Lecture05: Analysis Model

EGCI340: SOFTWARE DESIGN

Outline

Data Modeling Concepts

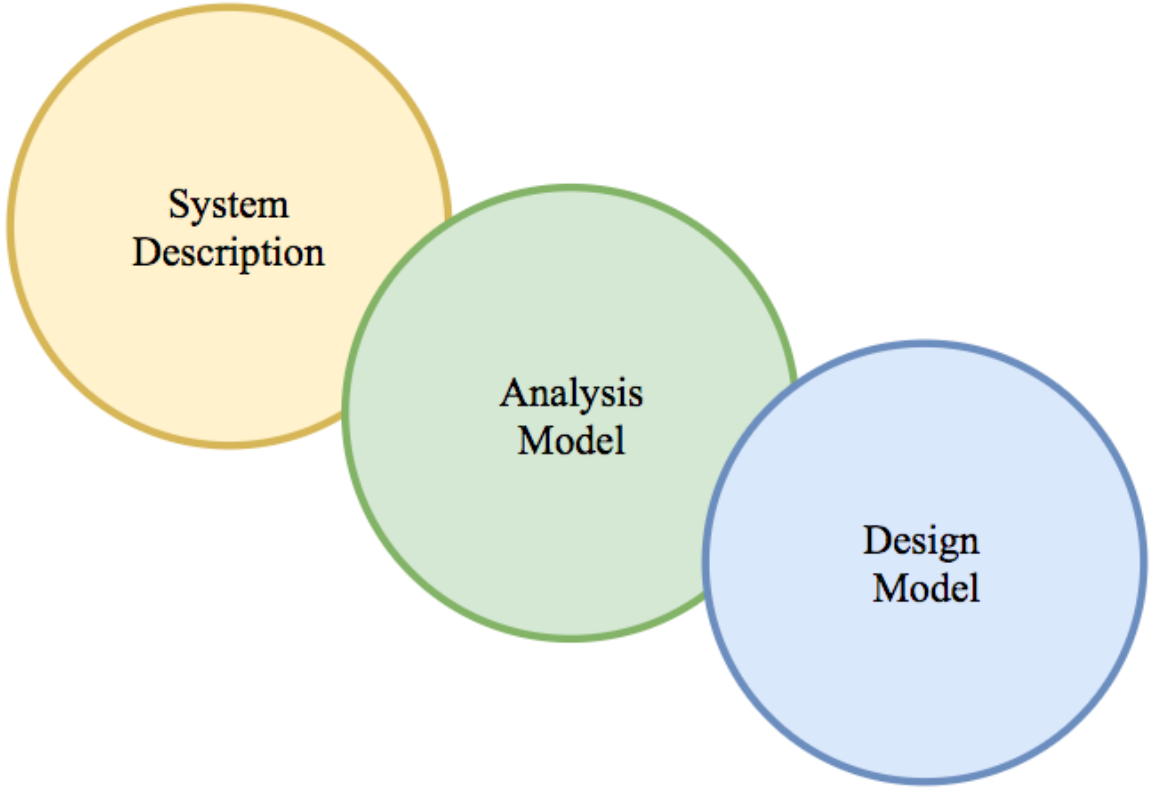
Object-Oriented Analysis

Elements of Analysis Model

- Scenario-Based Modeling
  - Use-Cases
  - Activity Diagram
- Flow-Oriented Modeling

Requirement Analysis

The analysis model as a bridge between the system description and the design model.



Data Modeling Concepts

Analysis modeling often begins with ***data modeling***

Data Modeling is composed of :

Data object

- A representation of any composite information that is processed by software

Data attributes

- The properties of a data object and take on one of three different characteristics
  - Name an instance of the data object
  - Describe the instance
  - Make reference to another instance in another table

Relationship

- The manner in which data objects are connected to one another

Object-Oriented Analysis

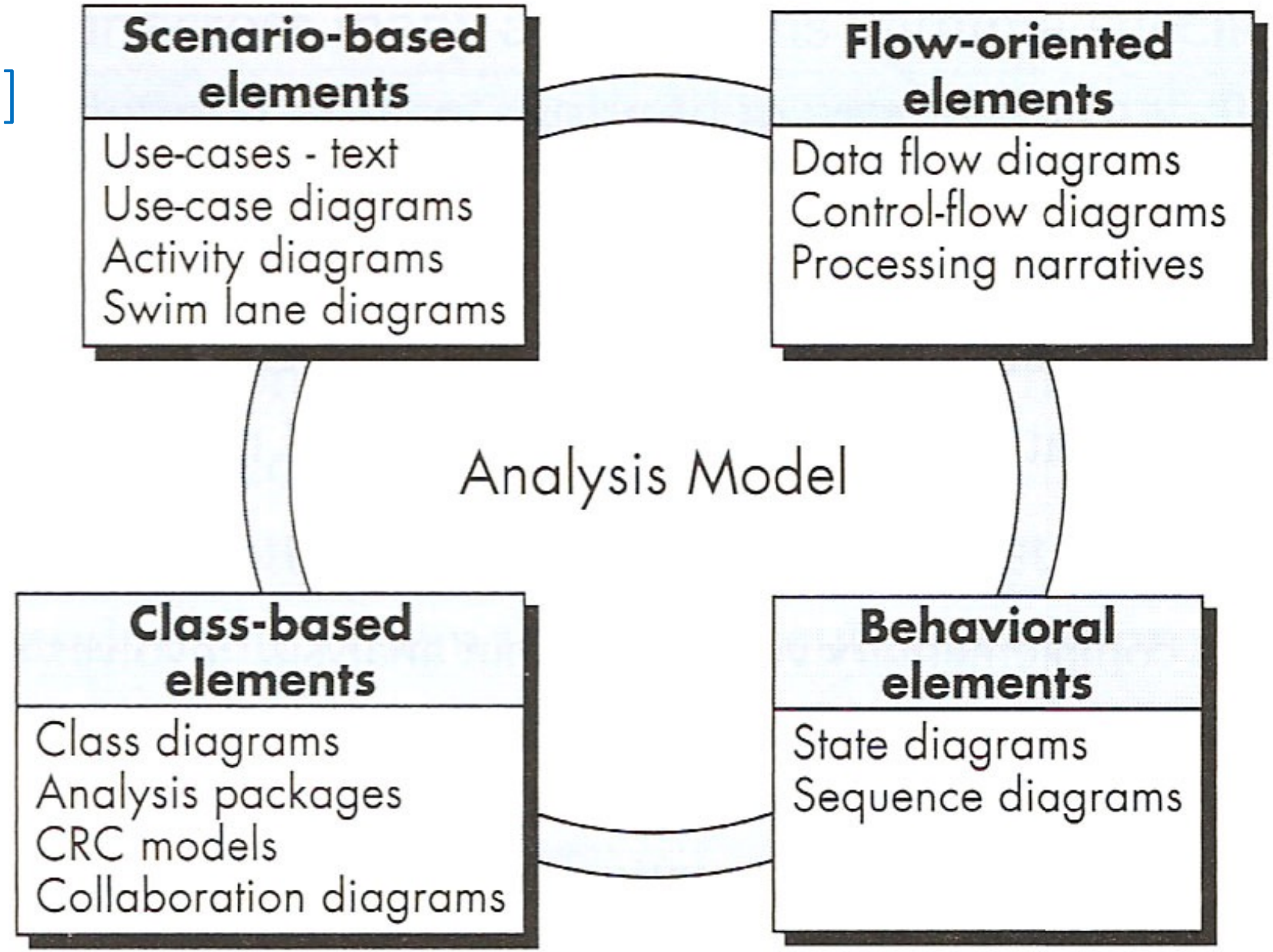
Object-oriented analysis (OOA) is to define all classes

- Including the relationships and behavior associated with classes
- Classes are relative to the problem to be solved

To accomplish OOA, a number of tasks must occur:

- Basic user requirements must be communicated between the customer and the software engineer
- Classes must be identified
- Class hierarchy is defined
- Object-to-object relationships should be represented
- Object behavior must be modeled
- All tasks are reapplied iteratively until the model is complete

Elements of Analysis Model [1]



Scenario-based Modeling

Success of a computer-based system or product is measured in many ways:

- User satisfaction resides at the top of the list

If software engineer understands how end-users want to interact with a system,

- Software team will be better able to properly characterize requirements and build meaningful analysis and design models

Analysis modeling with UML begins with the creation of scenarios in the form of :

- Use-cases, activity diagrams, and swim lane diagrams

Use-case Diagram

Use-case diagrams is functional diagram in that they portray the basic function of the system

- What the users can do
- How the system should respond to the user's action

Creating use-case diagram is a two-step:

1) Users work with the project team to write text-based **use-case description**
2) Project team translates the use-case descriptions into formal **use-case diagrams**

Elements of a Use-case Description

Overview Information

Relationships

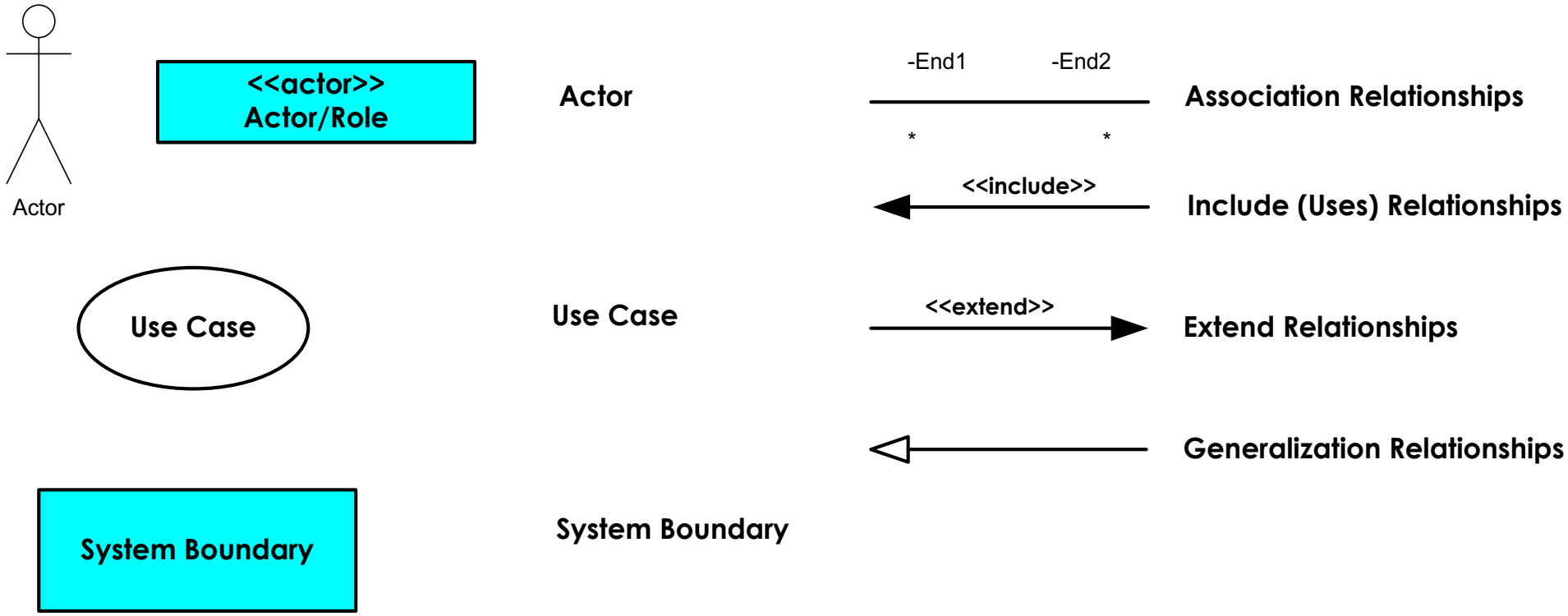
- Association
- Extend
- Include
- Generalization

Flow of Events

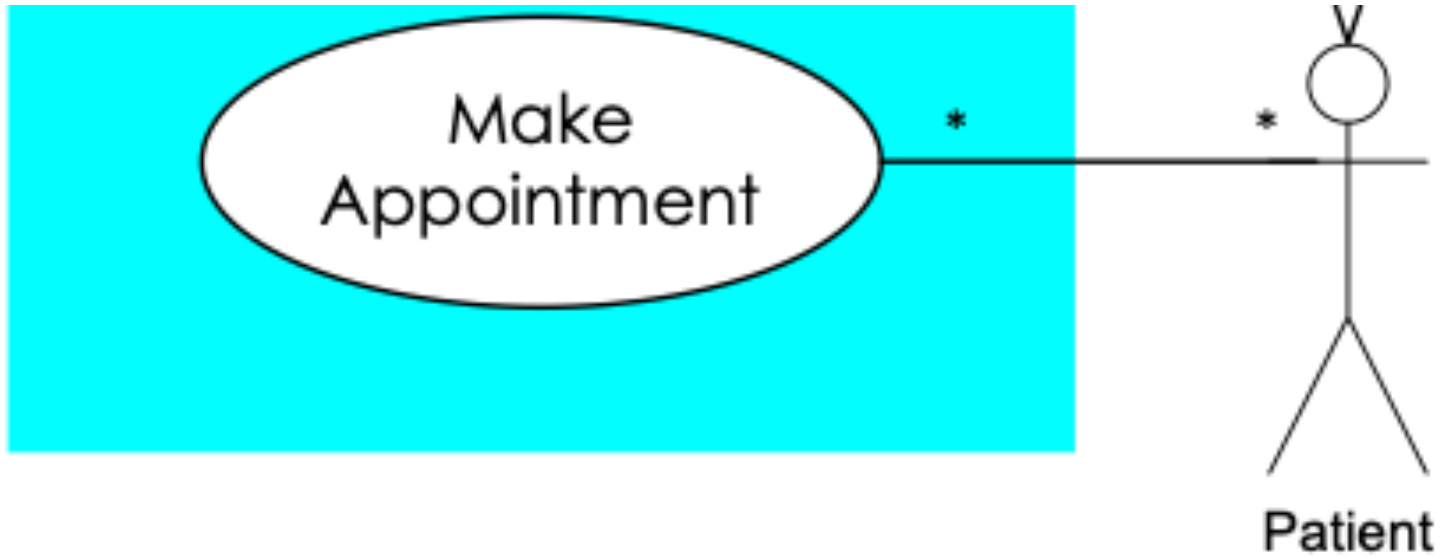
- Normal Flow of Events
- Subflows
- Alternate or Exceptional Flows

Optional Characteristics

Use-case Diagram



Create a Use-case Description for the diagram below



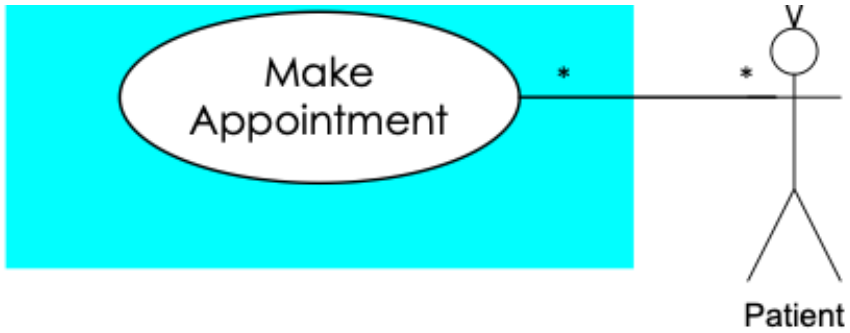
An Example of Use-case Description

Overview Information

Patient can make appointment

Relationships

Association



Flow of Events

Normal Flow

Optional Characteristics

Another Example of Use-case Description

Overview Information

Patients are composed of new patients and old patients

Relationships

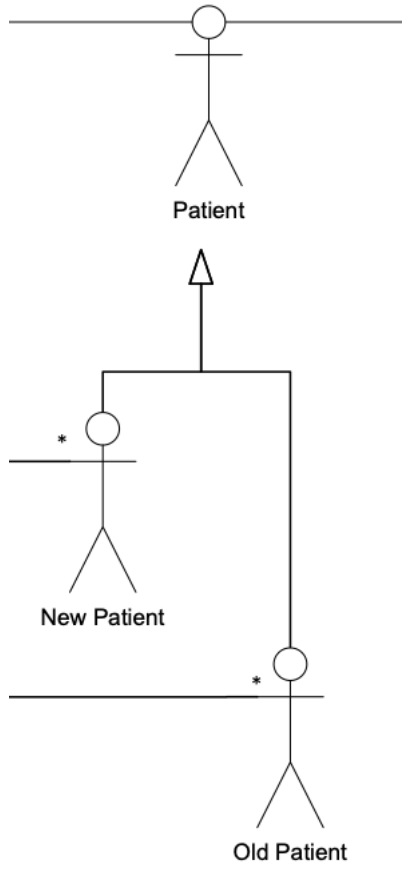
Generalization

Flow of Events

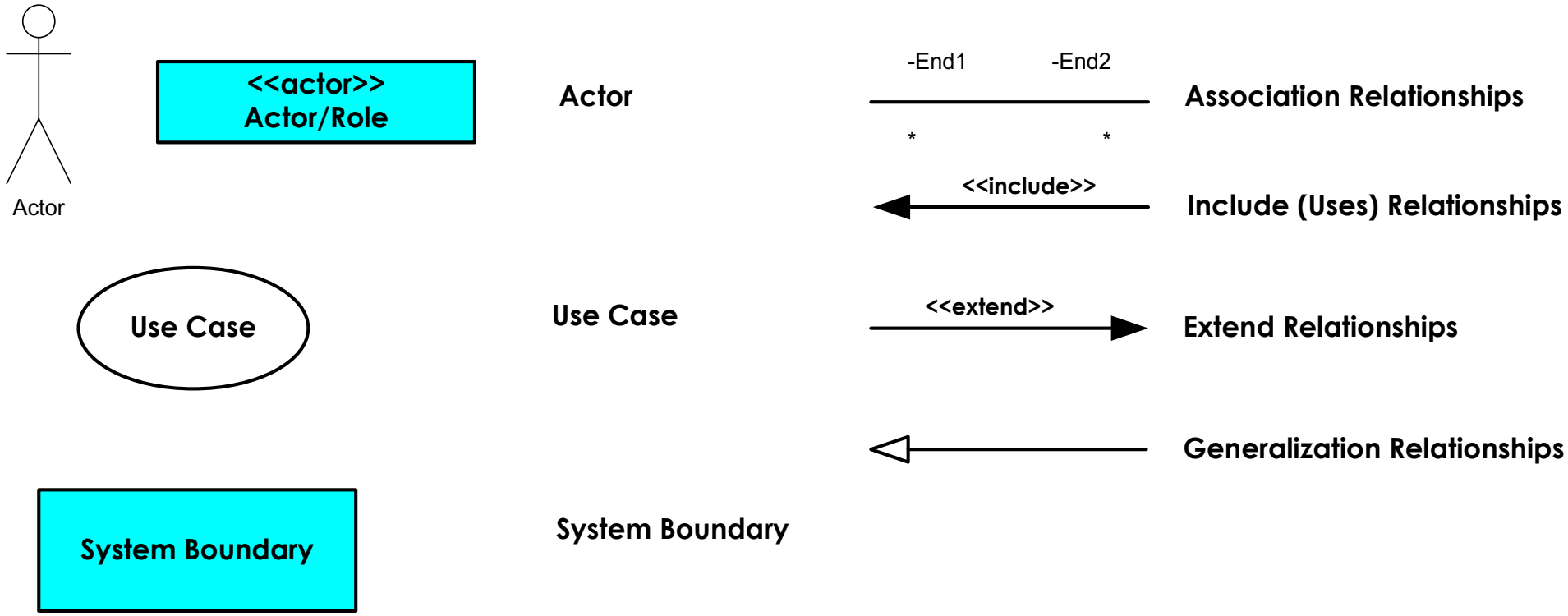
Normal Flow

Optional Characteristics

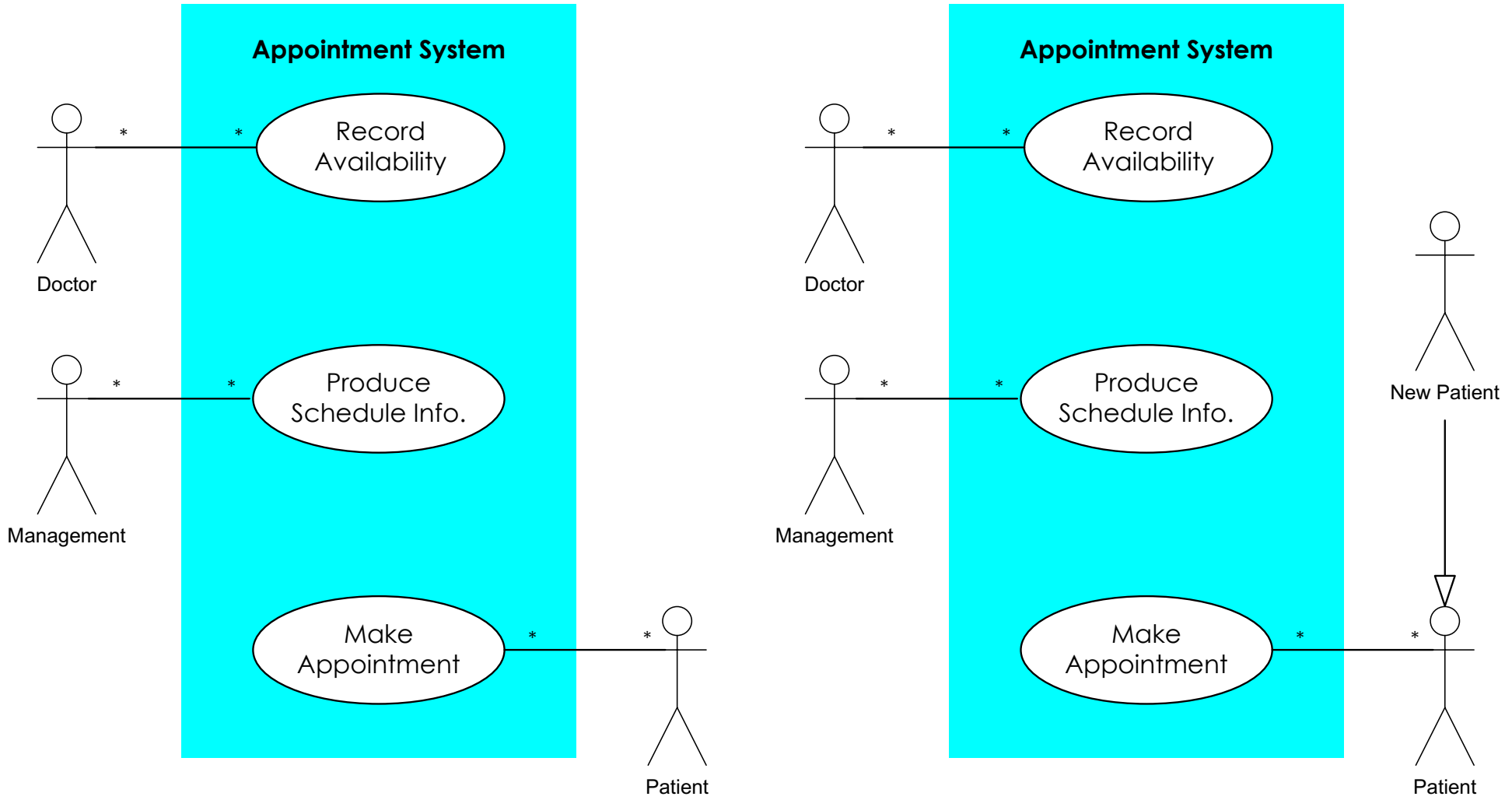
New Patient will turn to be old patient after creating patient account



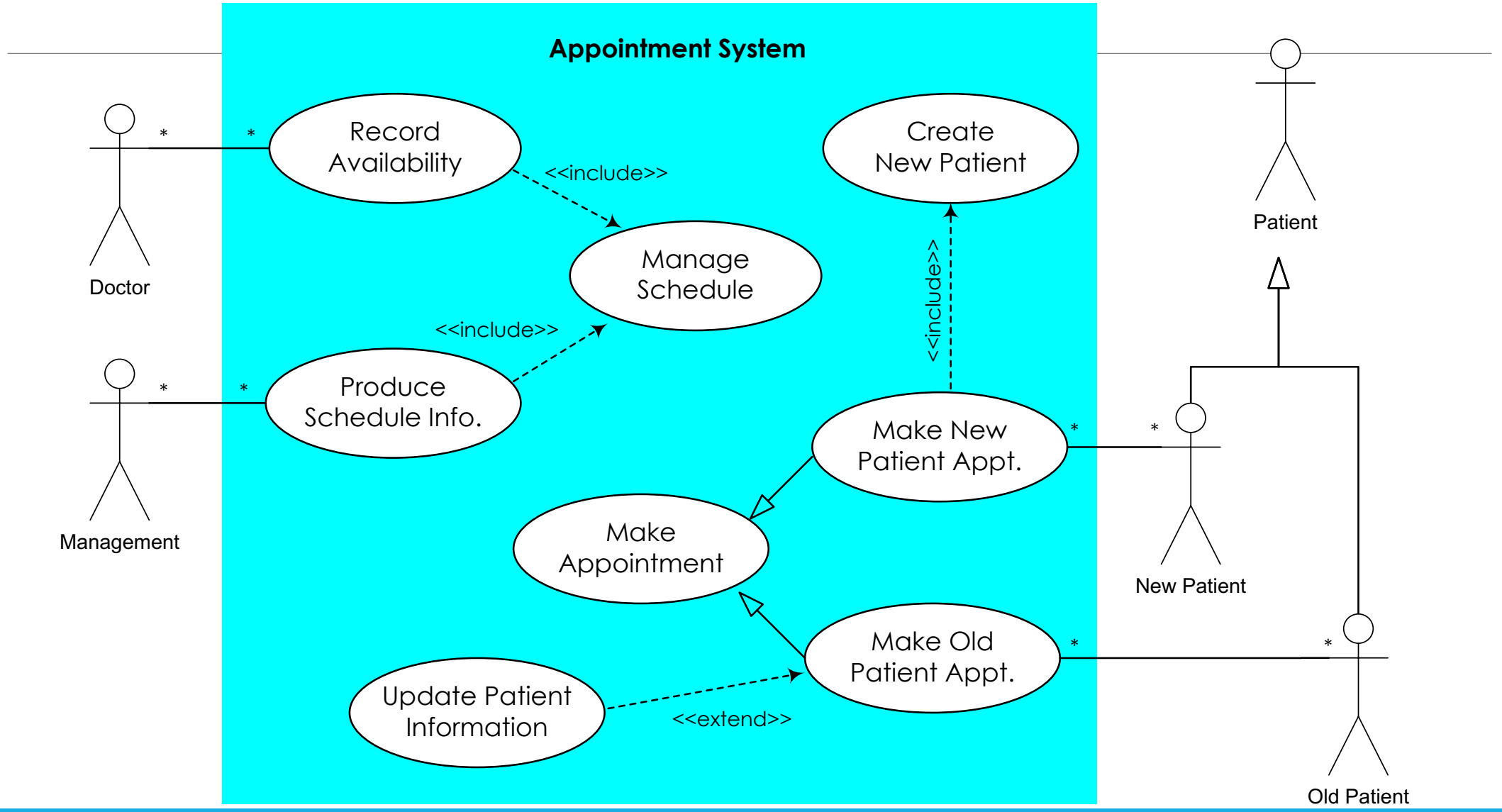
Use-case Diagram



Example 1: Use-case Diagram for Appointment System



Example 1: Extend and Include Relationships (Cont.)

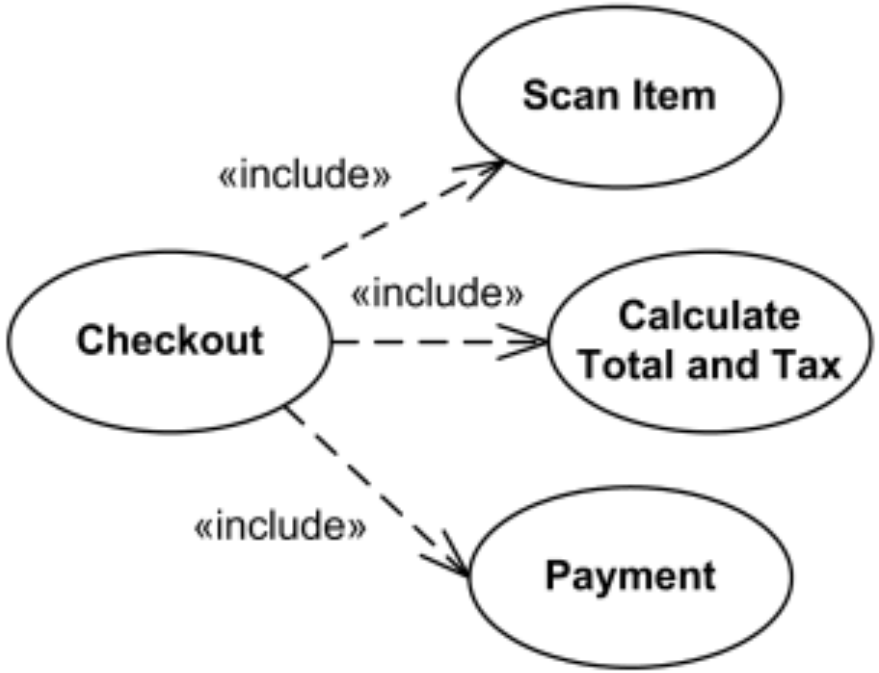


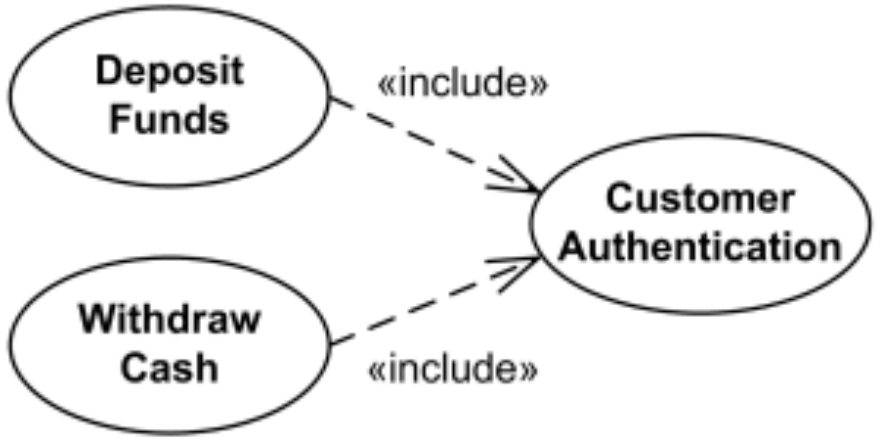
Extend and Include Relationships (Cont.)

- Process steps for include relationship
  - The process of “checking record availability” for doctor
  - Step 1: Do “Manage schedule process”
  - Step 2: Do “Check record availability”
- Process steps for extend relationship
  - The process of “Make old appointment” for old patient
  - Step1: Do “Make old appointment”
  - Step2: Do “Update patient information” (Optional step)

Use-case Diagram: Include Relationship

Reference: http://www.uml-diagrams.org/use-case-include.html





Checkout use case includes several use cases - **Scan Item, Calculate Total and Tax, and Payment**

Deposit Funds and **Withdraw Cash** use cases include **Customer Authentication** use case.

Write the process steps of

1. Deposit funds

2. Withdraw cash

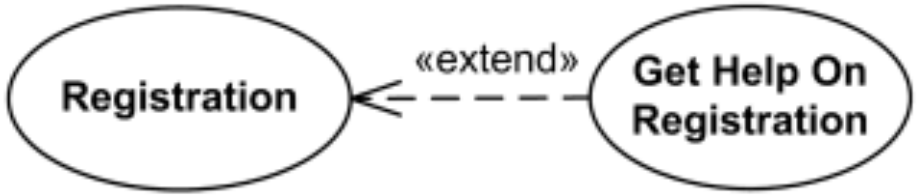
Write the process steps of

1. Deposit funds
   a) Do “Customer authentication”
   b) Do “Deposit funds”

2. Withdraw cash
   a) Do “Customer authentication”
   b) Do “Withdraw cash”

Use-case Diagram: Extend Relationship

Reference: http://www.uml-diagrams.org/use-case-extend.html



Extending use case is optional, supplementary

Registration use case is complete and meaningful on its own. It could be extended with optional **Get Help On Registration** use case.

Example 1.1: Login System Scenario

- User can **login** to the system
- For the first time, a new user must register to the system
- Administrator can manage user's accounts and approve new account

Example 1.1: Use-case Diagram for Login System

(Draw the use case diagram)

Actor:

- Admin
- User

Use case: (For Admin)

- Login Management
- Approve new account

Use case: (For User)

- Login
- Register and Login (For new user)

Example 1.1: Use-case diagram for Login System

Draw diagram here

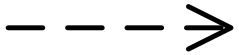
Activity Diagram

Elements of an activity diagram

Action or Activity

Object Node

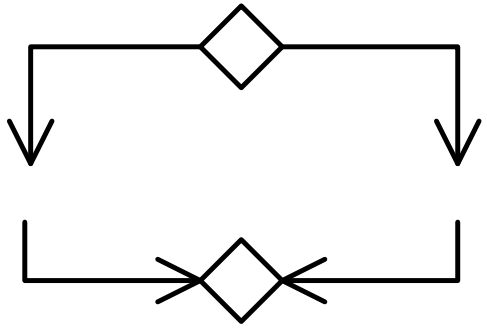
Control Flow



Object Flow

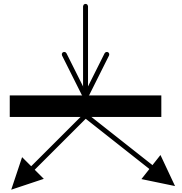
Initial Node

Final-Activity Node

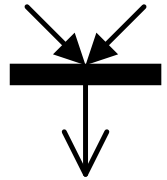


Decision Node

Merge Node



Fork Node



Join Node

Swimlane

Example 2: Activity Diagram for Appointment System

(Draw the activity diagram)

Action

- Get patient Information
- Create new patient

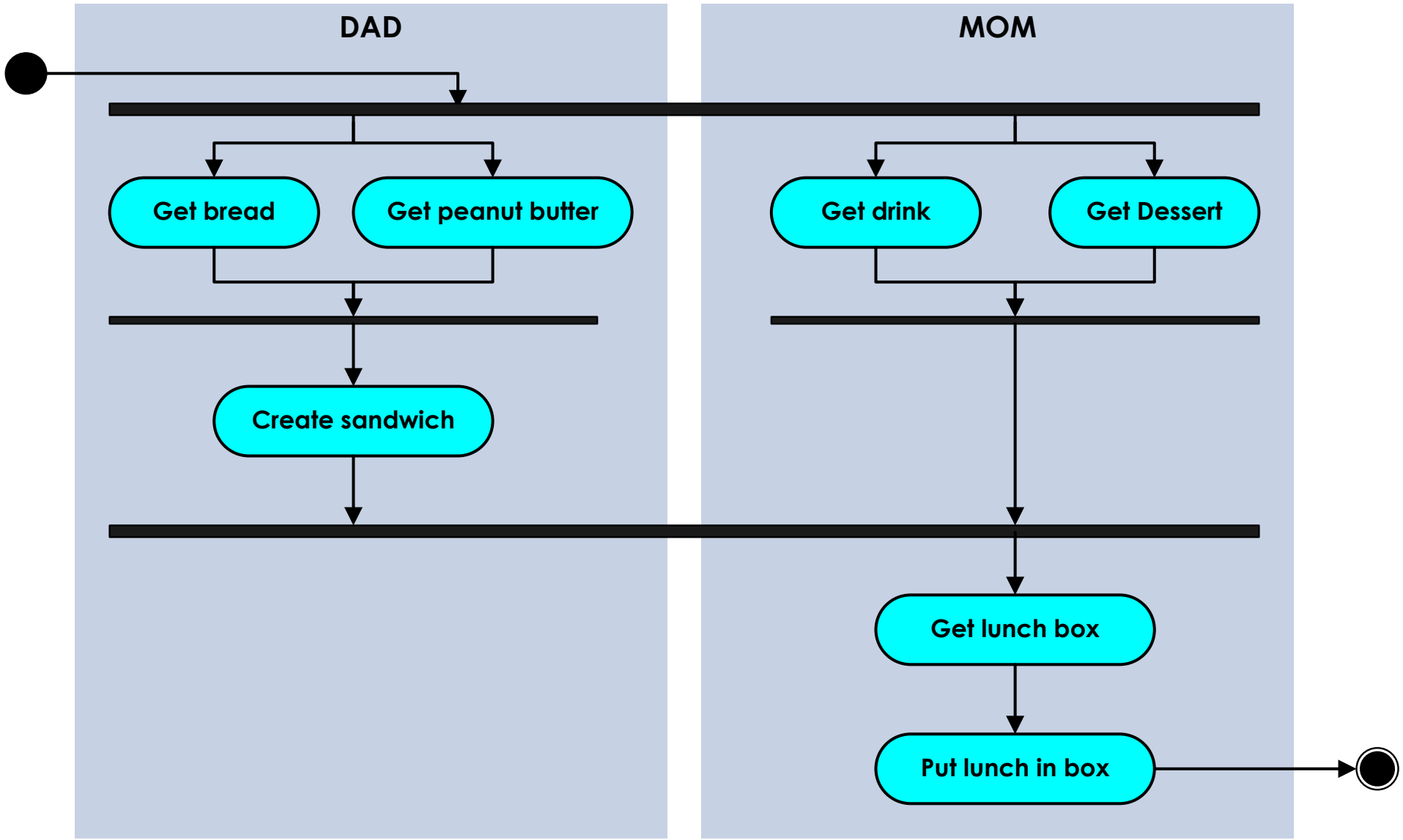
  (Old patient can skip this action)
- Make payment arrangements

  (Both old and new patients must go to “make payment arrangements” process)
- Create appointment
- Cancel appointment
- Change appointment

Example 2: Activity Diagram for Appointment System

Draw diagram here

Example 3: Activity Diagram for School Box Lunch



Flow-Oriented Modeling

Data flow-oriented modeling continues to be *one of the most widely used analysis* notations today

Although the data flow diagram (DFD) and related diagrams and information *are not a formal part of UML*,

- They can be used to complement UML diagram and provide additional insight into system requirements and flow

Differences Between DFDs and Flowcharts

Processes on DFDs can operate in parallel (*at-the-same-time*)

- Processes on flowcharts execute one at a time

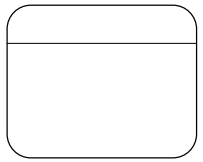
DFDs show the flow of data through a system

- Flowcharts show the flow of control (*sequence and transfer of control*)

Processes on a DFD can have dramatically different timing (*daily, weekly, on demand*)

- Processes on flowcharts are part of a single program with consistent timing

DFDs Symbols



Process

Step by step in structures are followed that transform inputs into outputs

Data Flow

Data flowing from place to place, such as an input or output to a process

External Agent

The source or destination of data outside the system

Data Store

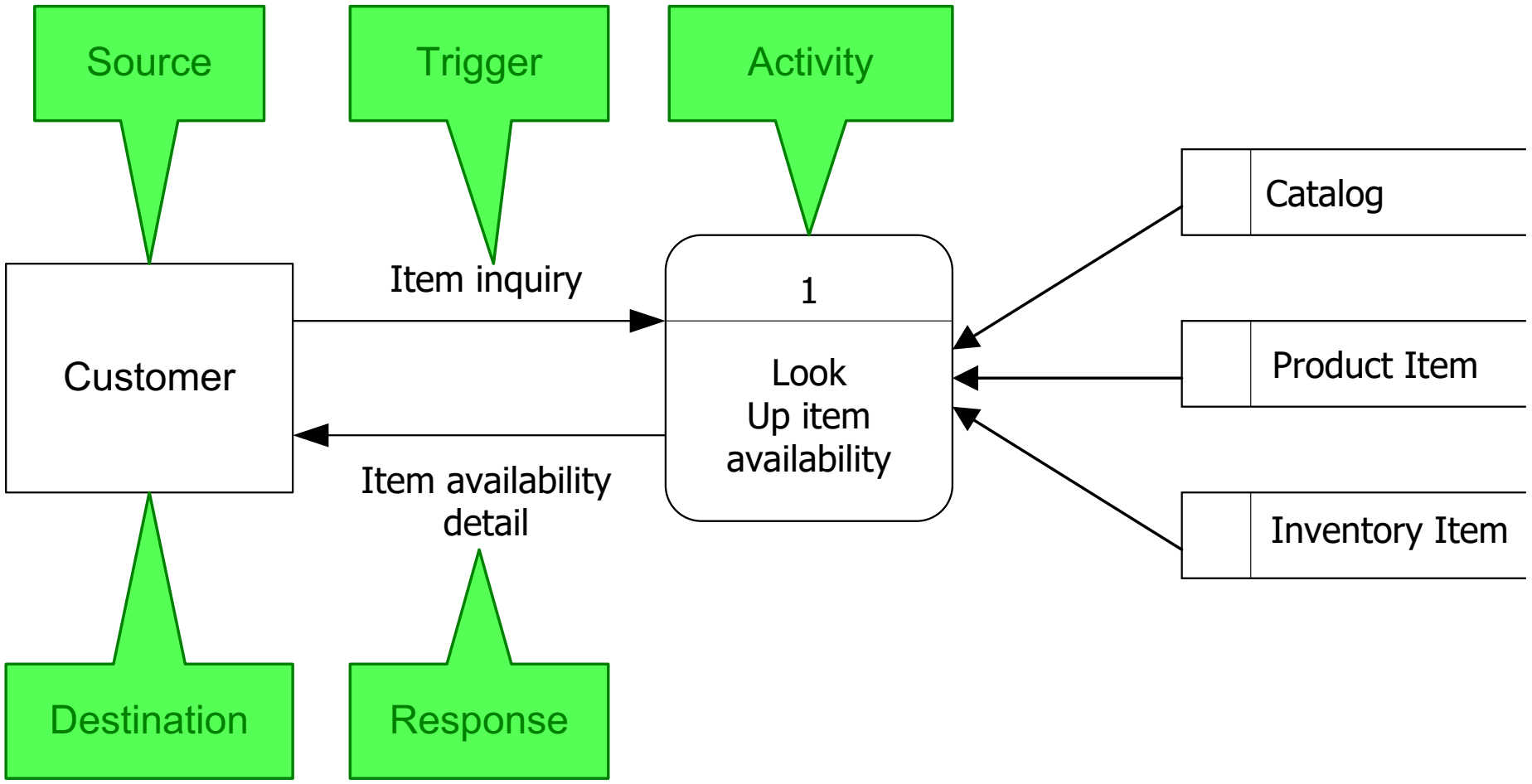
Data at rest, being stored for later use. Usually corresponds to a data entity on an entity-relationship diagram.

Event Table

Information about each event in an event table

Event	Trigger	Source	Activity/Use Case	Response	Destination
Customer wants to check item availability	Item inquery	Customer	Look up item availability	Item availability details	Customer

Event Table


Source
Trigger
Activity
Catalog
Item inquiry
1
Customer
Look
Up item
availability
Product Item
Item availability
detail
Inventory Item
Destination
Response

DFD and Levels of Abstraction

Decomposition

- Any modeling technique that ***breaks*** the system into a ***hierarchical set*** of increasingly more detailed models

Context Diagram

- A DFD that summarized all processing activity within the system in a single process symbol

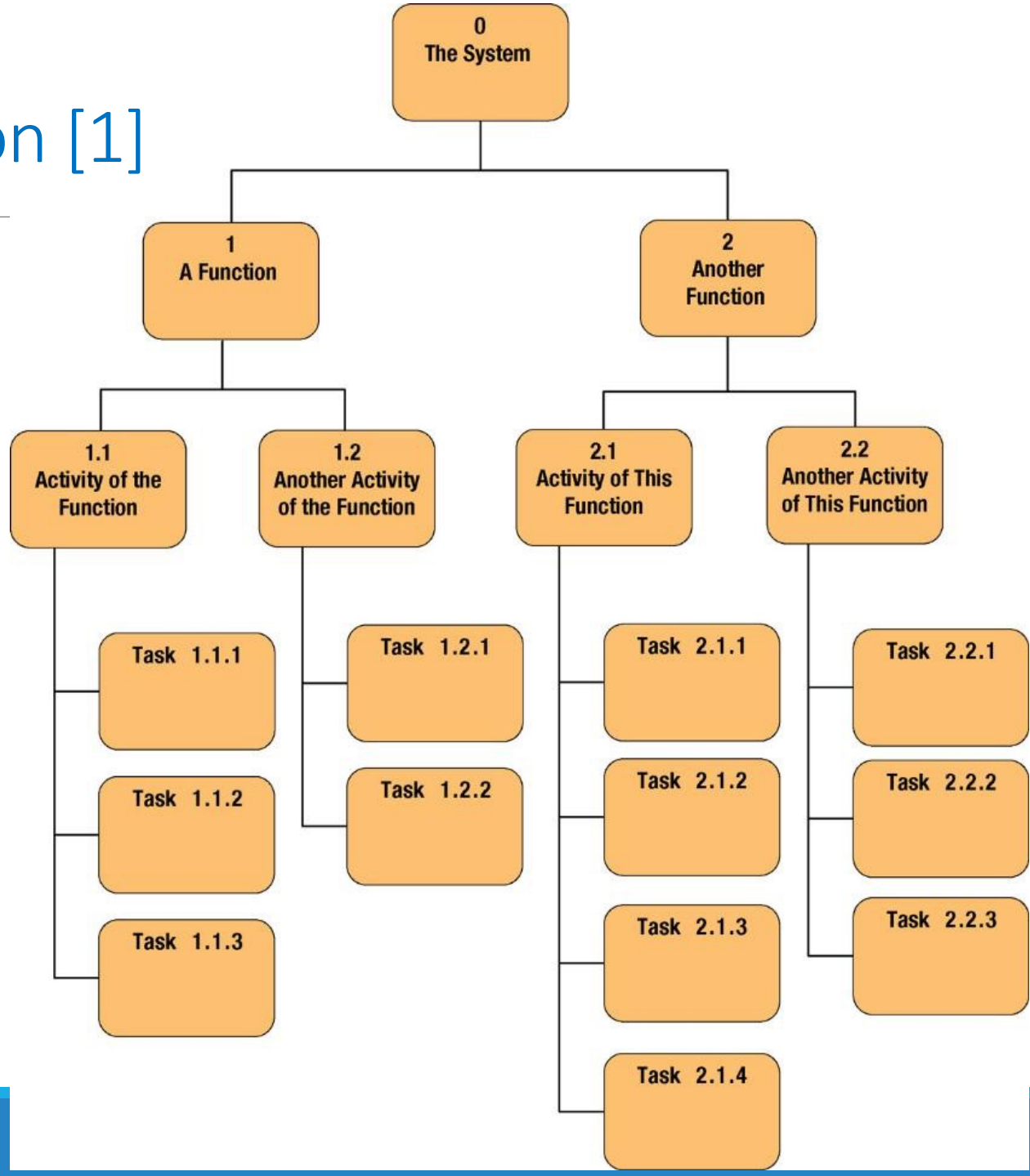
DFD and Levels of Abstraction [1]

Decomposition – the act of breaking a system into sub-components.

- Each level of abstraction reveals more or less detail

Decomposition Diagram - A tool used to depict the decomposition of a system

- Called hierarchy chart

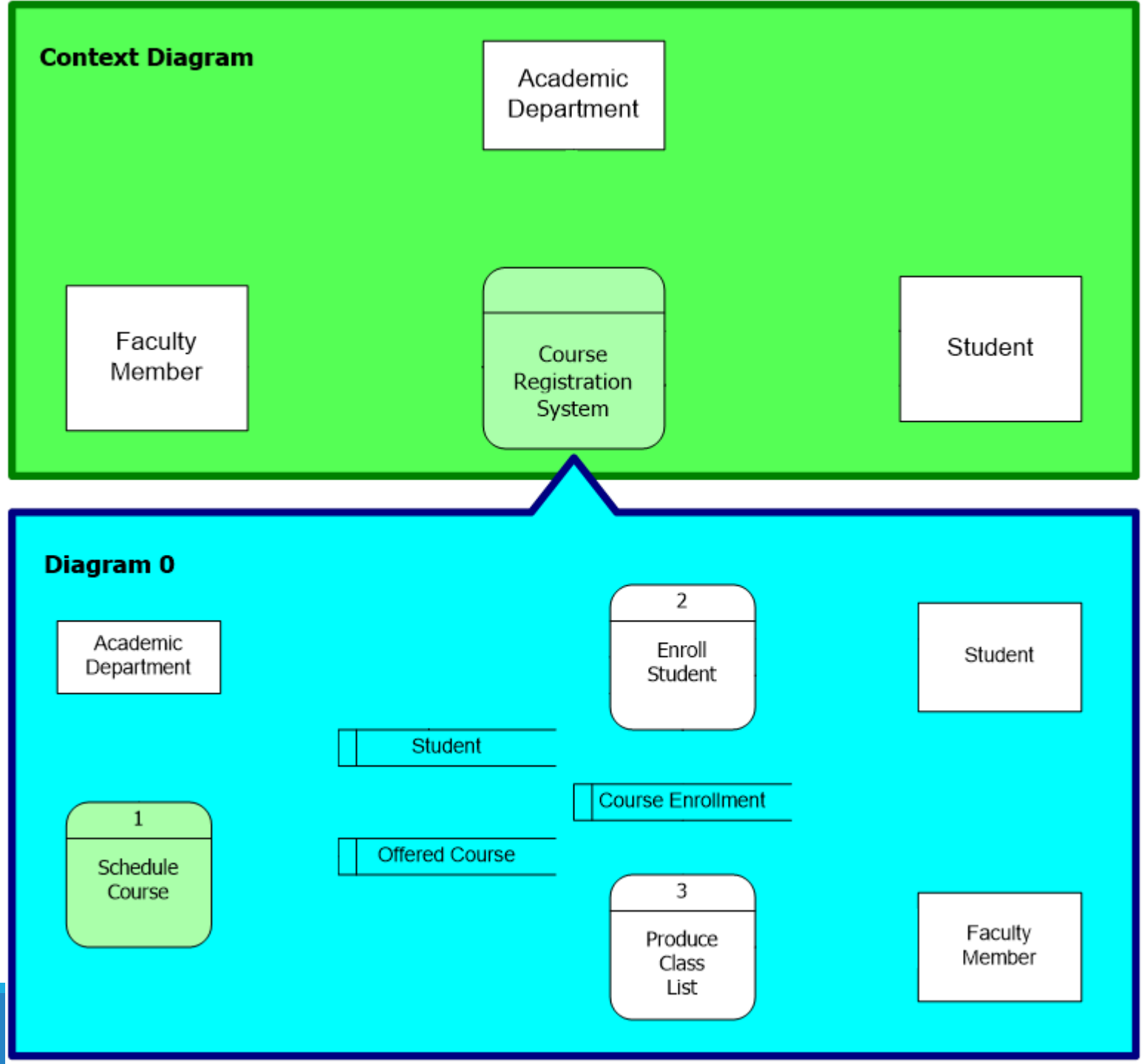


Event-Partitioned System Model (Diagram 0)

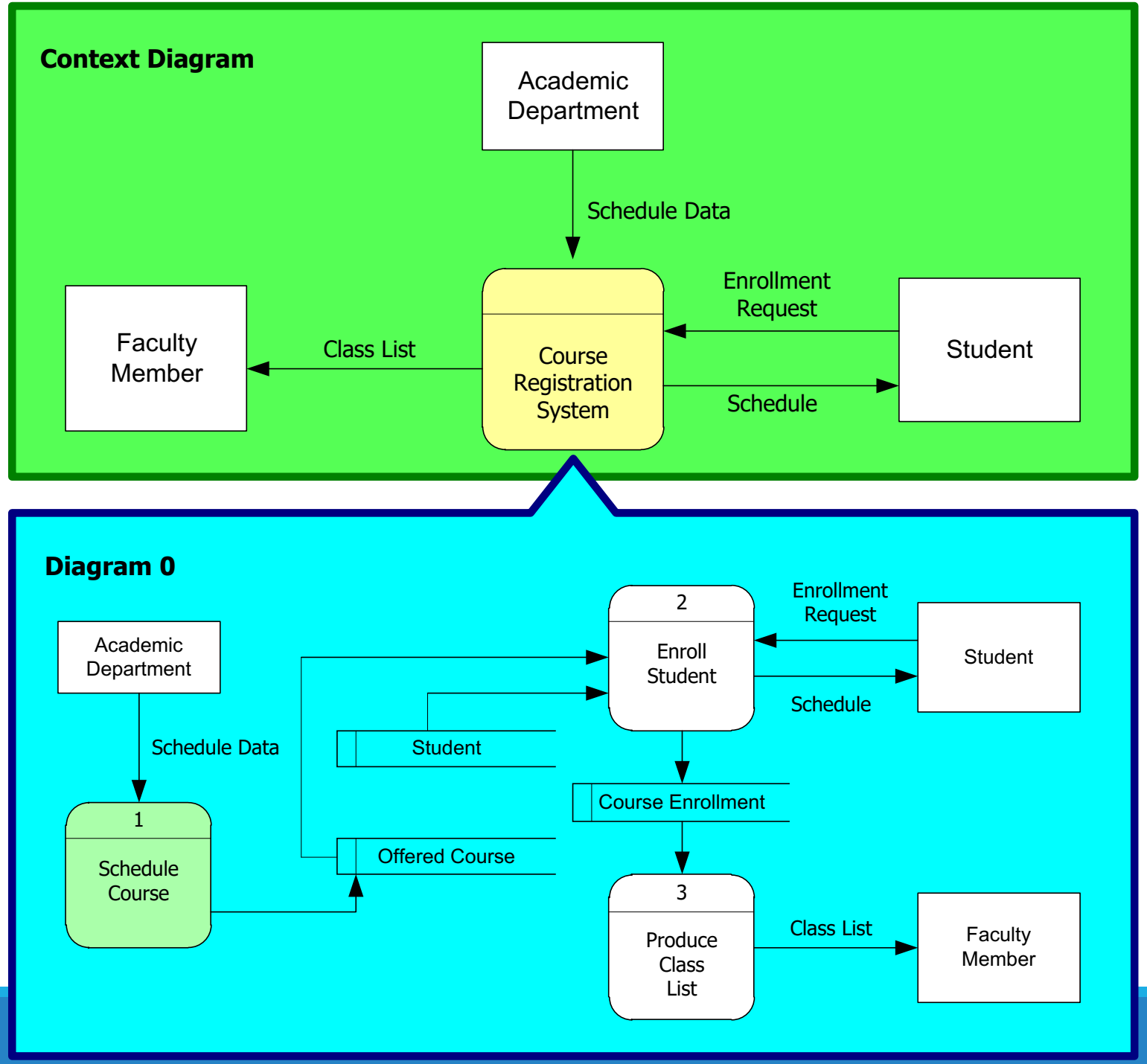
Event-Partitioned System Model

A DFD that models system requirements using a single process for each event in a system or subsystem

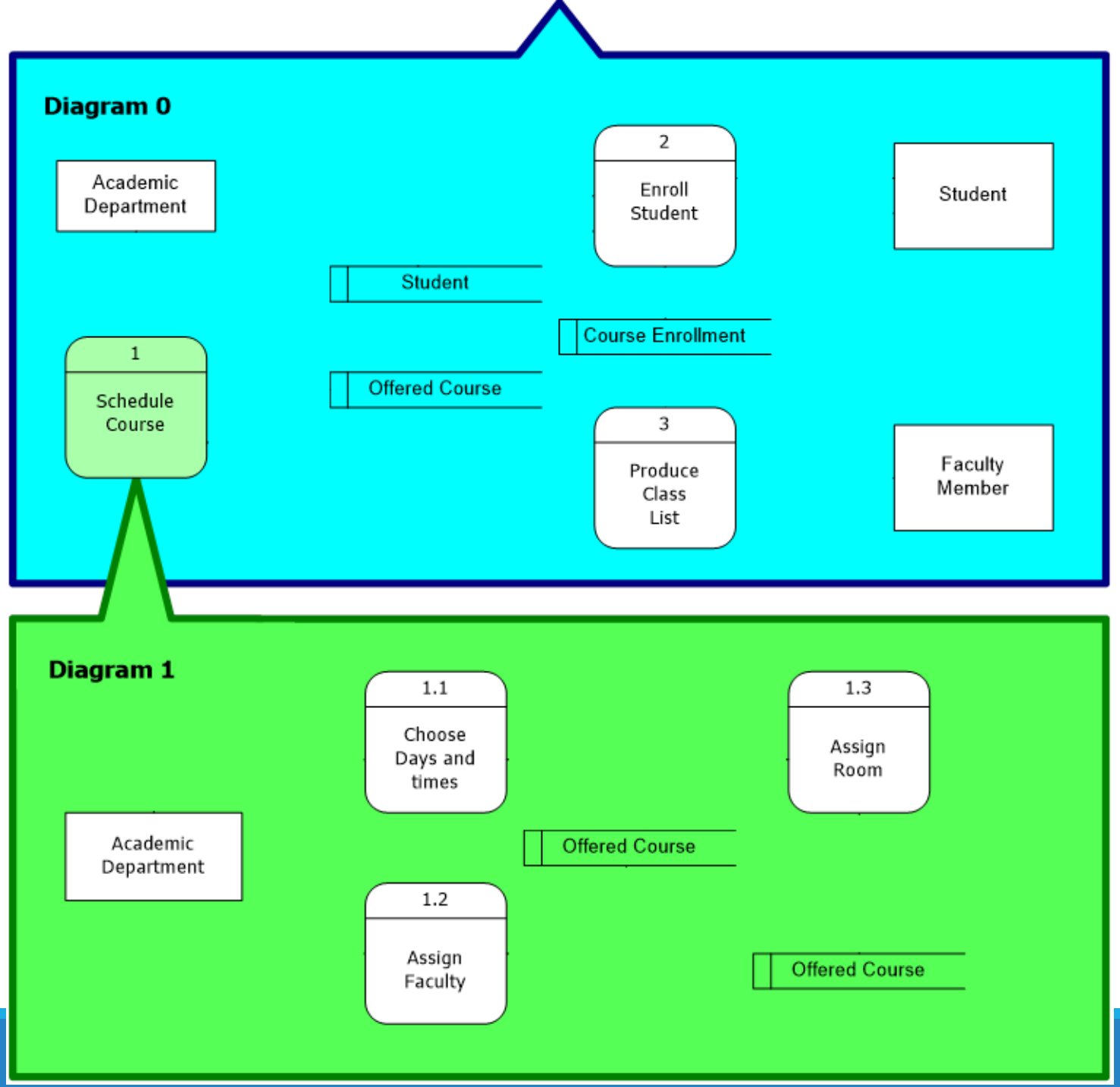
Example 4: DFDs



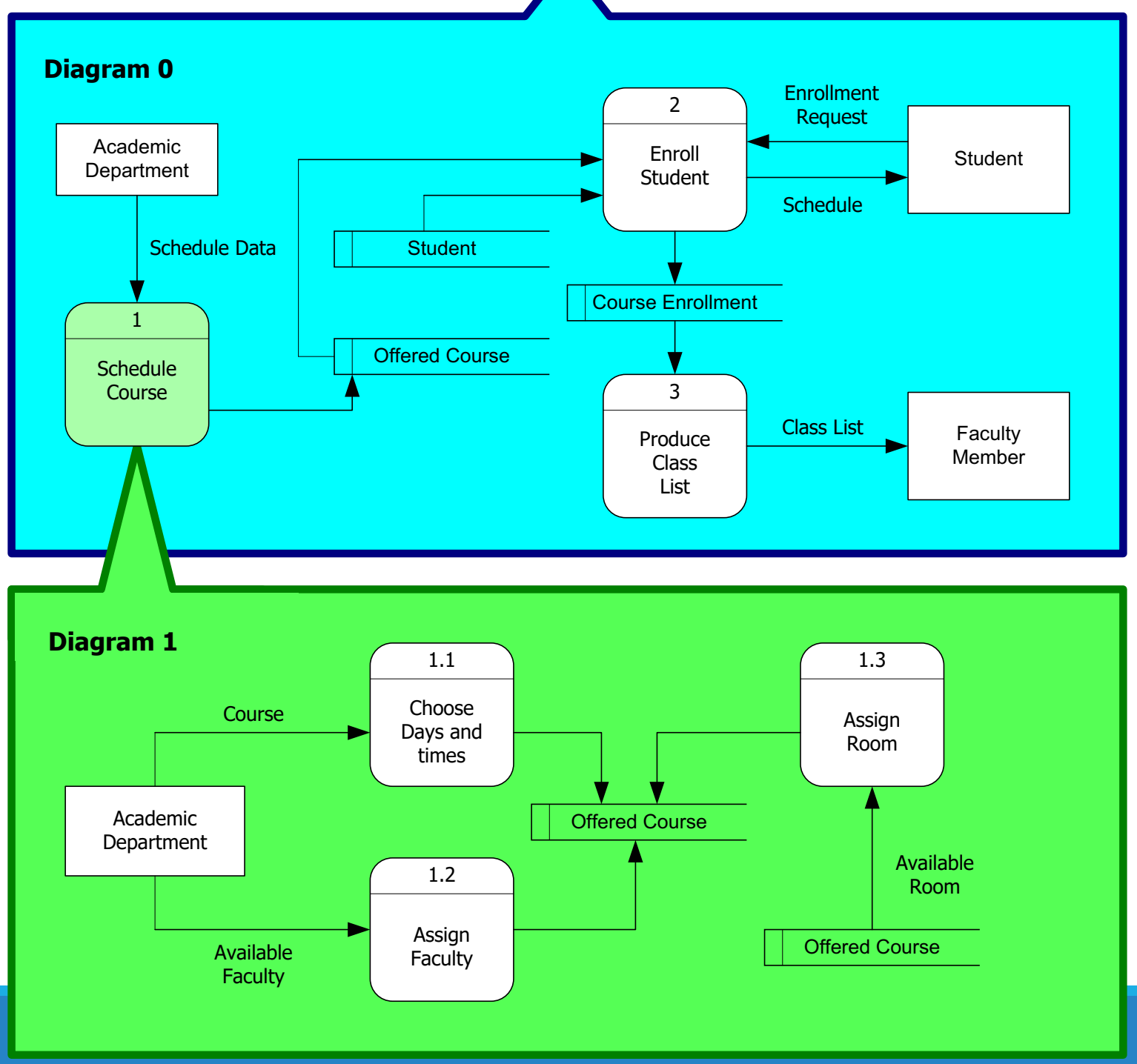
Example 4: DFDs



Example 4: DFDs (Cont.)

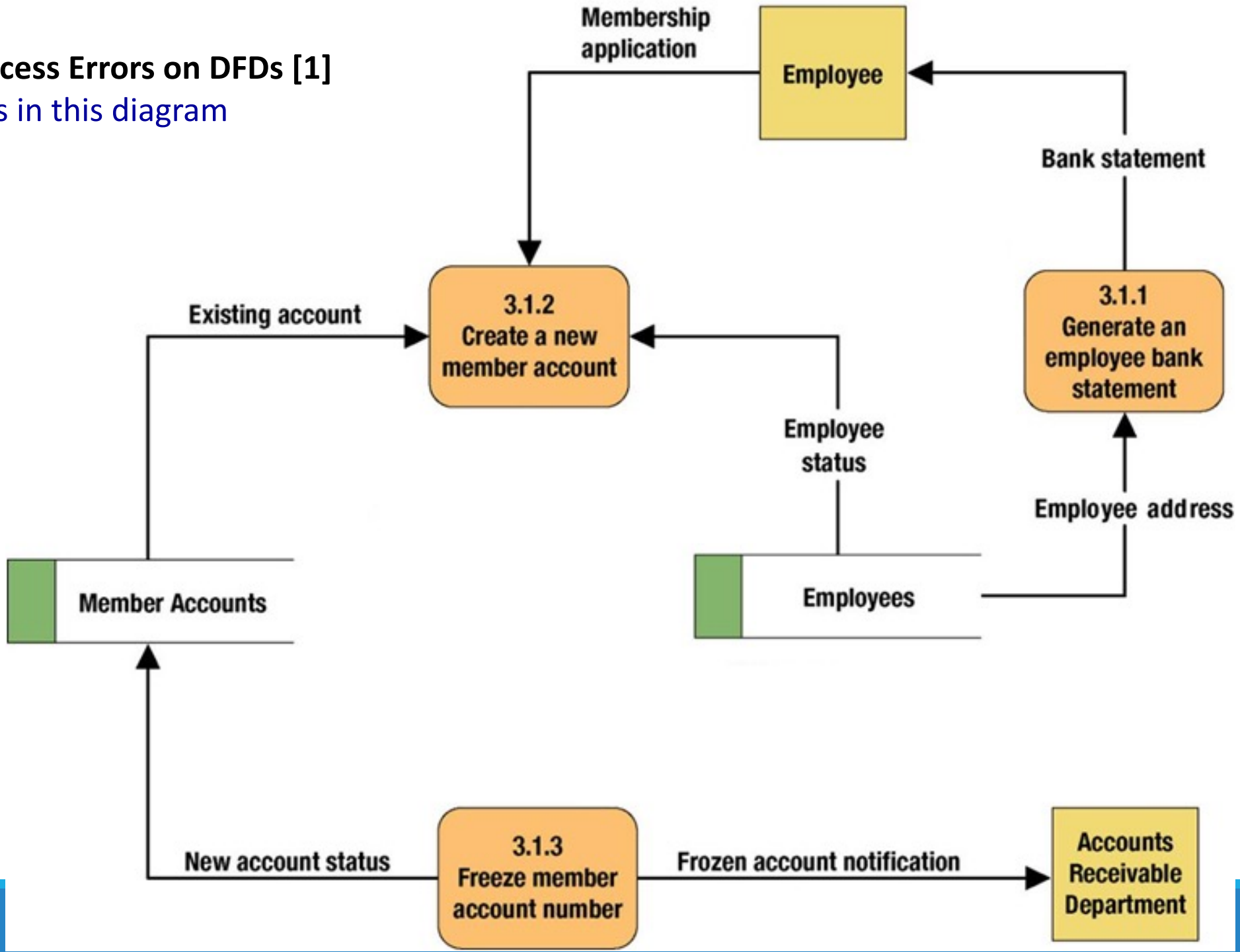


Example 4: DFDs (Cont.)



Common Process Errors on DFDs [1]

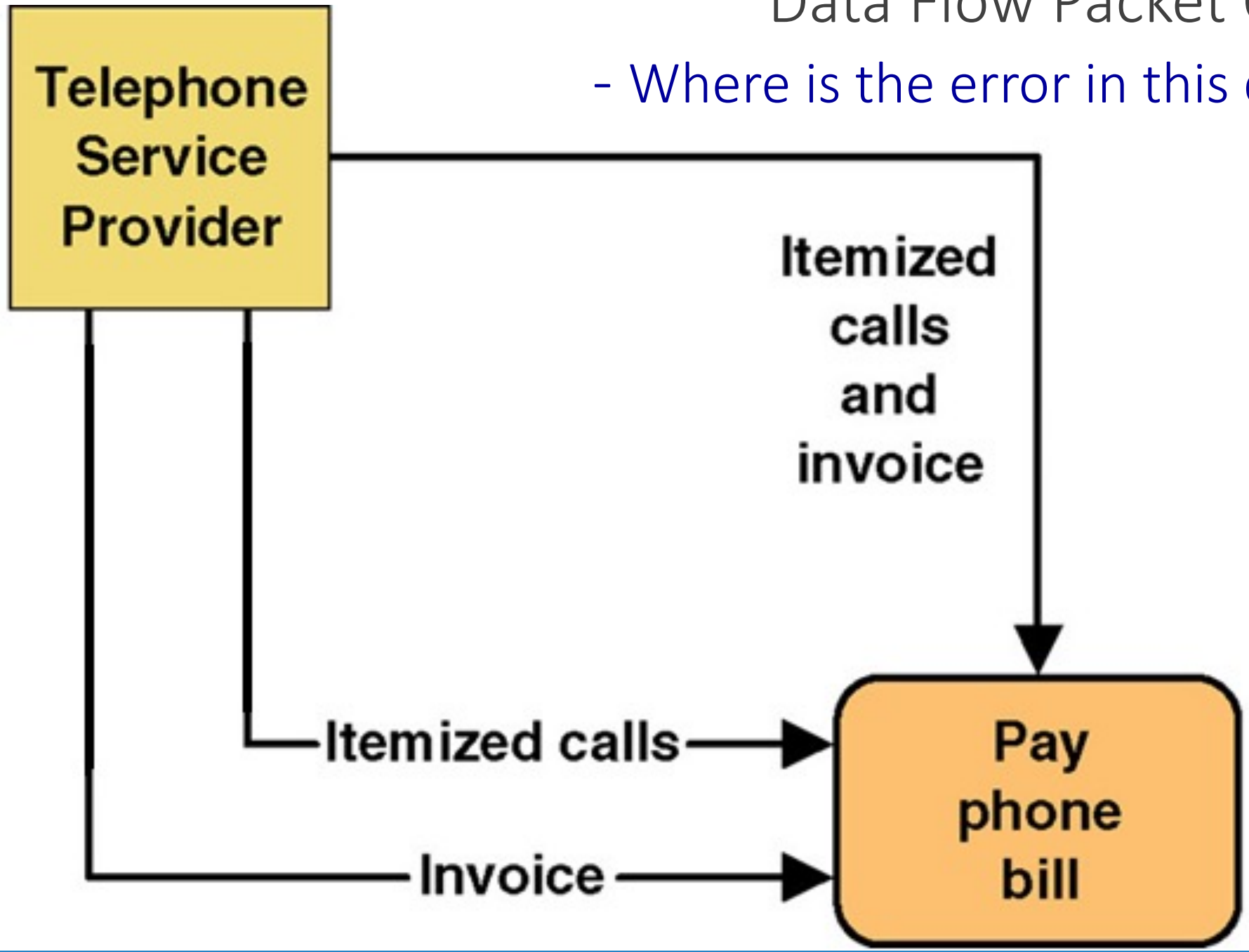
- Find 3 errors in this diagram



Common Process Errors on DFDs

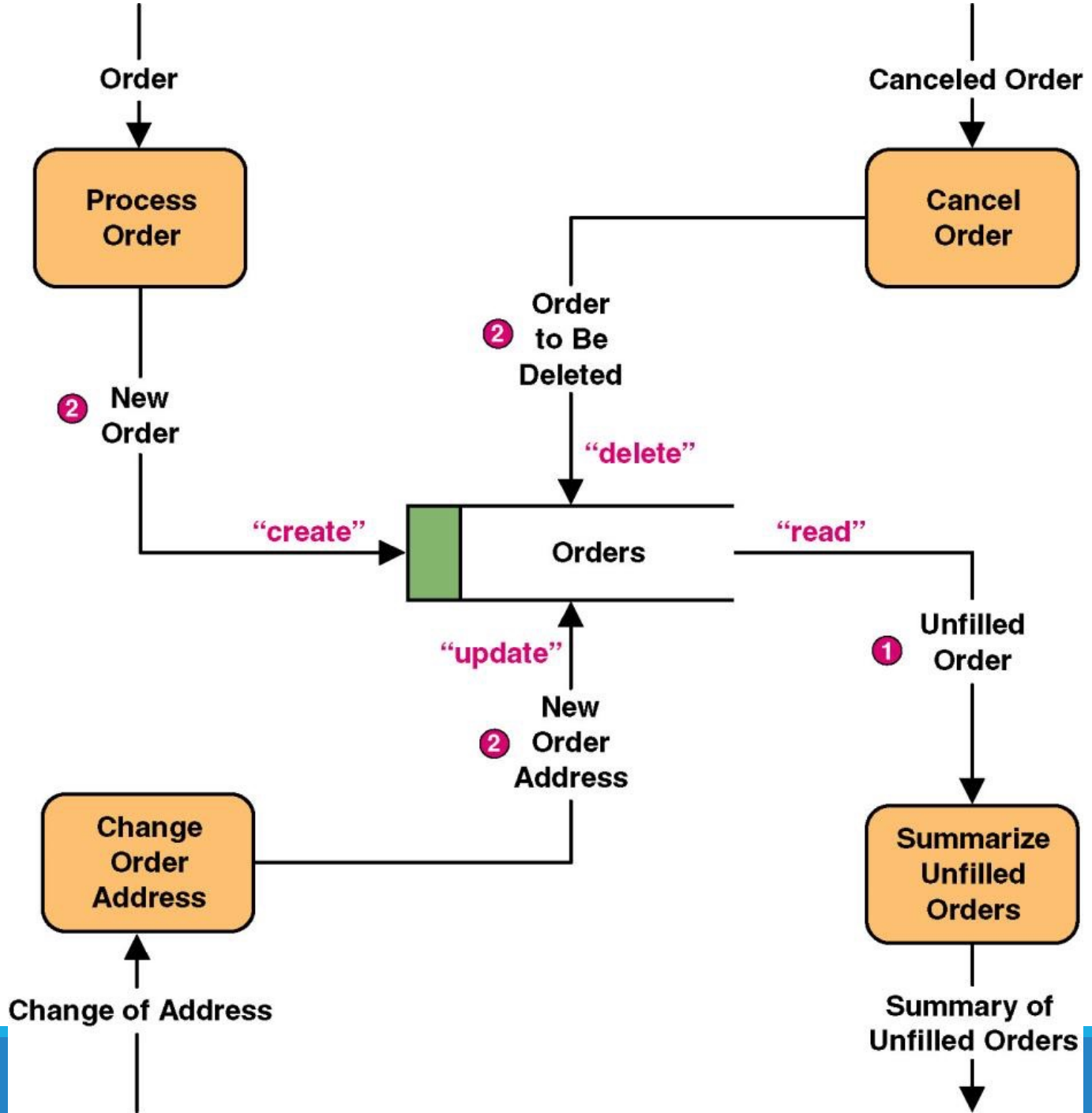
Data Flow Packet Concept

- Where is the error in this diagram?

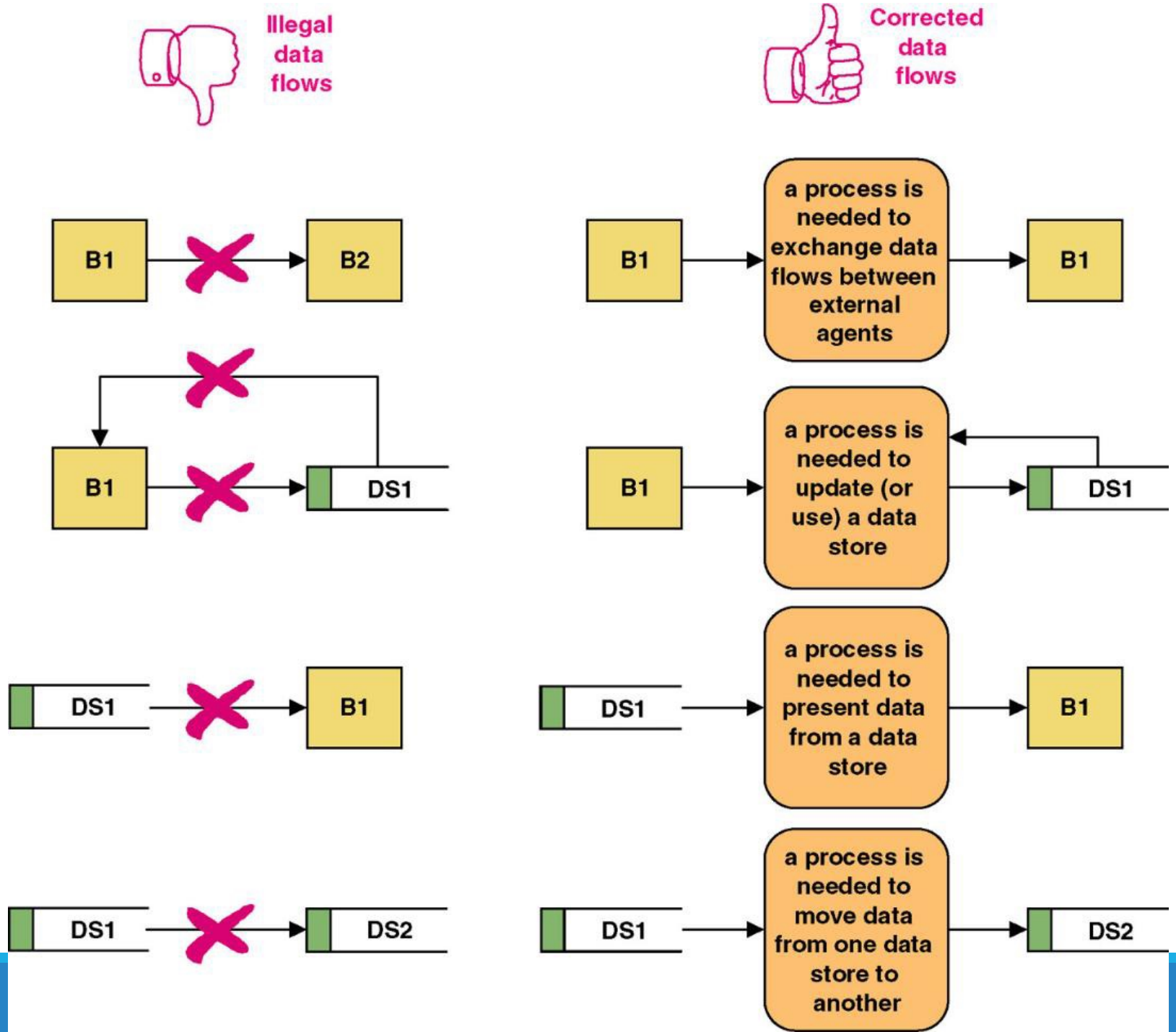


Data Flow Packet Concept

Data Flows to and from Data Stores [1]



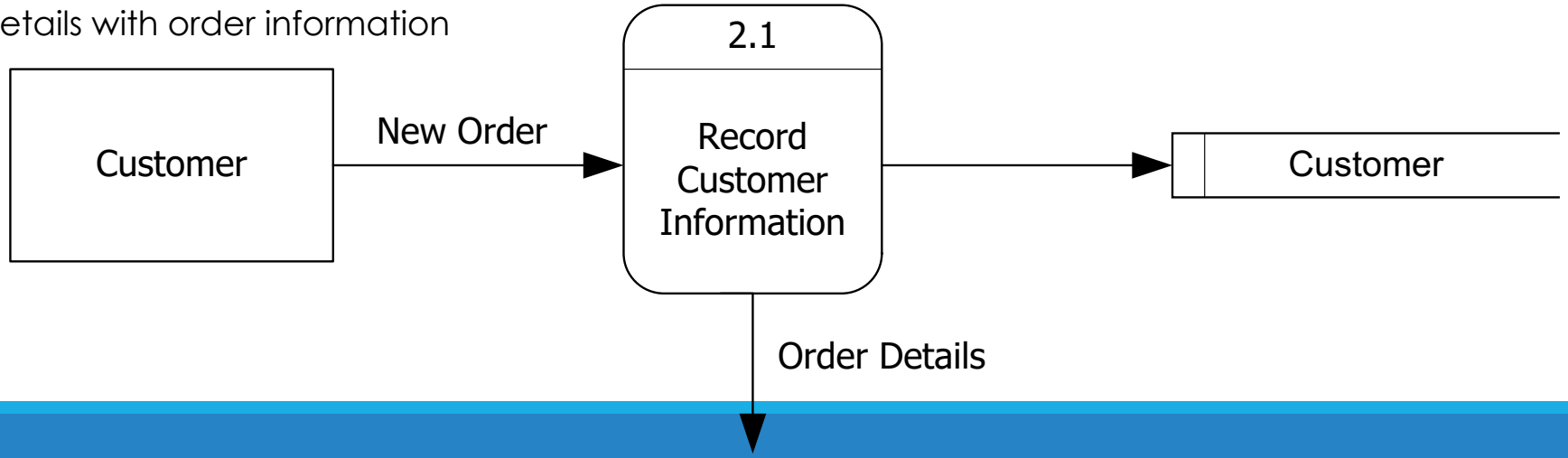
Illegal Data Flows [1]



Documentation of DFD Components

Process 2.1 – Record Customer Information

Ask if customer has an account (or has made a previous order)
If customer has an account then
    Ask for identification information
    Query database with identifying information
    Copy query response data to order details
Else
    Create an empty customer record in the database
    Ask customer for customer attributes
    Update empty customer record with customer attributes
End if
Ask customer for order information for first item
While more order items Do
    Update order details with order information
End while



Documentation of DFD Components

Process 2.1 – Record Customer Information

Draw diagram here

Documentation of DFD Components

Process 2.1 – Record Customer Information

CRUD Matrix [1]

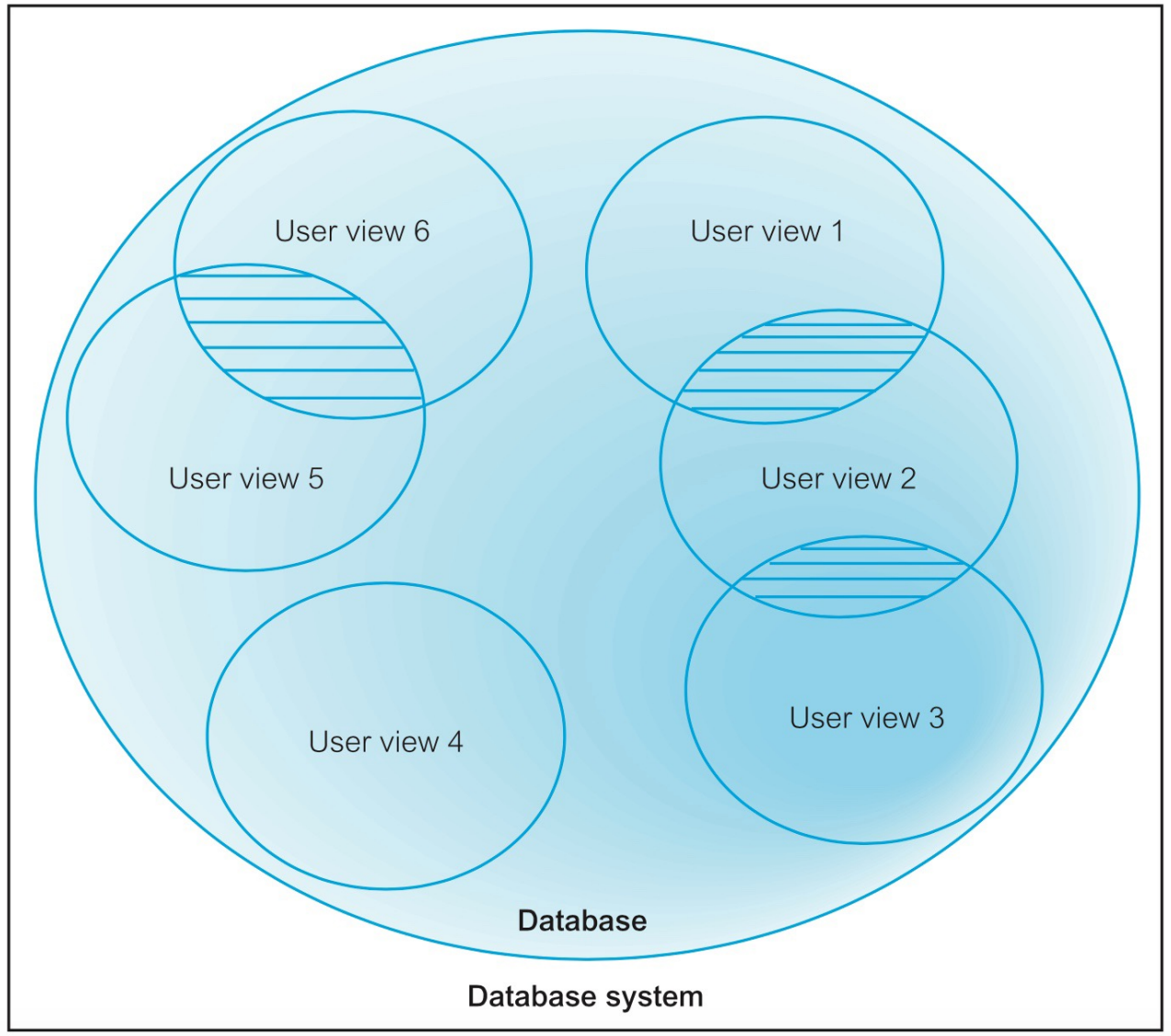
Acronym of CREATE, READ, UPDATE, and DELETE

Data-to-Process-CRUD Matrix

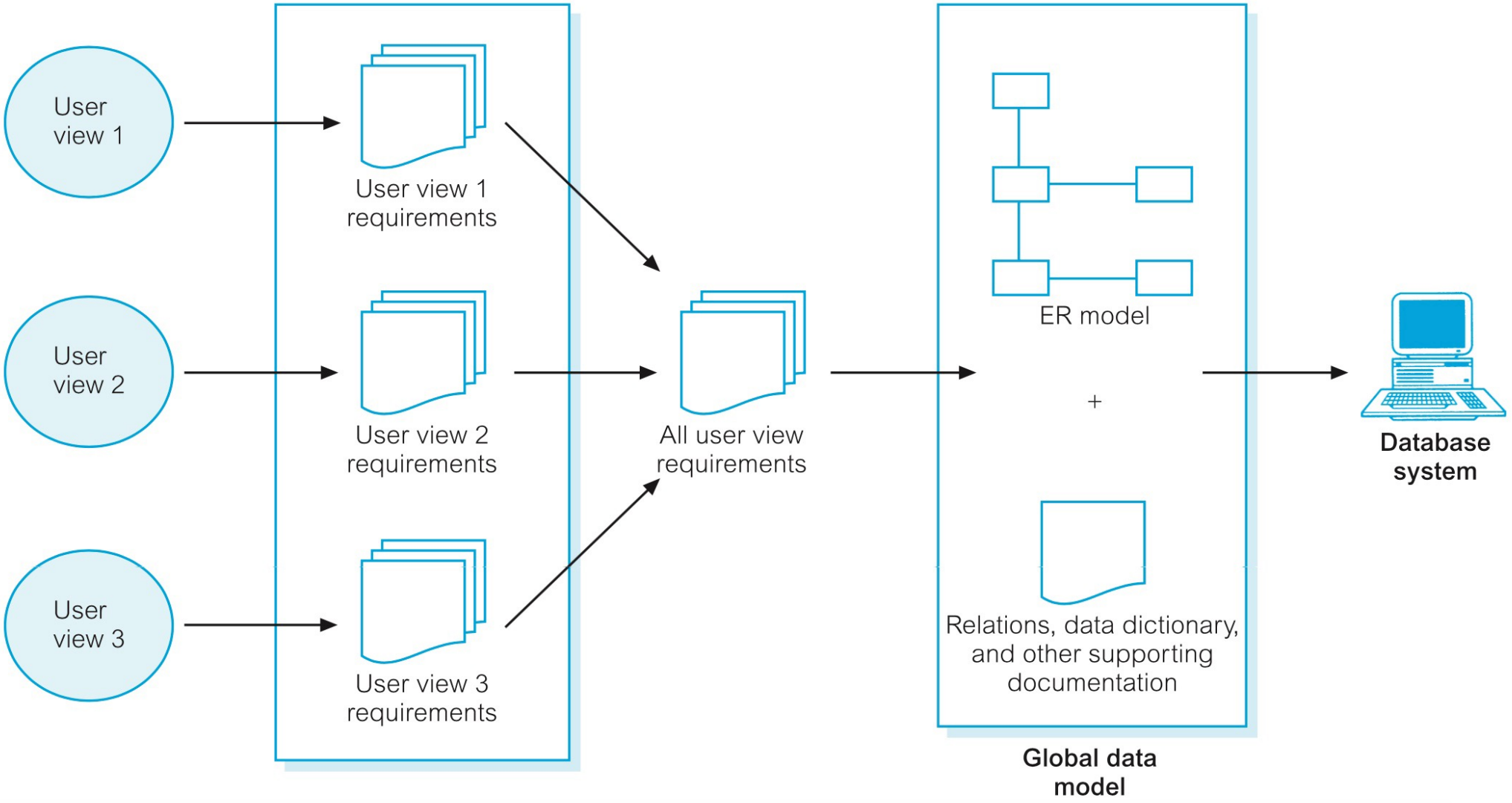
Entity . Attribute	Process Customer Application	Process Customer Credit Application	Process Customer Change of Address	Process Internal Customer Credit Change	Process New Customer Order	Process Customer Order Cancellation	Process Customer Change to Outstanding Order	Process Internal Change to Customer Order	Process New Product Addition	Process Product Withdrawal from Market	Process Product Price Change	Process Change to Product Specification	Process Product Inventory Adjustment
Customer	C	C			R	R	R	R					
.Customer Number	C	C			R	R	R	R					
.Customer Name	C	C	U		R		R	R					
.Customer Address	C	C	U		RU		RU	RU					
.Customer Credit Rating		C		U	R		R	R					
.Customer Balance Due					RU	U	R	R					
Order					C	D	RU	RU					
.Order Number					C		R	R					
.Order Date					C		U	U					
.Order Amount					C		U	U					
Ordered Product					C	D	CRUD	CRUD		RU			
.Quantity Ordered					C		CRUD	CRUD					
.Ordered Item Unit Price					C		CRUD	CRUD					
Product					R	R	R	R	C	D	RU	RU	RU
.Product Number					R	R	R	R	C			R	
.Product Name					R		R	R	C			RU	
.Product Description					R		R	R	C			RU	
.Product Unit of Measure					R		R	R	C		RU	RU	
.Product Current Unit Price					R		R	R			U		
.Product Quantity on Hand					RU	U	RU	RU					RU

C = create R = read U = update D = delete

Representation of a Database System



Database Design approach



Reference

1. Ian Sommerville, Software Engineering 10th Edition, Pearson, April 2015

ANY QUESTIONS?

:O)

Thank you