CMPS411
Spring 2018

Lecture 3

Structured Systems Analysis and Design Method (SSADM)

Outlines

- Structured Systems analysis and design method
- Data Flow Diagram (DFD) symbols
  - Processes
  - External entities (sources and sinks)
  - Data Stores
  - Data Flows
- Rules
- Examples

Requirements Document

- The requirements document is the official statement of what is required of the system developers.
- Should include both a definition of user requirements and a specification of the system requirements.
- It is NOT a design document. As far as possible, it should set of WHAT the system should do rather than HOW it should do it

System Modeling

- System modeling is the process of developing abstract models of a system,
- Each model presents a different view or perspective of that system.
- System modeling represents a system using various graphical notations and tools
- Data Flow Diagram(DFD) and Unified Modeling Language (UML) are examples of modeling tool
- System modelling helps the analyst to understand the functionality of the system better
- Models are used to communicate with customers.

Techniques Used to Analyze a Problem

- Object Oriented Analysis and Design (OOAD)

  (You have already seen this in the last lecture (Lecture-3), use cases)

  - Use cases
  - Domain model
  - Design class diagram
- Event Based System
  - Shows the system's reaction to events
  - Hardware-software embedded systems
  - Appropriate for real time systems
- Formal Method
  - Mathematical based techniques for specification, verification of software requirements
  - Formal specification
  - Precise specification
  - Theorem proving, model checking are often used
  - Difficult to use –high level expertise is needed.
  - Example: VDM, Z notation, Alloy.
- Structured Systems Analysis and Design Method (SSADM)
  - Data Flow Diagram
  - Structured chart

Object Oriented Analysis and Design (OOAD)

- Scenario based technique is usually used to model various aspects of the software
- Real world concepts are often used
- Centered on the concept of the object/domain/concept
- Define objects that are responsible for themselves
- Grouped objects into classes
- Easy to understand and reuse
- No central command structure
- Flexible for modifiability.

In this course, we are using OOAD method.

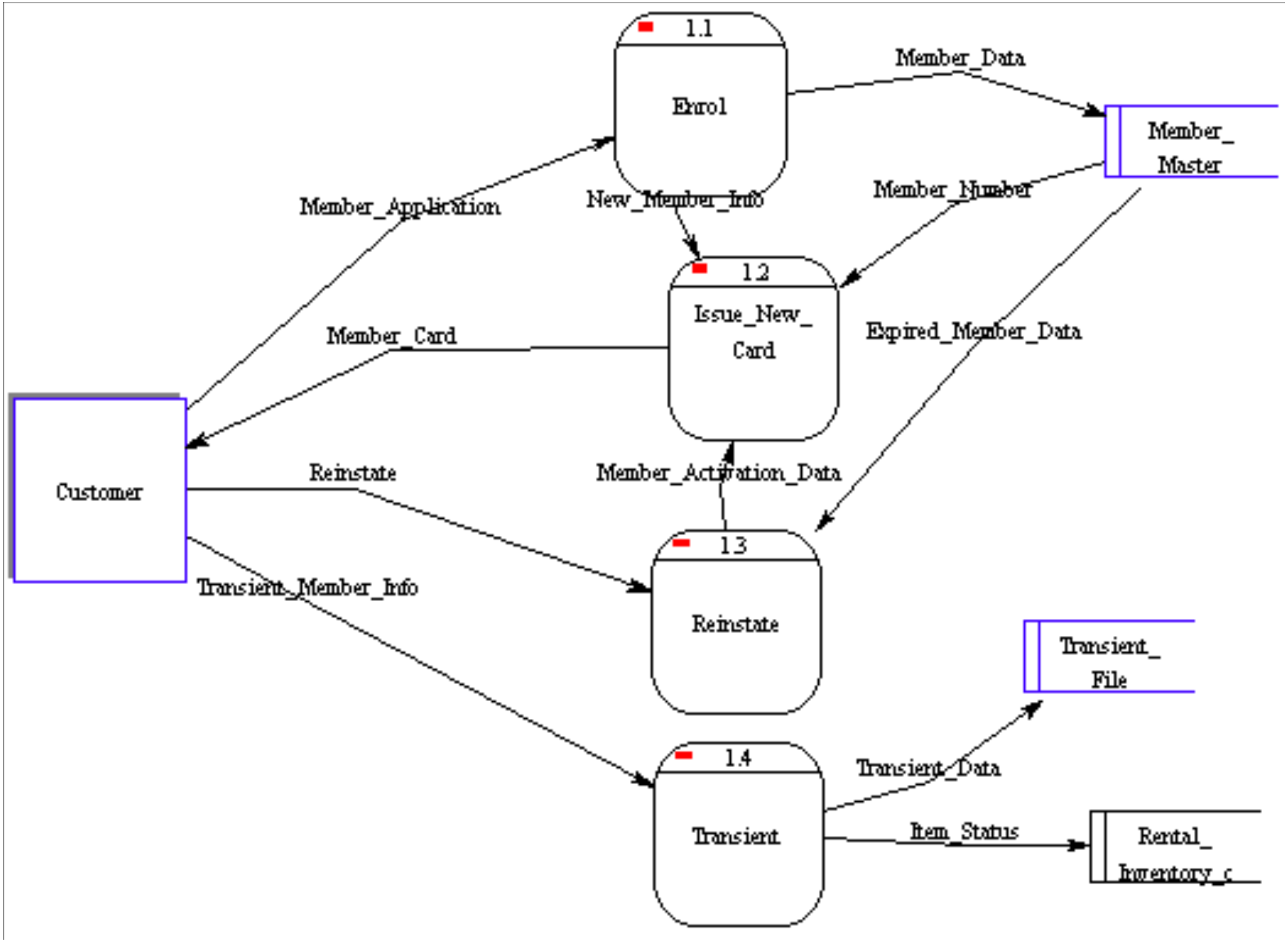
Structured Systems Analysis and Design Method (SSADM)

- Decompose the problem into smaller components
- This is also called **functional decomposition** because the problem is decomposed into the functions
- Focuses more on process and data
- More hierarchically structured
- Central command style module hierarchy
- Cannot handle changes properly
- Prototype based techniques is usually used to model the system
- Example, Data flow diagram, Structured Chart, etc.

What is a Data Flow Diagram?

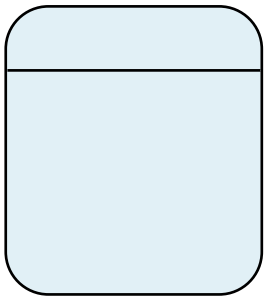
- A data flow diagram (DFD) is a graphical tool
- Allows system analysts (and system users) to depict the flow of data in an information system.
- The DFD is one of the methods that system analysts use to analyze information necessary to determine requirements
- A Data Flow Diagram is intended to serve as a communication tool among
  - systems analysts
  - end users
  - data base designers
  - system programmers
  - other members of the project team

A Sample DFD


1.1
Enrol
Member_Data
Member_
Master
Member_Application
New_Member_Info
Member_Number
1.2
Issue_New_
Card
Member_Card
Expired_Member_Data
Customer
Reinstate
Member_Activation_Data
1.3
Reinstate
Transient_Member_Info
Transient_
File
1.4
Transient_Data
Transient
Item_Status
Rental_
Inventory_c

DFD Symbols and Definitions

Process



- **Process** - performs some actions on data, such as computations, creates, modifies, stores, delete, etc. It cannot have any storage.

Data store

- **Data store** - information that is kept and accessed, may be in files, folders or a databases.

External Entity

- **External entity** - is the origin or destination of data. Entities are external to the system.

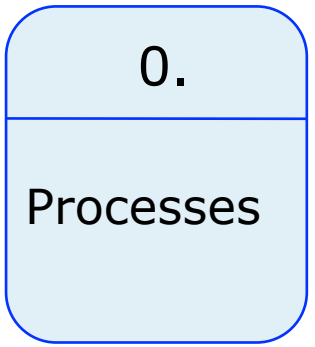
Data flow

- **Data flow** - the flow of data into or out of a process, data store or entity

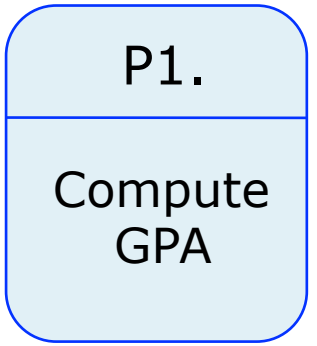
Processes

- A system activity/task
- Process transforms data
- Always internal to system
- Rules:
  - Data stays at rest unless moved by a process.
  - Processes cannot consume or create data, but it can transform data into another
  - Must have at least one input data flow
  - Must have at least one output data flow (to avoid black holes)
  - Should have sufficient inputs to create outputs
  - Only process can communicate with the data store, external entities and other processes.

Symbol:



Example:



Processes

- Valid processes include those that:
  - Perform computations (e.g., calculate grade point average)
  - Make decisions (determine availability of ordered products)
  - Sort, filter or otherwise summarize data (identify overdue invoices)
  - Organize data into useful information (e.g., generate a report or answer a question)
  - Trigger other processes (e.g., turn on the furnace or instruct a robot)
  - Use stored data (create, read, update or delete a record)
  - Create output (e.g., update database, create booking, etc)

External Entities

- External people, external systems and external data stores, machines,
- External entities reside outside the system, but interact with system, either
  - receive info from the system,
  - trigger the system into motion, or
  - provide information to the system
    - e.g. Customers, managers

Symbol:

External entity

Example:

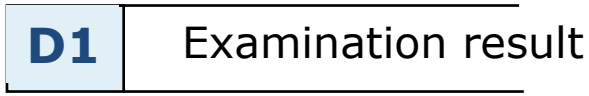
Student

Data Stores

- Internal to the system
- Data at rest
- Basic operations of data stores
  - Store, Add, Delete, Update, Search, etc
- Rules:
  - A data store cannot communicate directly with another data store
  - A data store cannot communicate directly with external entity
- Data stores can come in many forms:
  - Hanging file folders
  - Computer-based files
  - Notebooks
  - Databases
  - Data files, etc

Symbol:

Example:



Data Flows

- Data in motion, moving from one place to another in the system
  - From external entity (source) to the system
  - From the system to the external entity (sink)
  - From internal symbol to internal symbol, but always either start or end at a process within the system

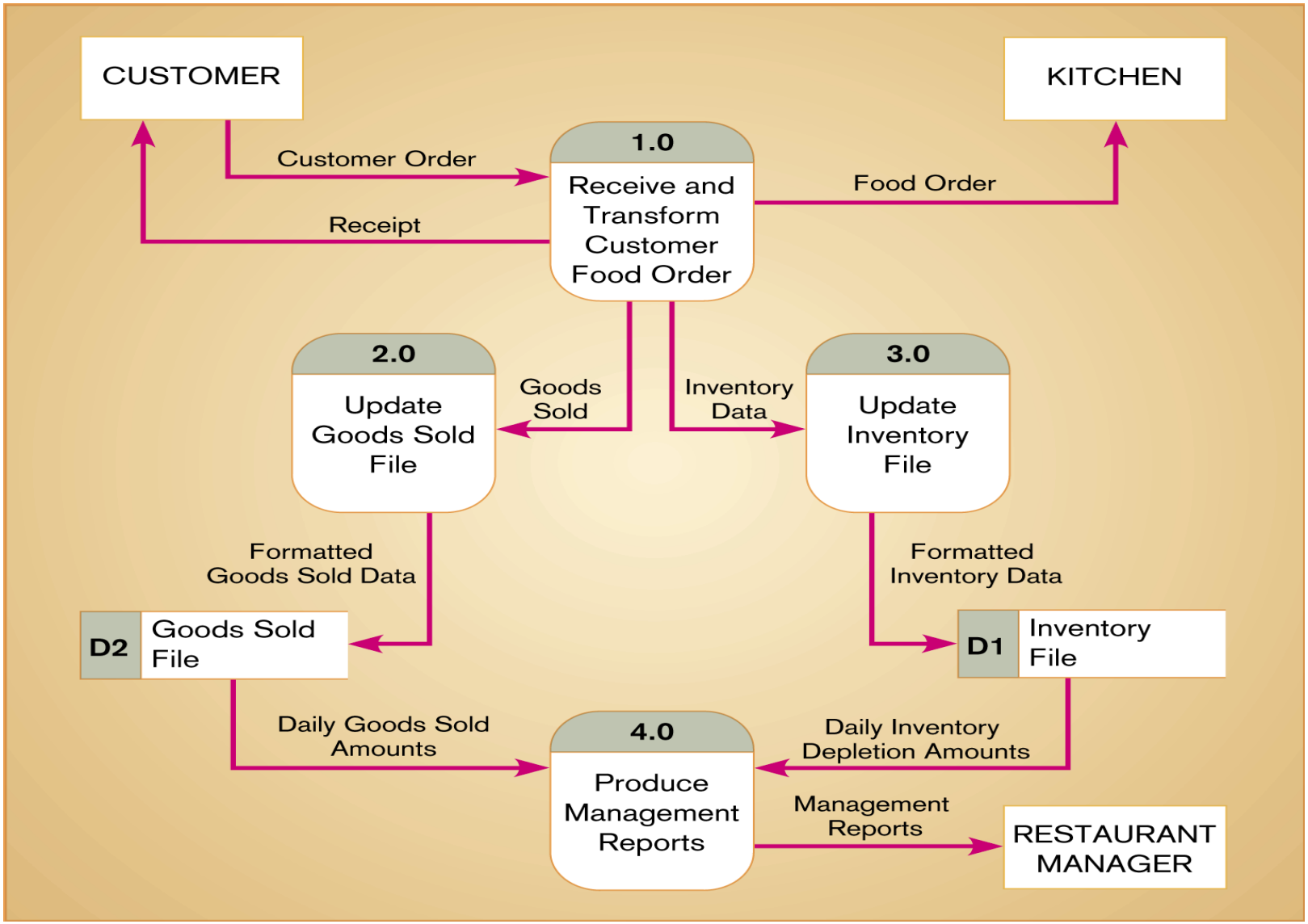
Symbol:

Data Flow 

Example:

Student id, course code 

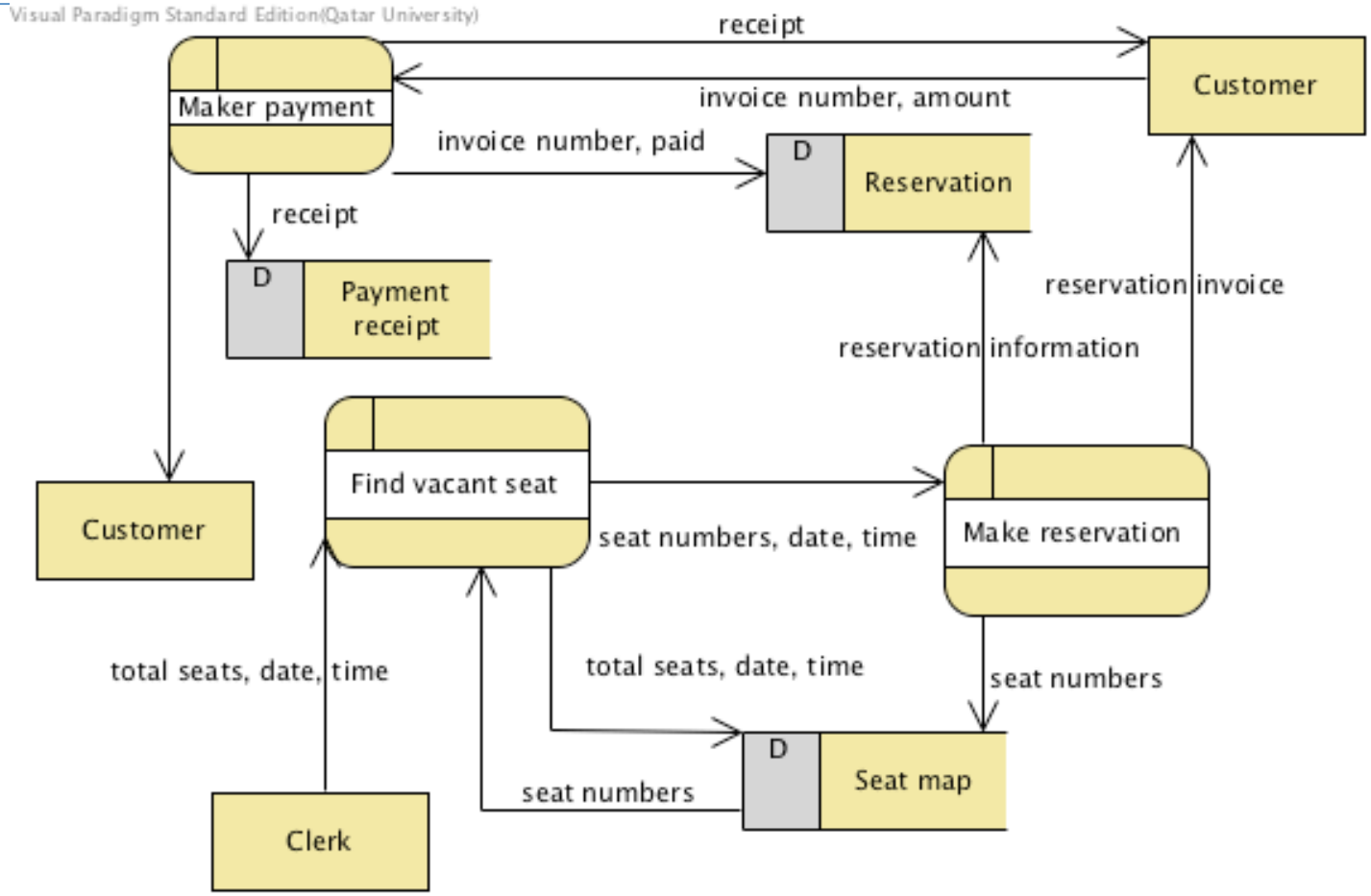
An Example: Food Ordering System



Another Example: Data Flow Diagram (DFD)

The requirements of a theatre reservation system:

The clerk enters the date and time of the show along with the number of seats that the customer wants to reserve for a show. The system first checks if the seats are available. If available, the system makes the reservation and update the seat of that day and time. The customer is provided with the confirmed reservation and the reservation is saved. The customer can make payment for the reservation, and gets a receipt of the payment from the system.



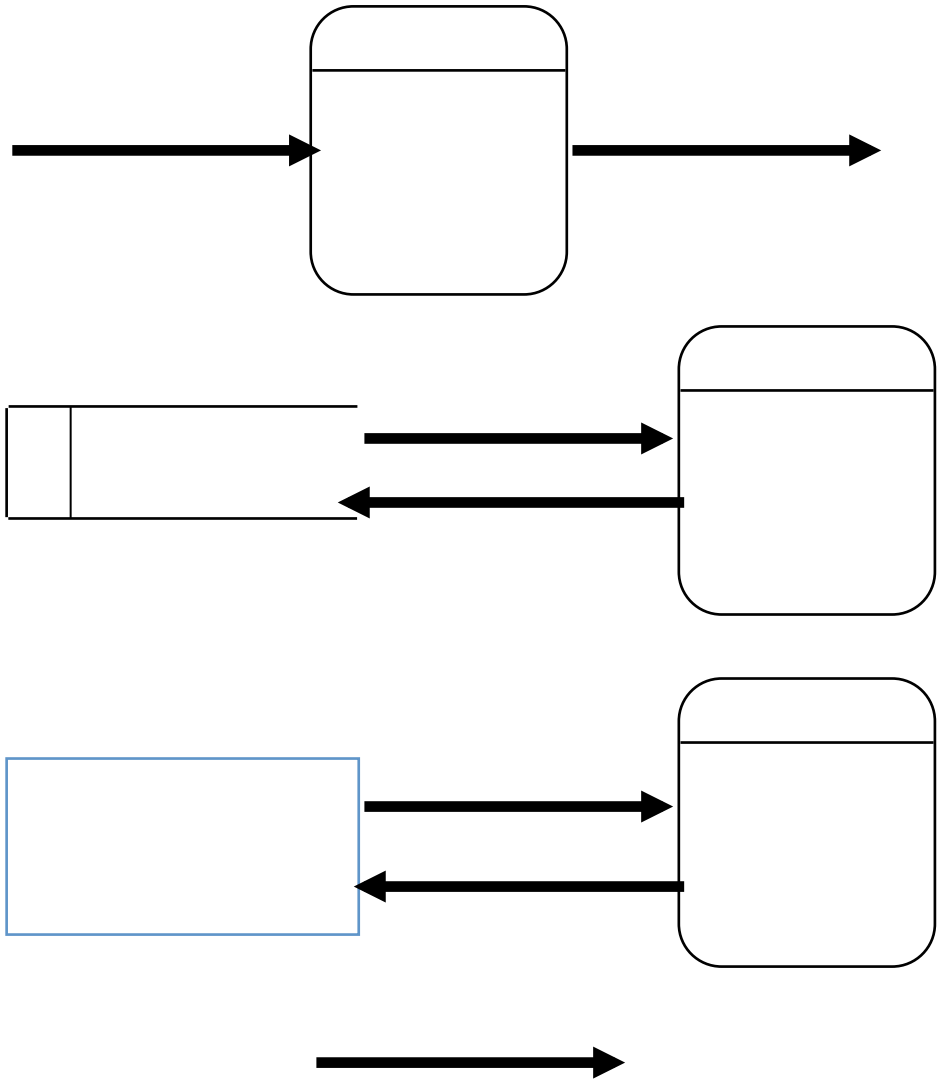
DFD Rules Revisited—
Process and data Stores

- Process
  - No process can have only outputs (a miracle)
  - No process can have only inputs (black hole)
  - A process must have a **verb phrase label**
- Data Store
  - Data cannot be moved directly from one store to another
  - Data cannot move directly from an external entity to a data store
  - Data cannot move directly from a data store to an external entity
  - Data store must have a **noun phrase label**

DFD Rules Revisited—
External Entities and Data Flows

- Source/Sink (external entities)
  - Data cannot move directly from one external entity to another external entity
  - External entities must have **noun phrase label**
- Data Flows
  - A data flow cannot go directly back to the same process it leaves
  - A data flow to a data store means update
  - A data flow from a data store means retrieve or use
  - Must have **noun phrase label**

Rules for Drawing DFD



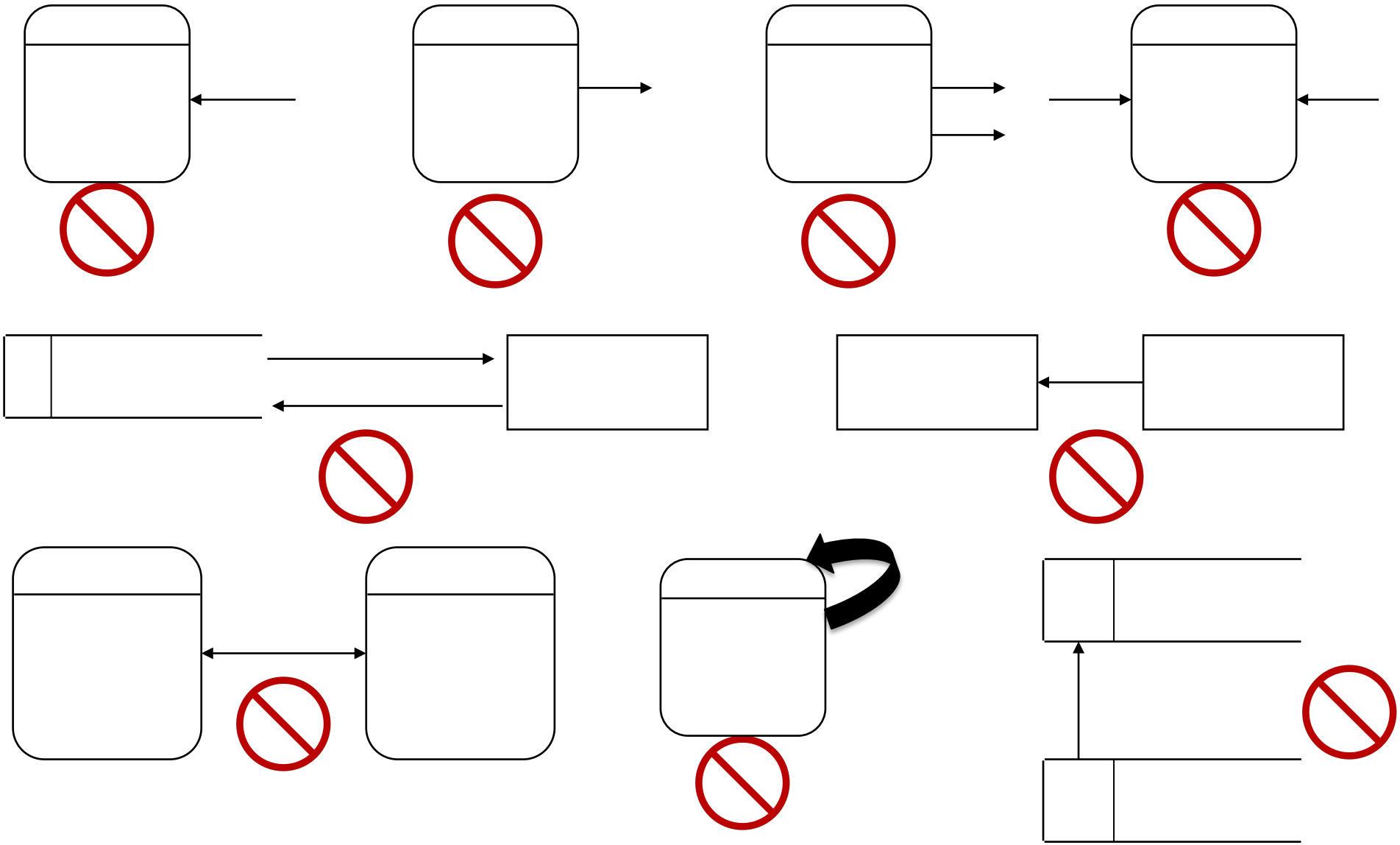
A **minimum** of one data flow in and one data flow out of a process

A datastore **must** be connected to a process (either in, out, or both)

An external entity **must** be connected to a process (either in, out, or both)

A single data flow **must** only flow one way

DFD: Common Mistakes



DFD Rules—General

- Basic rules that apply to all DFDs
  - Inputs to a process are always different than outputs
  - Objects always have a unique name
  - In order to keep the diagram uncluttered, you can repeat data stores and external entities on a diagram, but not the process

Reading resource

- https://www.smartdraw.com/data-flow-diagram/
- https://www.visual-paradigm.com/tutorials/data-flow-diagram-dfd.jsp
- https://www.visual-paradigm.com/guide/data-flow-diagram/what-is-data-flow-diagram/