Using Dataflow Diagrams

7

Systems Analysis and Design, 7e
Kendall & Kendall

Learning Objectives

- Comprehend the importance of using logical and physical data flow diagrams (DFDs) to graphically depict movement for humans and systems in an organization
- Create, use, and explode logical DFDs to capture and analyze the current system through parent and child levels
- Develop and explode logical DFDs that illustrate the proposed system
- Produce physical DFDs based on logical DFDs you have developed
- Understand and apply the concept of partitioning of physical DFDs

Kendall & Kendall   7-2

Data Flow Diagrams

- Graphically characterize data processes and flows in a business system
- Depict:
  - System inputs
  - Processes
  - outputs

Kendall & Kendall   7-3

Major Topics

- Data flow diagram symbols
- Data flow diagram levels
- Creating data flow diagrams
- Physical and logical data flow diagrams
- Partitioning
- Communicating Using Data Flow Diagrams

Kendall & Kendall   7-4

Advantages of the Data Flow Approach

- Freedom from committing to the technical implementation too early
- Understanding of the interrelatedness of systems and subsystems
- Communicating current system knowledge to users
- Analysis of the proposed system

Kendall & Kendall   7-5

Basic Symbols

- A double square for an external entity
- An arrow for movement of data from one point to another
- A rectangle with rounded corners for the occurrence of a transforming process
- An open-ended rectangle for a data store

Kendall & Kendall   7-6

Figure 7.1 The four basic symbols used in data flow diagrams, their meanings, and examples

Kendall & Kendall   7-7

External Entities

- Represent another department, a business, a person, or a machine
- A source or destination of data, outside the boundaries of the system
- Should be named with a noun

Kendall & Kendall   7-8

Data Flow

- Shows movement of data from one point to another
- Described with a noun
- Arrowhead indicates the flow direction
- Represents data about a person, place, or thing

Kendall & Kendall   7-9

Process

- Denotes a change in or transformation of data
- Represents work being performed in the system
- Naming convention
  - Assign the name of the whole system when naming a high-level process
  - To name a major subsystem attach the word subsystem to the name
  - Use the form verb-adjective-noun for detailed processes

Kendall & Kendall   7-10

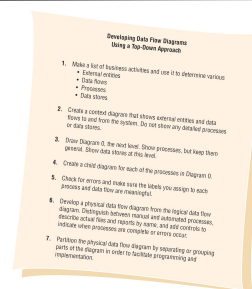## Data Store

- A depository for data that allows examination, addition, and retrieval of data
- Named with a noun, describing the data
- Data stores are usually given a unique reference number, such as D1, D2, D3
- Represents a:
  - Filing cabinet
  - Database
  - Computerized file

Kendall & Kendall   7-11

Figure 7.2 Steps in developing data flow diagrams
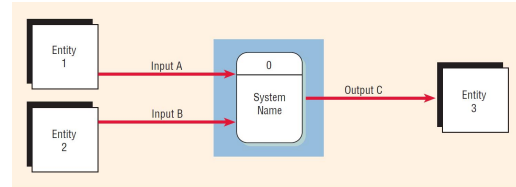

Kendall & Kendall   7-12

Creating the Context Diagram

- The highest level in a data flow diagram
- Contains only one process, representing the entire system
- The process is given the number 0
- All external entities, as well as Major data flows are shown

Figure 7.3 Context diagram

Drawing Diagram 0

- The explosion of the context diagram
- May include up to nine processes
- Each process is numbered
- Major data stores and all external entities are included

Drawing Diagram 0 (Continued)

- Start with the data flow from an entity on the input side
- Work backwards from an output data flow
- Examine the data flow to or from a data store
- Analyze a well-defined process
- Take note of any fuzzy areas

Figure 7.3 Note the greater detail in diagram 0
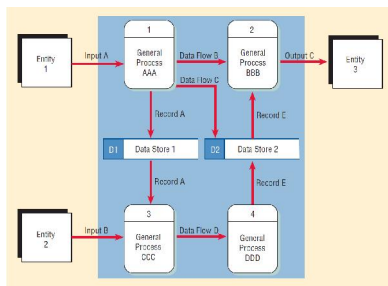
Data Flow Diagram Levels

- Data flow diagrams are built in layers
- The top level is the Context level
- Each process may explode to a lower level
- The lower level diagram number is the same as the parent process number
- Processes that do not create a child diagram are called primitive

3

Creating Child Diagrams

- Each process on diagram 0 may be exploded to create a child diagram
- A child diagram cannot produce output or receive input that the parent process does not also produce or receive
- The child process is given the same number as the parent process
  - Process 3 would explode to Diagram 3

Creating Child Diagrams (Continued)

- Entities are usually not shown on the child diagrams below Diagram 0
- If the parent process has data flow connecting to a data store, the child diagram may include the data store as well
- When a process is not exploded, it is called a primitive process

Figure 7.4 Differences between the parent diagram (above) and the child diagram (below)
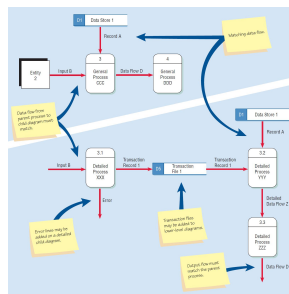
Checking the Diagrams for Errors

- Forgetting to include a data flow or pointing an arrow in the wrong direction
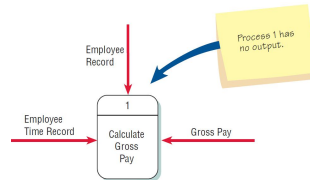
Checking the Diagrams for Errors (Continued)

- Connecting data stores and external entities directly to each other

Checking the Diagrams for Errors (Continued)

- Incorrectly labeling processes or data flow
- Including more than nine processes on a data flow diagram

Checking the Diagrams for Errors (Continued)

- Omitting data flow
- Creating unbalanced decomposition (or explosion) in child diagrams

Figure 7.5 Typical errors that can occur in a data flow diagram (payroll example)
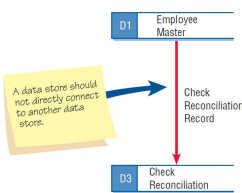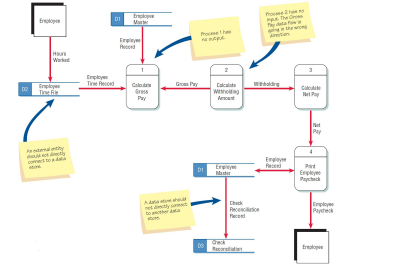
Logical and Physical Data Flow Diagrams

- Logical
  - Focuses on the business and how the business operates
  - Not concerned with how the system will be constructed
  - Describes the business events that take place and the data required and produced by each event
- Physical
  - Shows how the system will be implemented
  - Depicts the system

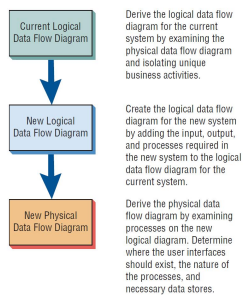**Figure 7.7** Features common of logical and physical data flow diagrams

Design Feature	Logical	Physical
What the model depicts	How the business operates.	How the system will be implemented (or how the current system operates).
What the processes represent	Business activities.	Programs, program modules, and manual procedures.
What the data stores represent	Collections of data regardless of how the data are stored.	Physical files and databases, manual files.
Type of data stores	Show data stores representing permanent data collections.	Master files, transition files. Any processes that operate at two different times must be connected by a data store.
System controls	Show business controls.	Show controls for validating input data, for obtaining a record (record found status), for ensuring successful completion of a process, and for system security (example: journal records).

Figure 7.8 The progression of models from logical to physical

Derive the logical data flow diagram for the current system by examining the physical data flow diagram and isolating unique business activities.

Create the logical data flow diagram for the new system by adding the input, output, and processes required in the new system to the logical data flow diagram for the current system.

Derive the physical data flow diagram by examining processes on the new logical diagram. Determine where the user interfaces should exist, the nature of the processes, and necessary data stores.

Developing Logical Data Flow Diagrams

- Better communication with users
- More stable systems
- Better understanding of the business by analysts
- Flexibility and maintenance
- Elimination of redundancy and easier creation of the physical model

Developing Physical Data Flow Diagrams

- Clarifying which processes are performed by humans and which are automated
- Describing processes in more detail
- Sequencing processes that have to be done in a particular order
- Identifying temporary data stores
- Specifying actual names of files and printouts
- Adding controls to ensure the processes are done properly

Kendall & Kendall 7-31

Figure 7.10 Physical data flow diagrams contain many items not found in logical data flow diagrams

Contents of Physical Data Flow Diagrams

- Manual processes
- Processes for adding, deleting, changing, and updating records
- Data entry and verifying processes
- Validation processes for ensuring accurate data input
- Sequencing processes to rearrange the order of records
- Processes to produce every unique system output
- Intermediate data stores
- Actual file names used to store data
- Controls to signify completion of tasks or error conditions

Kendall & Kendall 7-32

Event Modeling and Data Flow Diagrams

- An input flow from an external entity is sometimes called a trigger because it starts the activities of a process
- Events cause the system to do something and act as a trigger to the system
- An approach to creating physical data flow diagrams is to create a data flow diagram fragment for each unique system event

Kendall & Kendall 7-33

Event Response Tables

- An event table is used to create a data flow diagram by analyzing each event and the data used and produced by the event
- Every row in an event table represents a data flow diagram fragment and is used to create a single process on a data flow diagram
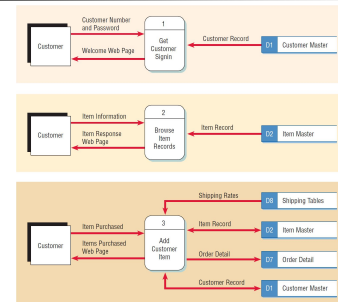
Kendall & Kendall 7-34

Figure 7.12 An event response table for an Internet storefront

Event	Source	Trigger	Activity	Response	Destination
Customer logs on	Customer	Customer number and password	Find customer record and verify password. Send Welcome Web page.	Welcome Web page	Customer
Customer browses items at Web storefront	Customer	Item information	Find item price and quantity available. Send Item Response Web page.	Item Response Web page	Customer
Customer places item into shopping basket at Web storefront	Customer	Item purchase (item number and quantity)	Store data on Order Detail Record. Calculate shipping cost using shipping tables. Update customer total. Update item quantity on hand.	Items Purchased Web page	Customer
Customer checks out	Customer	Clicks "Check Out" button on Web page	Display Customer Order Web page.	Verification Web page	
Obtain customer payment	Customer	Credit card information	Verify credit card amount with credit card company. Send.	Credit card data Customer feedback	Credit card company Customer
Send customer email		Temporal, hourly	Send customer an email confirming shipment.		Customer

Kendall & Kendall 7-35

Figure 7.13 Data flow diagrams for the first three rows of the Internet storefront event response table

Kendall & Kendall 7-36

Use Cases and Data Flow Diagrams

- Each use case defines one activity and its trigger, input, and output
- Allows the analyst to work with users to understand the nature of the processes and activities and then create a single data flow diagram fragment

Kendall & Kendall   7-37

Partitioning Data Flow Diagrams

- Partitioning is the process of examining a data flow diagram and determining how it should be divided into collections of manual procedures and computer programs
- A dashed line is drawn around a process or group of processes that should be placed in a single computer program

Kendall & Kendall   7-38

Reasons for Partitioning

- Different user groups
- Timing
- Processes may be separated into different programs for security
- Similar tasks
- Efficiency
- Consistency
- Security

Kendall & Kendall   7-39

Partitioning Web Sites

- Improves the way humans use the site
- Improves speed of processing
- Ease of maintaining the site
- Keep the transaction secure

Kendall & Kendall   7-40

Communicating Using Data Flow Diagrams

- Use unexploded data flow diagrams early when ascertaining information requirements
- Meaningful labels for all data components

Kendall & Kendall   7-41

Summary

- Data flow diagrams
  - Structured analysis and design tools that allow the analyst to comprehend the system and subsystems visually as a set of interrelated data flows
- DFD symbols
  - Rounded rectangle
  - Double square
  - An arrow
  - Open-ended rectangle

Kendall & Kendall   7-42

Summary (Continued)

- Creating the logical DFD
  - Context-level data flow diagram
  - Level 0 logical data flow diagram
  - Child diagrams
- Creating the physical DFD
  - Create from the logical data flow diagram
  - Partitioned to facilitate programming

Kendall & Kendall

7-43

Summary (Continued)

- Partitioning data flow diagrams
  - Whether processes are performed by different user groups
  - Processes execute at the same time
  - Processes perform similar tasks
  - Batch processes can be combined for efficiency of data
  - Processes may be partitioned into different programs for security reasons

Kendall & Kendall

7-44