

King Saud University

College of Engineering

IE – 462: “Industrial Information Systems”

Fall – 2023 (1<sup>st</sup> Sem. 1445H)

[Chapter 4:](#)

***Structured Analysis and Functional  
Architecture Design – p2 – DFD – i - Fundamentals***

Prepared by: Ahmed M. El-Sherbeeny, PhD

# Lesson Overview

- Modeling IIS – (p1)
- Integrated Computer-Aided Manufacturing Definition 0 (IDEF0) – (p1)
- **Data Flow Diagram (DFD) – (p2)**
  - i. **Fundamentals**
  - ii. Diagramming Rules
  - iii. Case Studies

# DFD – part i – Fundamentals

- [Introduction to DFD](#)
- [DFA/DFD Modeling Primitives](#)

# Functional/Process Modeling:

## 2. Data Flow Diagram (DFD)

# Introduction to DFD



# Layers of IS Design Process

## Conceptual Layer

Functional  
Architecture

Informational  
Architecture

## Implementation Layer

- Software Environment
- Platform Hardware
- Network Architecture

## Execution Layer

- Database Management System
- Forms and Reports

# Functional/Process Modeling

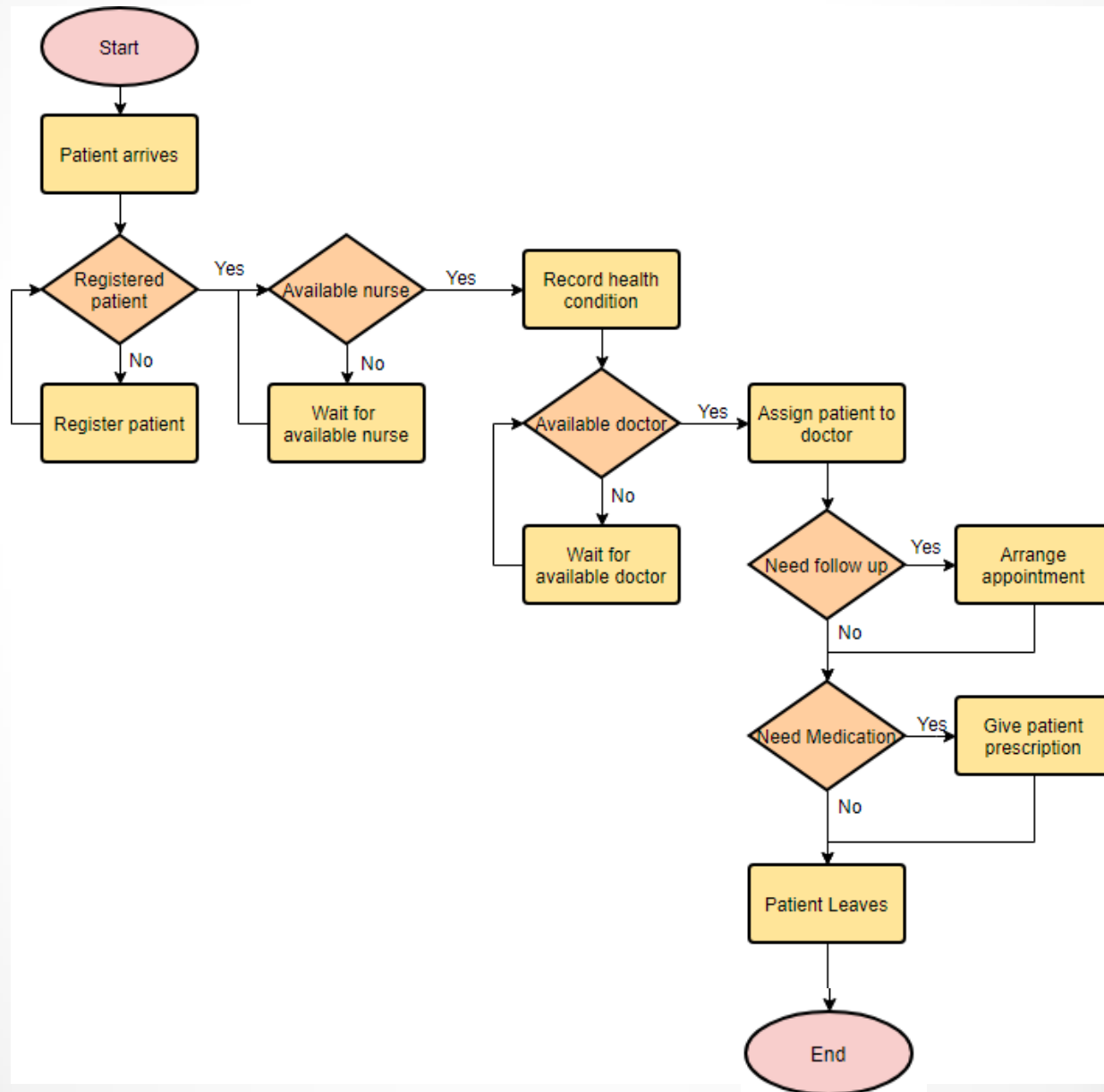
- *Remember*: two methodologies for designing a functional architecture, known as “structured analysis” techniques:
  - **data flow diagrams** (1979), widely used by information system professionals in all industries
  - **structured analysis and design technique** (SADT) (1988), adapted for manufacturing enterprises under the name **integrated computer-aided manufacturing definition 0** (IDEF0)
  - Both methodologies are based on graphical notations used to describe information flows among processes of the enterprise being documented

# Functional/Process Modeling

- Graphically represent the processes that capture, manipulate, store, and distribute data:
  - between a system and its environment
  - and among system components
- Useful for depicting purely *logical* information flows
- DFDs differ from system flowcharts which depict a *procedure* ([see next slide](#))



# System Flowchart (example)



# DFDs vs. IDEF0

- **Data Flow Diagrams (DFD)**, aka **Data Flow Analysis (DFA)** is an alternative to IDEF0 that is widely used in all industries, both in modeling:
  - manufacturing, and
  - service processes and operations
- It differs from IDEF0 in that it focuses exclusively on business *processes* and the *information that flows* among processes, *ignoring* (unlike IDEF0):
  - material flows
  - mechanisms, and
  - controls

# DFA/DFD Modeling Primitives



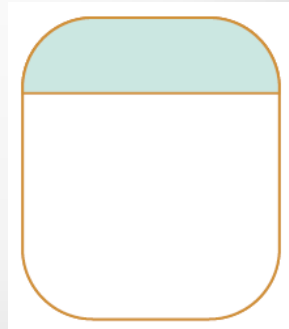
# Components of DFD

- Data flow diagrams are constructed using four symbols:
  - **Process**
  - **Data Flow**
  - **Data Store**
  - **Sources and Sinks**

# Components of DFD (cont.)

- **Process:**

- work or actions performed on data (inside the system) so that they are transformed, stored, or distributed
- represents people/procedures that transform data
- each process *must* have data entering it and exiting it (otherwise, it does not belong in a DFD)
- [Gane and Sarson symbol:](#)
  - upper portion is used to indicate the number of the process
  - lower portion is a name for the process (e.g. Generate Paycheck, Calculate Overtime Pay)



# Components of DFD (cont.)

- **Data flow:**

- arrows indicate the direction in which the data move (i.e. “*data in motion*,” from one place in a system to another)
- “data” is a general concept; e.g. data sent to a computer file, or information given from one process to another process
- remember, arrows are *not* used to indicate physical flow of materials (as in IDEF0)
- arrow is labeled with a meaningful name for data in motion (e.g. Customer Order, Sales Receipt, or Paycheck)



# Components of DFD (cont.)

- **Data store:**

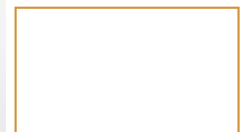
- place where data are preserved as a record inside the system (“*data at rest*”)
- e.g. computer file or paper filing cabinet
- note, there is *no* explicit construct in IDEF0 that is analogous to a data store
- [Gane and Sarson symbol:](#)
  - left end: a small box used to number the data store
  - inside the main part of the rectangle is a meaningful label for the data store (e.g. Student File, Transcripts, or Roster of Classes)



# Components of DFD (cont.)

- **Sources and sinks:**

- external entity that is origin –source– or destination –sink– of data (outside the system)
- it represents how at the boundaries, DFA system interacts with outside people, processes, organizations, other information systems (note, this is similar to IDEF0 model)
- **Sources:** entities outside the system that provide data input to the system (usually trigger events in the system); e.g. customer
- **Sinks:** entities outside the system that receive data
- note, same entity may be both a source and a sink if it both sends data to and receives data from the system
- [sources/sinks](#) have a name that states what the external agent is (e.g. Customer, Teller, Inventory Control System)



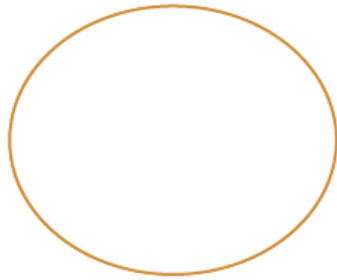


# Components of DFD (cont.)

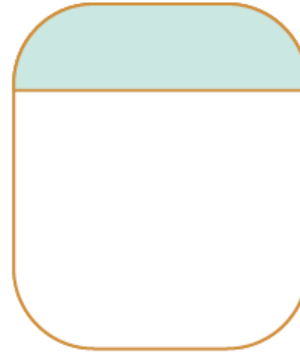
## Sources and sinks (cont.):

- Sources and sinks, *do not* consider the following:
  - Interactions that occur between sources and sinks
  - What a source or sink does with information (i.e. source or sink is a “black box”)
  - How to control or redesign a source or sink (assumed to be fixed)
  - How to provide sources and sinks direct access to stored data (since they cannot directly access or manipulate data stored within the system)

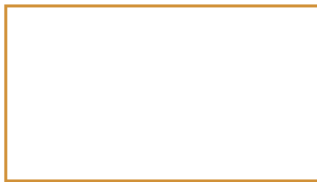
# DFD Symbols/Notation



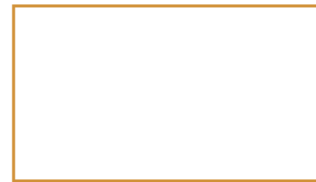
process



data store



source/sink



data flow

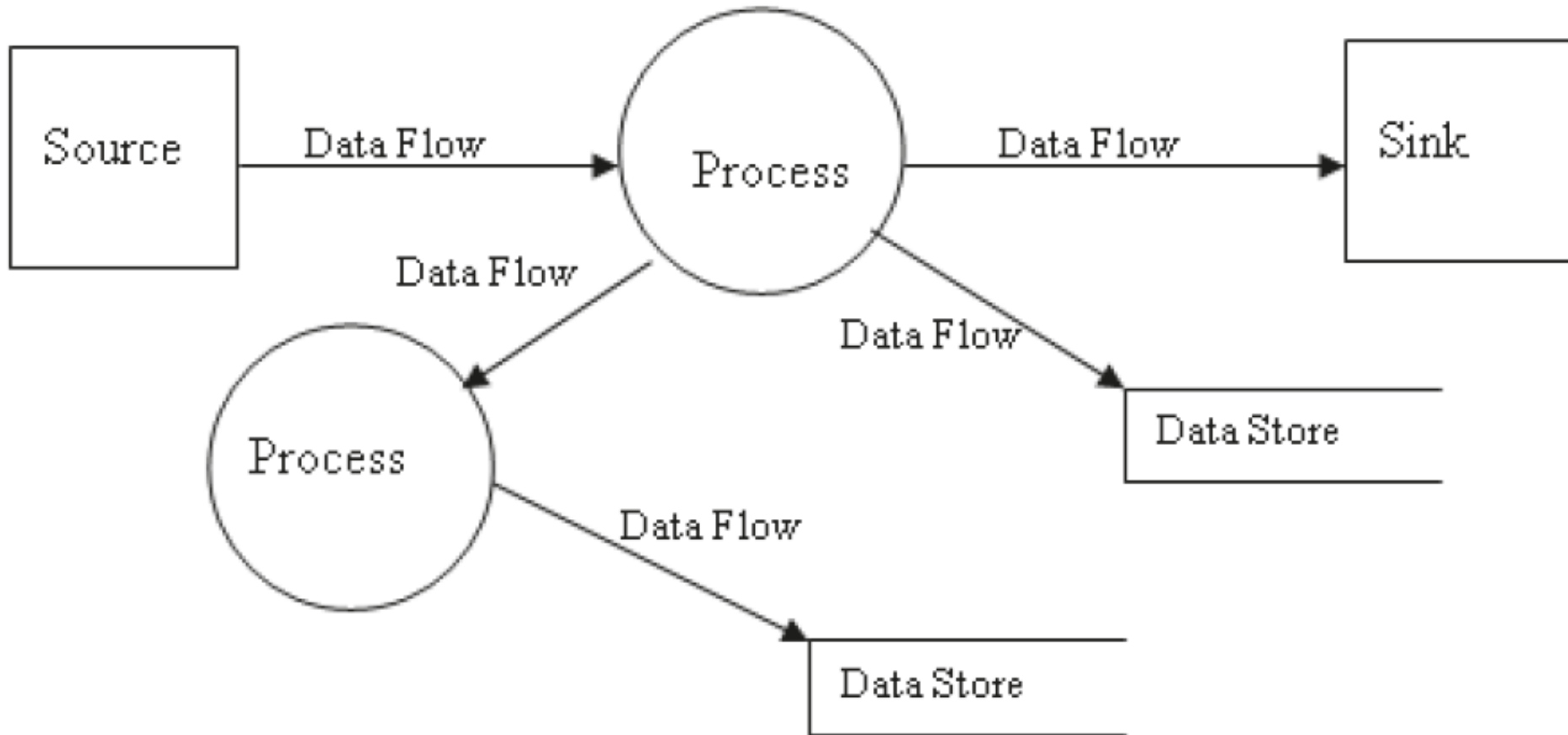


DeMarco and Yourdon  
symbols

Gane and Sarson  
symbols

**FIGURE 7-2**  
Comparison of DeMarco and Yourdon  
and Gane and Sarson DFD symbol sets

# DFD Symbols/Notation (contd.)



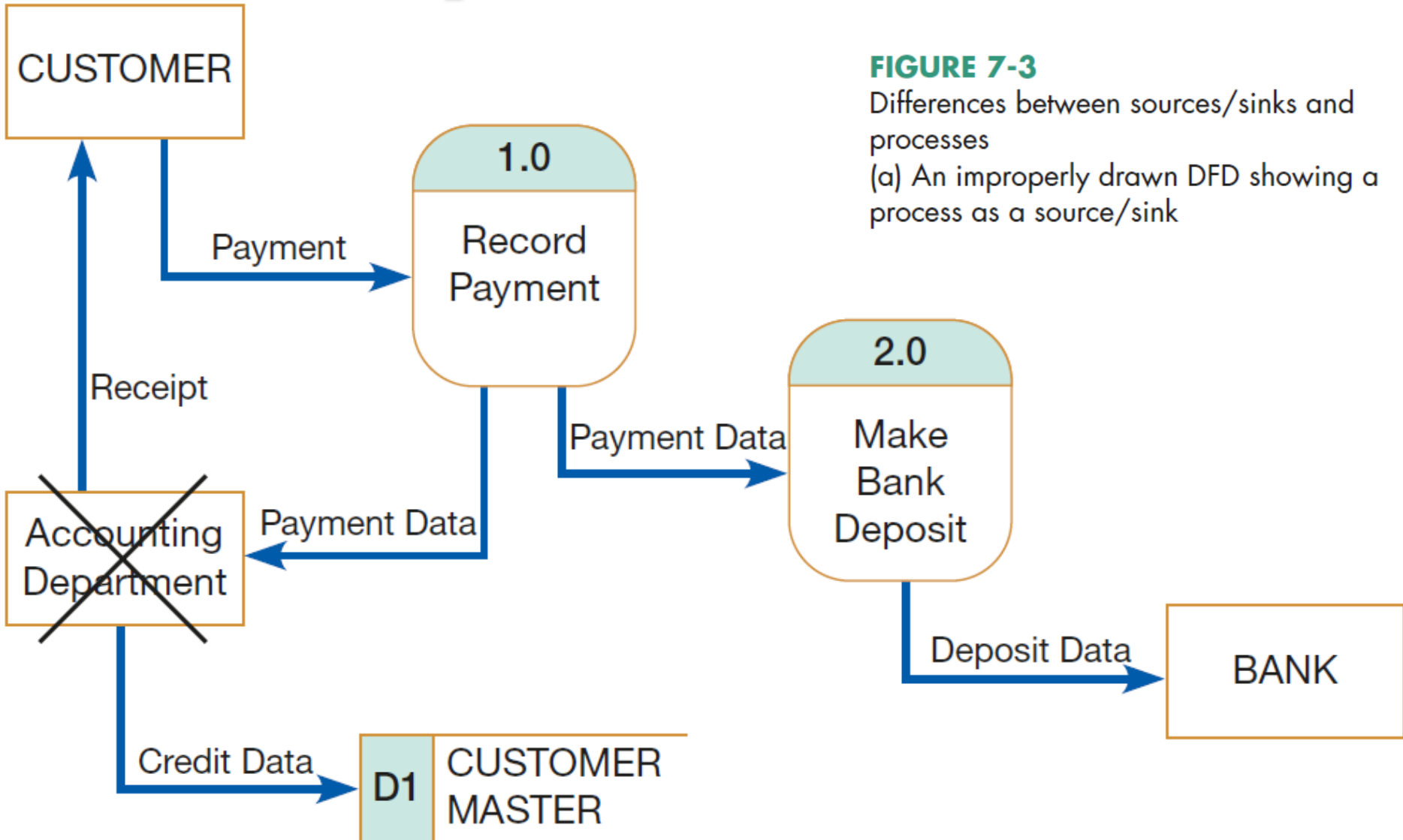
**Figure 4.14** Generic data flow diagram.

# Components of DFD (cont.)

## Sources and sinks (cont.):

- Careful not to *confuse* whether something is a source/sink or a process within a system:
  - occurs most often when the data flows in a system cross office or departmental boundaries (see [e.g.](#))
  - students are then tempted to identify the second office as a [source/sink](#) (to emphasize that data moved from one physical location to another)
  - we are not concerned with *where* the data are physically located, rather *how* they are moving through the system and how they are being processed
  - if the other office is *controlled* by your system  $\Rightarrow$  then you should represent the second office as one or more [processes](#)

# Components of DFD (cont.)



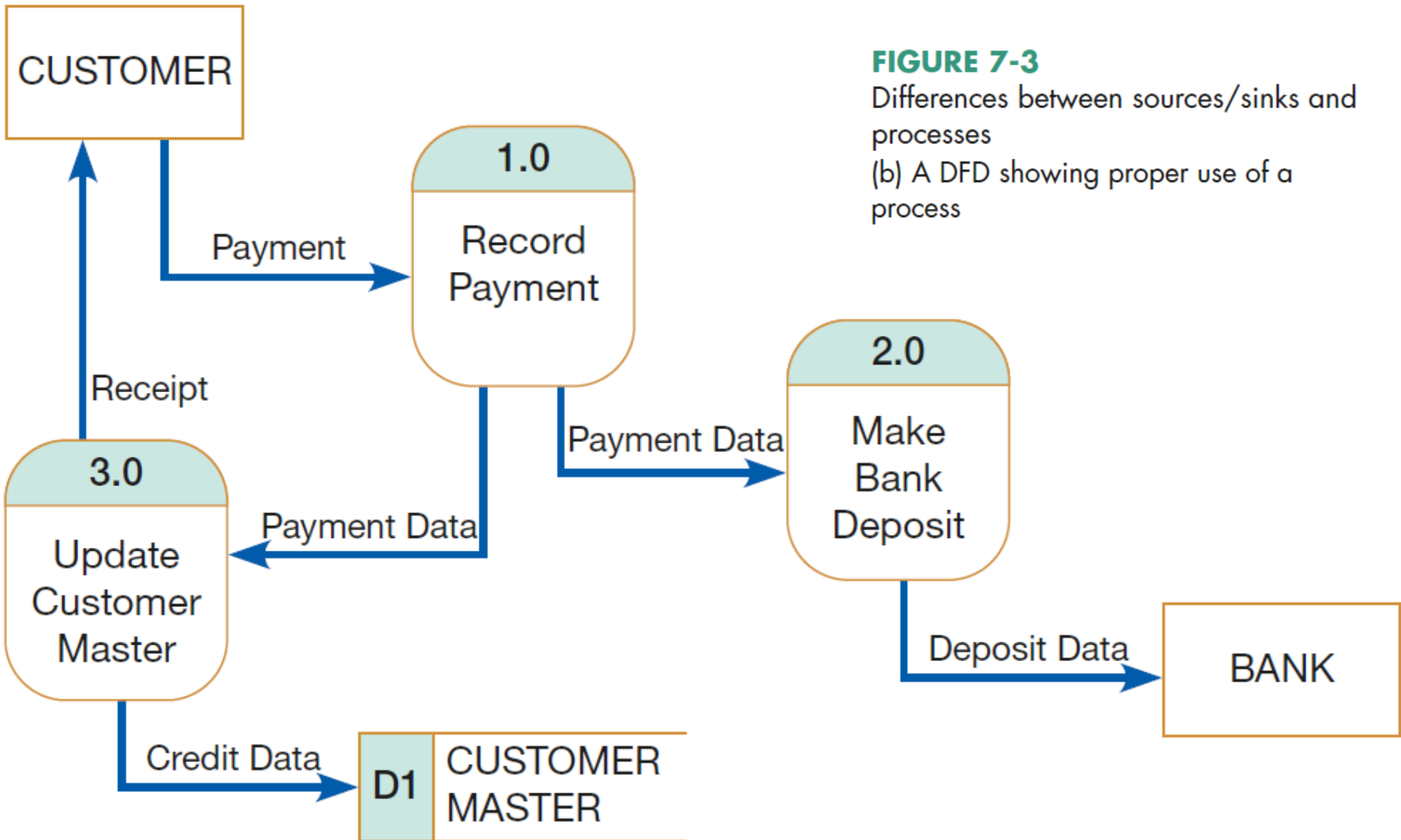
**FIGURE 7-3**

Differences between sources/sinks and processes  
(a) An improperly drawn DFD showing a process as a source/sink

# Components of DFD (cont.)

**FIGURE 7-3**

Differences between sources/sinks and processes  
(b) A DFD showing proper use of a process



# Videos to Watch

- **What is DFD? Data Flow Diagram Symbols and More**  
<https://youtu.be/6VGTvgaJlIM> (*Smartdraw*)
- **How to Draw Data Flow Diagram?**  
<https://youtu.be/ztZsEl6C-mI> (*Visual Paradigm*)
- **DFD Diagram 0**  
<https://youtu.be/lk85hZkyYPA> (*Visible Analyst*)

# Sources

- **Design of Industrial Information Systems.** Thomas Boucher, and Ali Yalcin. Academic Press. First Ed. 2006. [Chapter 4.](#)
- **Modern Systems Analysis and Design.** Joseph S. Valacich and Joey F. George. Pearson. Eighth Ed. 2017. Chapter 7.

