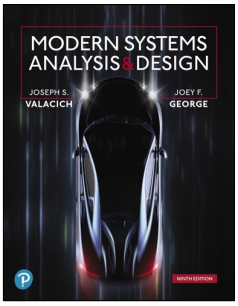
Modern Systems Analysis and Design

Ninth Edition



Chapter 7

Structured System Process Requirements

Pearson

1

Learning Objectives

7.1 Understand the logical modeling of processes by studying examples of data flow diagrams

7.2 Draw data flow diagrams following specific rules and guidelines that lead to accurate and well-structured process models

7.3 Decompose data flow diagrams into lower-level diagrams

7.4 Balance higher-level and lower-level data flow diagrams

7.5 Use data flow diagrams as a tool to support the analysis of information systems

7.6 Discuss process modeling for electronic commerce applications

7.7 Use decision tables to represent the logic of choice in conditional statements

Pearson

2

Introduction

- **Process models** are diagrams that map the movement of data between processes
- Two important concepts related to data flow diagrams:
  - Balancing
  - Decomposition

Pearson

3

Process Modeling

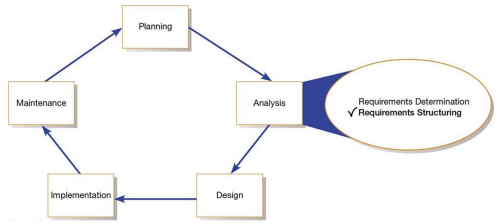
7.1 Understand the logical modeling of processes by studying examples of data flow diagrams

- Process modeling – graphically representing functions, processes that capture, manipulate, store, and distribute data between a system and components within a system
- **Data flow diagram (DFD)** – picture of the movement of data between external entities and the processes and data stores within a system.

Pearson

4

Figure 7-1: Systems Development Life Cycle with the Analysis Phase Highlighted



Pearson

5

Table 7-1: Deliverables for Process Modeling

Deliverables for Process Modeling
1. Context DFD
2. DFDs of the system (adequately decomposed)
3. Thorough descriptions of each DFD component

Pearson

6

Deliverables and Outcomes

7.1 Understand the logical modeling of processes by studying examples of data flow diagrams

- Context data flow diagram
  - Scope of the system
- Data Flow Diagram (DFD) of the system
  - Which processes move and transform data
- DFDs of current logical system
  - Allows analysts to understand the current system
  - Abstract this system to show how new system should meet users requirements

Pearson

7

Data Flow Diagramming Mechanics

7.2 Draw data flow diagrams following specific rules and guidelines that lead to accurate and well-structured process models

- Represent both physical and logical systems
- Only four symbols are used
- Useful for depicting purely logical information flows
- DFDs that detail physical systems differ from system flowcharts which depict details of physical computing equipment

Pearson

8

Definitions and Symbols (1 of 2)

7.2 Draw data flow diagrams following specific rules and guidelines that lead to accurate and well-structured process models

- **Data flow** described as data in motion
- **Data store** – data at rest, which may take the form of many different physical representations
- **Process** – work or actions performed on data so that they are transformed, stored, or distributed

Pearson

9

Definitions and Symbols (2 of 2)

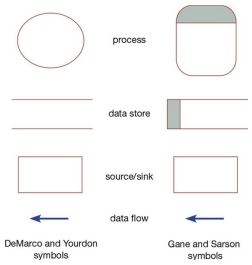
7.2 Draw data flow diagrams following specific rules and guidelines that lead to accurate and well-structured process models

- **Source/sink** – origin and/or destination of data; sometimes referred to as external entities and may consist of:
  - Another organization or unit that sends/receives data to system being analyzed
  - A person supported by the system being analyzed
  - Another IS that is exchanging information with the system being analyzed

Pearson

10

Figure 7-2: Comparison of DeMarco and Yourdon with Gane and Sarson DFD Symbol Sets

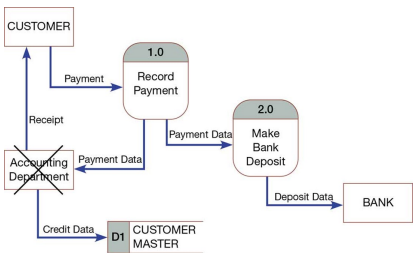


(**Source:** Based on Celko, 1987)

Pearson

11

Figure 7-3: Differences Between Sources/Sinks and Processes (a) An Improperly Drawn DFD Showing a Process as a Source/Sink

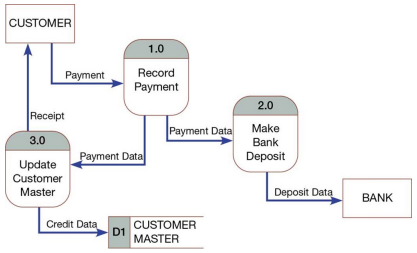


(**Source:** Based on Celko, 1987)

Pearson

12

Figure 7-3: Differences Between Sources/Sinks and Processes (b) A DFD Showing Proper Use of a Process

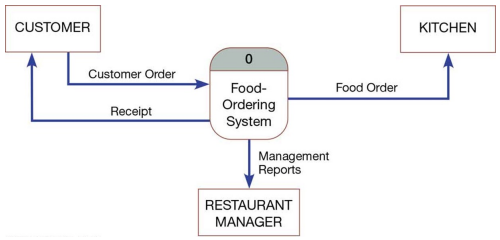


(**Source:** Based on Celko, 1987)

Pearson

13

Figure 7-4: Context Diagram of Hoosier Burger's Food-Ordering System



Pearson

14

Developing DFDs (1 of 2)

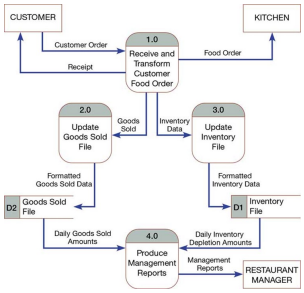
7.2 Draw data flow diagrams following specific rules and guidelines that lead to accurate and well-structured process models

- **Context diagram** – overview of an organizational system that shows the system boundaries, external entities that interact with the system, and the major information flows between the entities and the system

Pearson

15

Figure 7-5: Level-0 DFD of Hoosier Burger's Food-Ordering System



Pearson

16

Developing DFDs (2 of 2)

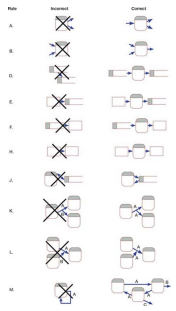
7.2 Draw data flow diagrams following specific rules and guidelines that lead to accurate and well-structured process models

- **Level-0 diagram** – DFD that represents a system's major processes, data flows, and data stores at a high level of detail
- DFD Rules (in addition of those found in table 7-2):
  - The inputs to a process are different from the outputs of that process
  - Objects on a DFD have unique names

Pearson

17

Figure 7-6: Incorrect and Correct Ways to Draw DFDs



Pearson

18

Table 7-2: Rules Governing Data Flow Diagramming (1 of 2)

Rules Governing Data Flow Diagramming
Process:
A. No process can have only outputs. It would be making data from nothing (a miracle). If an object has only outputs, then it must be a source.
B. No process can have only inputs (a black hole). If an object has only inputs, then it must be a sink.
C. A process has a verb phrase label.
Data Store:
D. Data cannot move directly from one data store to another data store. Data must be moved by a process.
E. Data cannot move directly from an outside source to a data store. Data must be moved by a process that receives data from the source and places the data into the data store.
F. Data cannot move directly to an outside sink from a data store. Data must be moved by a process.
G. A data store has a noun phrase label.
Source/Sink:
H. Data cannot move directly from a source to a sink. It must be moved by a process if the data are of any concern to our system. Otherwise, the data flow is not shown on the DFD.
I. A source/sink has a noun phrase label.

(**Source:** Based on Celko, 1987)

Pearson

19

Table 7-2: Rules Governing Data Flow Diagramming (2 of 2)

Rules Governing Data Flow Diagramming
Data Flow:
J. A data flow has only one direction of flow between symbols. It may flow in both directions between a process and a data store to show a read before an update. The latter is usually indicated, however, by two separate arrows because these happen at different times.
K. A fork in a data flow means that exactly the same data goes from a common location to two or more different processes, data stores, or sources/sinks (this usually indicates different copies of the same data going to different locations).
L. A join in a data flow means that exactly the same data come from any of two or more different processes, data stores, or sources/sinks to a common location.
M. A data flow cannot go directly back to the same process it leaves. There must be at least one other process that handles the data flow, produces some other data flow, and returns the original data flow to the beginning process.
N. A data flow to a data store means update (delete or change).
O. A data flow from a data store means retrieve or use.
P. A data flow has a noun phrase label. More than one data flow noun phrase can appear on a single arrow as long as all of the flows on the same arrow move together as one package.

(**Source:** Based on Celko, 1987)

Pearson

20

Decomposition of DFDs (1 of 2)

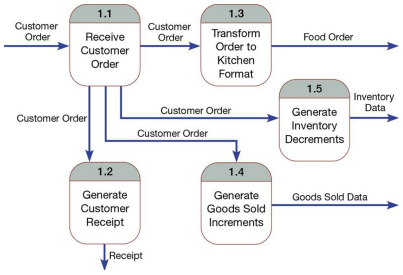
7.3 Decompose data flow diagrams into lower-level diagrams

- **Functional decomposition** – iterative process of breaking the description of a system down into finer and finer detail, which creates a set of charts in which one process on a given chart is explained in greater detail on another chart
  - Example: In figure 7-5 we could break down Process 1.0 into the following tasks: (see figure 7-7)
    1. Receive customer order
    2. Transform order into meaningful form for kitchen
    3. Transform order into a printed receipt for the customer
    4. Transform order into goods sold data
    5. Transform order into inventory data

Pearson

21

Figure 7-7: Level-1 Diagram Showing the Decomposition of Process 1.0 from the level-0 Diagram for Hoosier Burger's Food-Ordering System



22

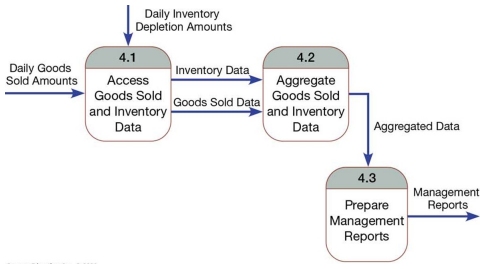
Decomposition of DFDs (2 of 2)

7.3 Decompose data flow diagrams into lower-level diagrams

- **Level-n diagram** – DFD that is the result of n nested decompositions from a process on a level-0 diagram
- **Level-1 diagram** – result of decomposition of a Level-0 diagram
  - Example: Figure 7-7 breaking down the Process 1.0
- Rule of thumb—No DFD should have more than about seven processes
  - Result makes the DFD crowded and difficult to understand

23

Figure 7-8: Level-1 Diagram Showing the Decomposition of Process 4.0 from the level-0 Diagram for Hoosier Burger's Food-Ordering System



24

Figure 7-9: Level-2 Diagram Showing the Decomposition of Process 4.3 from the Level-1 Diagram for Process 4.0 for Hoosier Burger's Food-Ordering System

Pearson

25

Balancing DFDs (1 of 2)

7.4 Balance higher-level and lower-level data flow diagrams

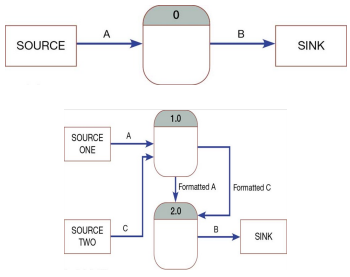
- **Conservation principle** states that you should conserve inputs and outputs to a process at the next level of decomposition
- **Balancing** – conservation of inputs and outputs to a DFD process when that process is decomposed to a lower level

Pearson

26

Figure 7-10: An Unbalanced Set of DFDs

(a) Context Diagram (b) Level-0 Diagram



Pearson

27

Balancing DFDs (2 of 2)

7.4 Balance higher-level and lower-level data flow diagrams

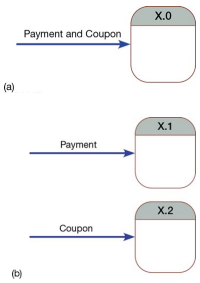
- **Data flow splitting** – when a higher level is split with different paths to a lower level DFD
  - Resulting DFD remains balanced, just split in two
- To "balance" means to have the same number of inputs as outputs on lower level DFDs
- Balancing and keeping DFD as simple as possible led to advance rules (see table 7-3)

Pearson

28

Figure 7-11: Example of Data Flow Splitting

(a) Composite Data Flow (b) Disaggregated Data Flow



Pearson

29

Table 7-3: Advanced Rules Governing Data Flow Diagramming

Advanced Rules Governing Data Flow Diagramming
Q. A composite data flow on one level can be split into component data flows at the next level, but no new data can be added and all data in the composite must be accounted for in one or more subflows.
R. The inputs to a process must be sufficient to produce the outputs (including data placed in data stores) from the process. Thus, all outputs can be produced, and all data in inputs move somewhere: to another process or to a data store outside the process or onto a more detailed DFD showing a decomposition of that process.
S. At the lowest level of DFDs, new data flows may be added to represent data that are transmitted under exceptional conditions; these data flows typically represent error messages (e.g., "Customer not known; do you want to create a new customer?") or confirmation notices (e.g., "Do you want to delete this record?").
T. To avoid having data flow lines cross each other, you may repeat data stores or sources/sinks on a DFD. Use an additional symbol, like a double line on the middle vertical line of a data store symbol or a diagonal line in a corner of a sink/source square, to indicate a repeated symbol.

(**Source:** Based on Celko, 1987)

Pearson

30

Guidelines for Drawing DFDs (1 of 2)

7.4 Balance higher-level and lower-level data flow diagrams

- Guidelines for drawing DFDs:
  - **DFD completeness** – extent to which all necessary components of a DFD have been included and fully described
    - Must include all necessary components for system
    - Each component must be fully described in the project dictionary
  - **DFD consistency** – extent to which information contained on one level of a set of nested DFDs is also included on other levels
    - Example of a gross violation = A level-1 diagram with no level-0 diagram

Pearson

31

Guidelines for Drawing DFDs (2 of 2)

7.4 Balance higher-level and lower-level data flow diagrams

- **Timing**
  - Not represented well on DFDs
  - Draw DFDs as if system has never started nor will ever stop
- **Iterative Development**
  - Count on drawing same diagram over and over
- **Primitive DFD** – lowest level of decomposition for a DFD. Rules include:
  - Reduce each process to a single decision, calculation, or database
  - Each data store represents data about a singe entity
  - When system user does not need to see any more detail
  - When each data flow does not need to be split any further
  - When each business form, transaction is shown as a single data flow
  - When there is a separate process for each choice on lowest level

Pearson

32

Using DFDs as Analysis Tools

7.5 Use data flow diagrams as a tool to support the analysis of information systems

- **Gap analysis** – process of discovering discrepancies between two or more sets of DFDs or discrepancies within a single DFD
- Inefficiencies in a system can be identified through DFDs

Pearson

33

Using DFDs in Business Process Reengineering

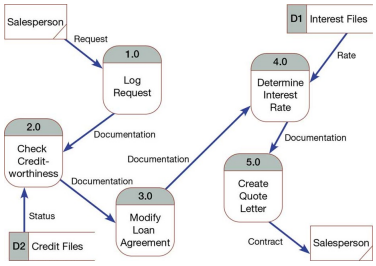
7.5 Use data flow diagrams as a tool to support the analysis of information systems

- DFDs are useful for modeling processes in business process reengineering (BPR)

Pearson

34

Figure 7-16: IBM Credit Corporation's Primary Work Process Before BPR

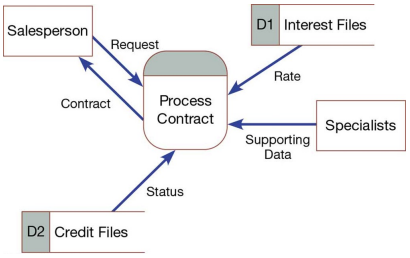


(**Source:** Based on Hammer & Champy, 1993)

Pearson

35

Figure 7-17: IBM Credit Corporation's Primary Work Process After BPR



(**Source:** Based on Hammer and Champy, 1993)

Pearson

36

Modeling Logic with Decision Tables (1 of 2)

7.7 Use decision tables to represent the logic of choice in conditional statements

Decision table – matrix representation of the logic of a decision; it specifies the possible conditions for the decision and the resulting actions

Condition stubs – part of a decision table that lists the conditions relevant to the decision

Action stubs – part of a decision table that lists the actions that result for a given set of conditions

Rules – part of a decision table that specifies which actions are to be followed for a given set of conditions

Pearson

37

Figure 7-18: Complete Decision Table for Payroll System Example

	Conditions/ Courses of Action	Rules					
		1	2	3	4	5	6
Condition Stubs	Employee type	S	H	S	H	S	H
	Hours worked	<40	<40	40	40	>40	>40
Action Stubs	Pay base salary	X		X		X	
	Calculate hourly wage		X		X		X
	Calculate overtime						X
	Produce absence report		X				

Pearson

38

Modeling Logic with Decision Tables (2 of 2)

7.7 Use decision tables to represent the logic of choice in conditional statements

- **Indifferent condition –** in a decision table, a condition whose value does not affect which actions are taken for two or more roles
- Procedures for creating decisions tables:
  1. Name the condition and the values that each can assume
  2. Name all possible actions that can occur
  3. List all possible rules
  4. Define the actions for each rule
  5. Simplify the table

Pearson

39

Figure 7-19: Reduced Decision Table for Payroll System Example

Conditions/ Courses of Action	Rules			
	1	2	3	4
Employee type	S	H	H	H
Hours worked	–	<40	40	>40
Pay base salary	X			
Calculate hourly wage		X	X	X
Calculate overtime				X
Produce absence report		X		

Pearson

40

Electronic Commerce Application: Process Modeling Using Data Flow Diagrams

7.6 Discuss process modeling for electronic commerce applications

- Process modeling for Pine Valley Furniture's WebStore:
  - Completed JAD session
  - Began translating the WebStore structure into data flow diagrams
    - Identified six high-level processes (see figure 7-4)

Pearson

41

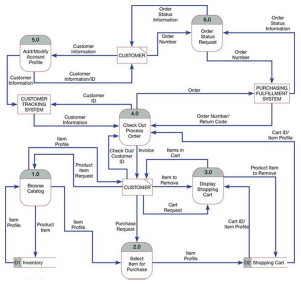
Table 7-4: System Structure of the WebStore and Corresponding Level-0 Processes

WebStore System	Processes
• Main Page – Product Line (Catalog) ▪ Desks ▪ Chairs ▪ Tables ▪ File Cabinets – Shopping Cart – Checkout – Account Profile – Order Status/History – Customer Comments	Information Display (minor/ no processes) 1.0 Browse Catalog 2.0 Select Item for Purchase 3.0 Display Shopping Cart 4.0 Check Out Process Order 5.0 Add/Modify Account Profile 6.0 Order Status Request Information Display (minor/no processes)
• Company Information	
• Feedback	
• Contact Information	

Pearson

42

Figure 7-22: Level-0 for the WebStore



Pearson

43

Summary (1 of 2)

- In this chapter you learned how to:
  - Understand the logical modeling of processes by studying examples of data flow diagrams
  - Draw data flow diagrams following specific rules and guidelines that lead to accurate and well-structured process models
  - Decompose data flow diagrams into lower-level diagrams
  - Balance higher-level and lower-level data flow diagrams

Pearson

44

Summary (2 of 2)

- In this chapter you learned how to:
  - Use data flow diagrams as a tool to support the analysis of information systems
  - Discuss process modeling for electronic commerce applications
  - Use decision tables to represent the logic of choice in conditional statements

Pearson

45

Copyright

46