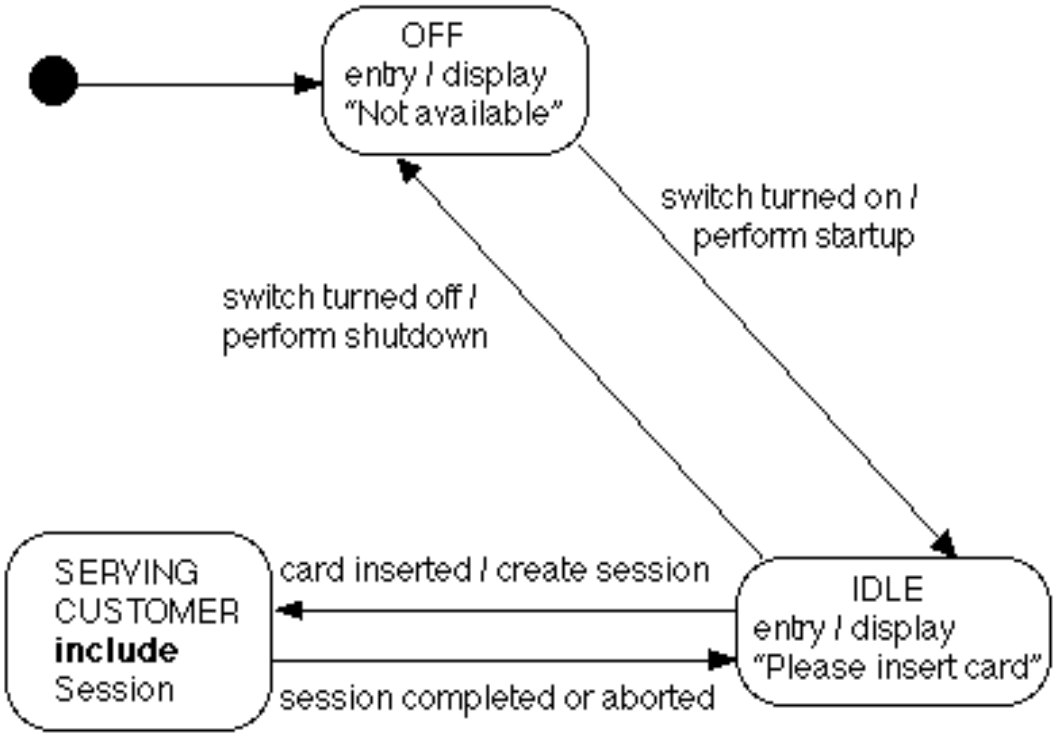


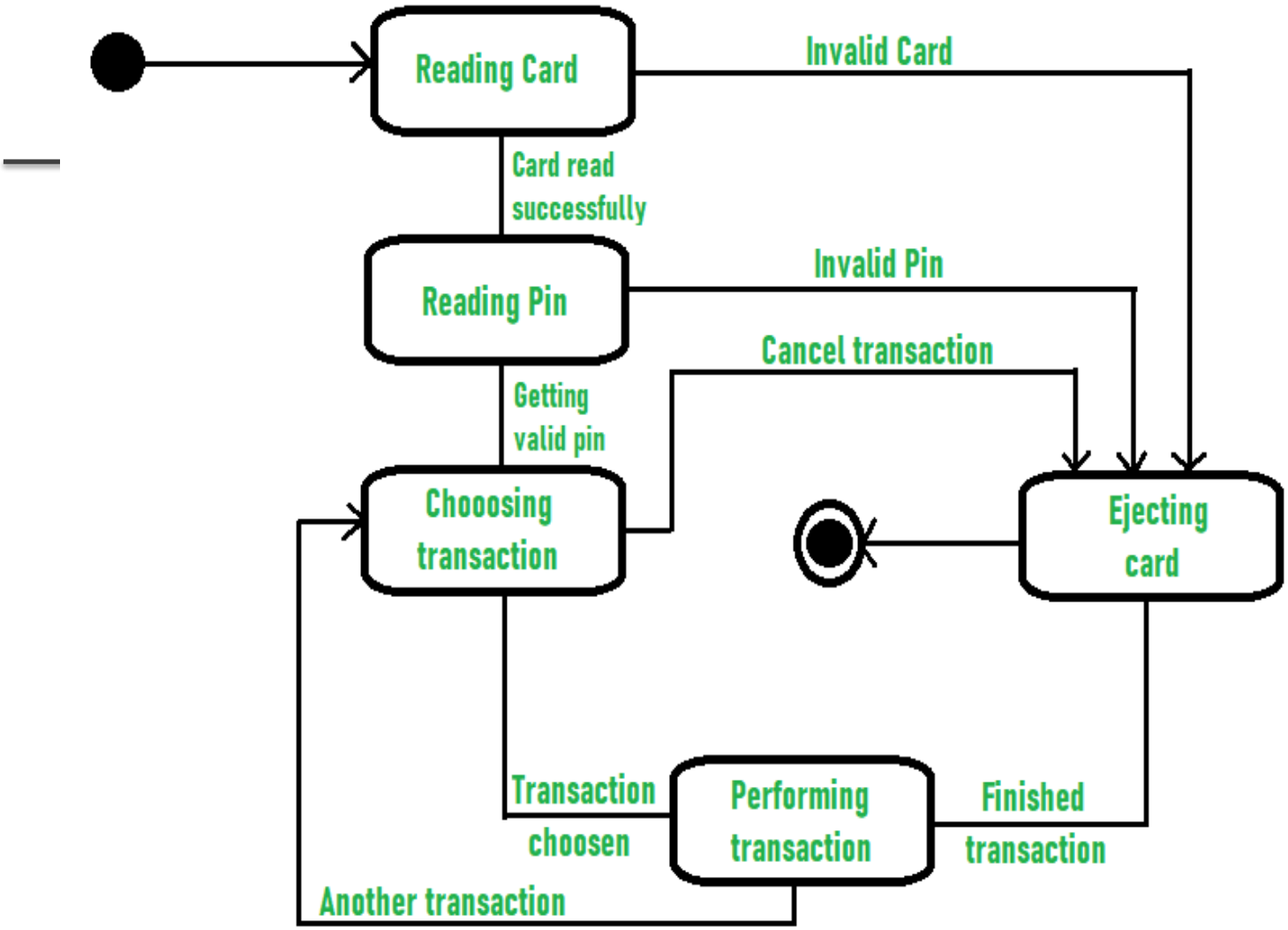
- Many business systems are data-processing systems that are primarily driven by data. They are controlled by the data input to the system, with relatively little external event processing.
- Data-driven models show the sequence of actions involved in processing input data and generating an associated output.
- They are particularly useful during the analysis of requirements as they can be used to show end-to-end processing in a system. Data-driven models can be created using UML activity diagrams:
- And Event driven systems can use State diagrams and State Transition Diagrams to represent the system.

Example



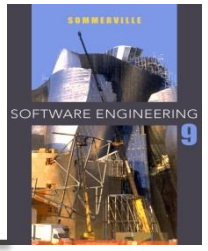
State-Chart for Overall ATM (includes System Startup and System Shutdown Use Cases)





State Transition Diagram for ATM System

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.