Chapter 7
Process Modeling and Data Flow Diagrams (DFD)

Process Modeling and Dataflow Diagrams (DFDs)

Merrill Warkentin
Mississippi State University

Major Topics

- ✓ process modeling
- ✓ data flow diagram symbols
- ✓ data flow diagram levels
- ✓ creating data flow diagrams
- ✓ physical and logical DFDs
- ✓ partitioning
- ✓ (event driven modeling)
- ✓ (use case and data flow diagrams)

2

Process Modeling

- ✓ Graphically representing the functions, or processes, which capture, manipulate, store, and distribute data between a system and its environment and between components of a system
- ✓ Data Flow Diagram (DFD)

3

Data Flow Diagrams (DFD)

- ✓ graphical representation of data processes and flows throughout the organization
- ✓ emphasizes logic of underlying system
- ✓ based on general systems model

4

Why Use DFD?

- ✓ logical information flow of the system
- ✓ determination of physical system construction requirements
- ✓ simplicity of notation
- ✓ establishment of manual and automated systems requirements

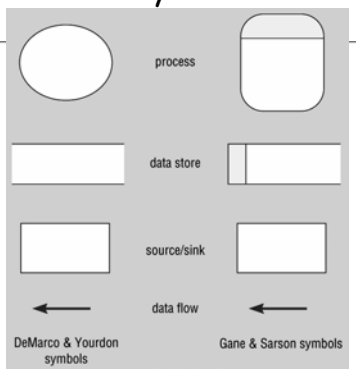
5

Advantages of DFD

- ✓ early understanding of system technical implications
- ✓ establishment of inter-relatedness between business and systems
- ✓ user involvement with system development
- ✓ determine if data and processes have been defined properly

6

DFD - Four Symbols



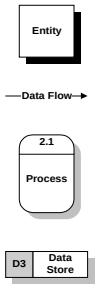
7 © Merrill Warkentin, 2005

DFD - Four Symbols

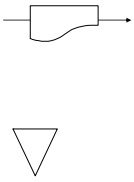
- ✓ Process - work or actions performed on data so that they are transformed, stored, or distributed
- ✓ Date Store - data at rest, notebook, file folder, database
- ✓ Source/Sink - origin and/or destination of the data (external entities)
- ✓ Data flow - data in motion

8 © Merrill Warkentin, 2005

DFD Formatting Symbols



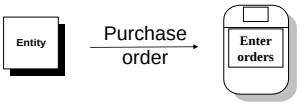
- ✓ external to boundaries of system
  - × sources and destinations of data
- ✓ data movement
  - × electronic exchange
  - × paper documents (sometimes shown as)
- ✓ transformation of data
  - × from input to output
- ✓ data depository for
  - × electronic files
  - × paper files (sometimes shown as)



9 © Merrill Warkentin, 2005

External Entities

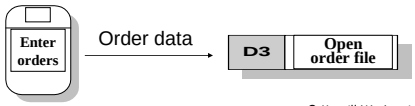
- ✓ supply data to the system
- ✓ receive data/information from the system
- ✓ but not considered part of the system



10 © Merrill Warkentin, 2005

Data Flows

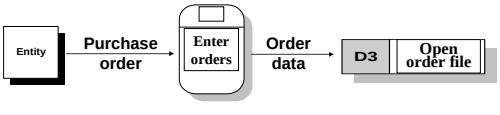
- ✓ represent data being moved to or from
  - × an external entity
  - × a process
  - × a data store
- ✓ shown as a line with arrowheads
- ✓ labeled with description of the type of data, document, or report



11 © Merrill Warkentin, 2005

Processes

- ✓ set of tasks performed on data to convert inputs to outputs
- ✓ labeled as a verb/noun
  - × e.g., ENTER ORDERS
- ✓ data flow out *always* labeled differently than data flow in



12 © Merrill Warkentin, 2005

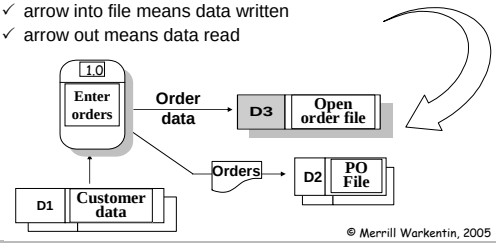
Processes - 4 Types

- ✓ create new data
- ✓ change quality but not content of input
  - × e.g., verify customer number
  - × output = input = (verified) customer number
- ✓ reorganize input
  - × sort
  - × reformat
  - × filter/select
- ✓ convey input without any change

13

Data Stores

- ✓ data at rest
- ✓ corresponds to data files or tables
- ✓ can be electronic or paper files
- ✓ shown as a rectangle (electronic)
- ✓ arrow into file means data written
- ✓ arrow out means data read



14

Balancing DFDs

- ✓ Inputs and outputs must be conserved between levels of DFDs
- ✓ Level n & n+1 must have the same inputs and outputs

15

Five guidelines for drawing DFDs

- ✓ Completeness - all necessary components are included and described
- ✓ Consistency - information on one level is included on other levels
- ✓ Timing - a DFD does NOT indicate when the system is flowing
- ✓ Iterative development - it may take 3 or more DFDs to get it right
- ✓ Primitive DFDs (the lowest level DFD) - stop decomposing when the lowest logical level is reached

16

Problems with DFDs

- ✓ Identify problems on the DFD (see handout)

17

Describe problems (handout)

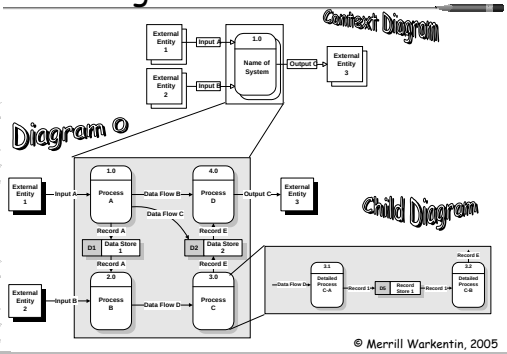
18

DFD Development/Levels

- ✓ context diagram
  - × general highest level diagram
  - × one process representing entire system
- ✓ diagram 0
  - × exploded view of the entire system
  - × up to nine processes
  - × all external entities and data stores
- ✓ child diagrams
  - × exploded view of each individual process
  - × outlined in Diagram 0

19

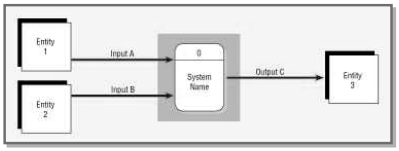
DFD Organization



20

Creating the Context Diagram

- ✓ original should be an overview
  - × basic inputs
  - × general system
  - × outputs
- ✓ highest level
- ✓ contains only one process



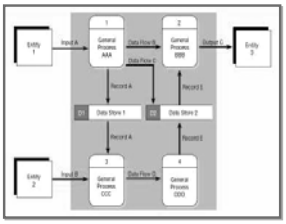
21

Context Diagram Example

22

Drawing Diagram 0

- ✓ explode context diagram
  - × more detail
  - × close-ups of processes
  - × data stores
  - × lower-level data flows
- ✓ may include up to 9 processes



23

Drawing Diagram 0

- ✓ start with data flow from an external entity
- ✓ work backwards from an output data flow
- ✓ examine data flow
- ✓ analyze well defined process
- ✓ take note of problem areas

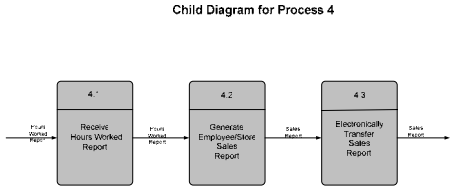
24

Diagram 0 Example

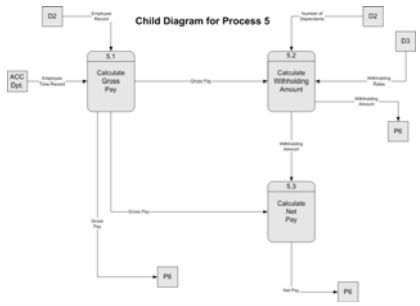
Creating Child Diagrams

- ✓ more detailed levels
  - × each process exploded further
- ✓ parent process
  - × exploded diagram 0 process
- ✓ child diagram
  - × resulting diagram
  - × entities usually not shown at this level
  - × may have error lines here only

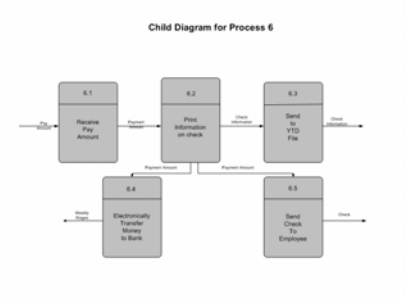
Child Diagram Examples



Child Diagram Examples



Child Diagram Examples



Rules & Guidelines for DFD

A data store must always be connected to a process.

This:

Not this:

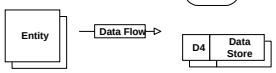
Nor this:

Rules & Guidelines for DFD

External entity must always be connected to a process.

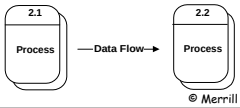
OK -

not OK -



Data may flow out of one process and into another.

OK -



31

Naming Rules for Data Flows

- ✓ data flows must be named
  - × use a NOUN
    (what data are being used by a process?)
- ✓ output flow from a process
  - × must have a different name than the input flow
  - × output cannot be the same as the input
- ✓ data traveling together
  - × should be shown
    and named as one data flow
  - × example: orders and payments flowing together
    should be labeled as "orders and payments"

32

Data Stores' Rules & Guidelines

- ✓ if no input data flowing to them,
  may not be properly represented
  - × where did the data
    in the data store come from?
- ✓ should be uniquely numbered
  on the flow diagram
- ✓ description of the data attributes should be
  contained in a similarly numbered table
  structure / data attribute dictionary
  - × you can create the dictionary by printing out the
    ACCESS data structures for each table

33

Rules & Guidelines for Processes

- ✓ process performs one simple, well-defined task,
  activity or function
- ✓ must have at least one input and
  one output data flow
  - × if no input, process is creating output from "thin air"
    (where is the data to be processed
    into output coming from?
  - × if no output, process is "black hole" or "data sink"
    (why is data flowing to a process
    that does not produce any output?)
- ✓ data should be sent only to processes
  that use them

34

Rules & Guidelines for Processes

- ✓ should be uniquely named and numbered
- ✓ description of the process should be
  in a similarly numbered and indexed
  "process dictionary"
- ✓ data store to can be the only receiver
  of output from a process
- ✓ data store to can be the only source
  of input to a process

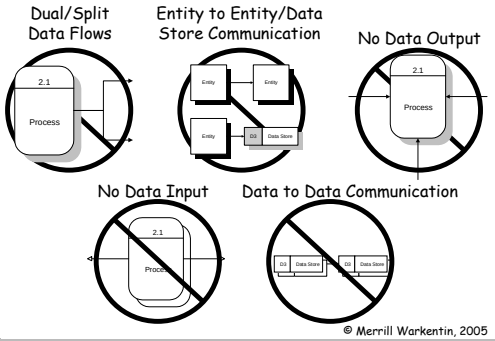
35

Common Errors

- ✓ not including data flow
- ✓ placing arrowhead in the wrong direction
- ✓ directly connecting data stores & external
  entities
- ✓ labeling incorrectly
- ✓ including more than 9 processes
- ✓ omitting data flow
- ✓ having unbalanced composition

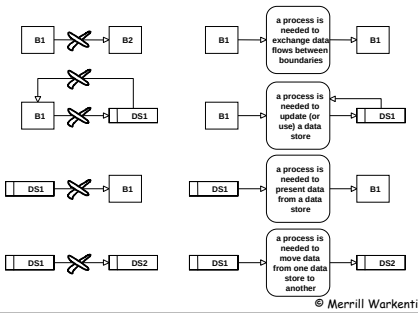
36

Common DFD Errors



37

Illegal Data Flows



38

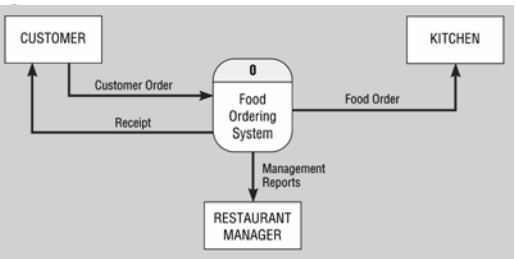
Practice Question - THINK

Which is an error condition?

39

DFD for Hoosier Burger (food ordering system)

Context diagram - highest view of the system



40

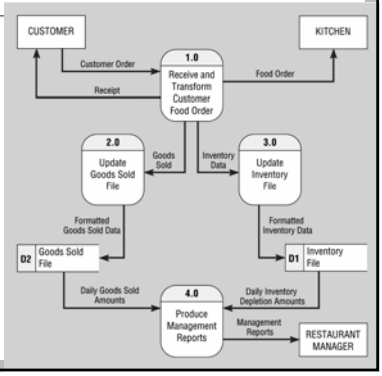
DFD for Hoosier Burger (food ordering system)

Level-0 DFD -

systems major processes, data flows, and data stores

Level-n DFD -

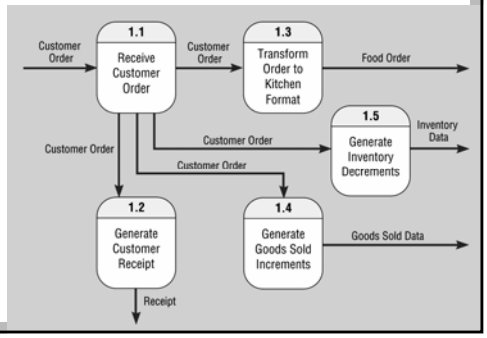
n nested decompositions of sub-processes from the level-0 diagram



41

DFD for Hoosier Burger (food ordering system)

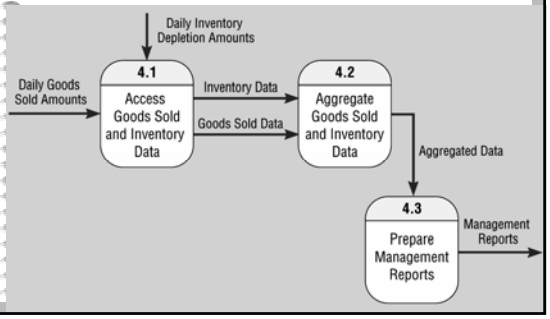
Level-1 - decomposes process 1.0 from level-0



42

DFD for Hoosier Burger (food ordering system)

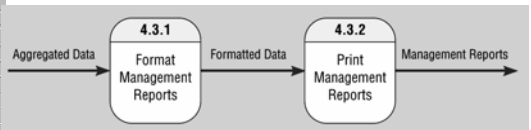
Level-1 - decomposes process 4.0 from level-0



43

DFD for Hoosier Burger (food ordering system)

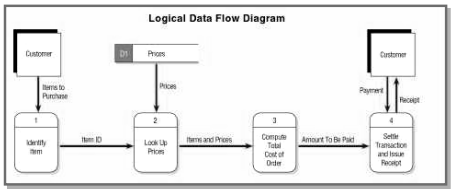
Level-2 Diagram Showing the Decomposition of Process 4.3 from the Level-1 Diagram for Process 4.0



44

Logical DFD

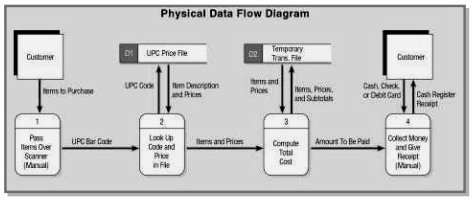
- ✓ focus on how the business operates
- ✓ processes that exist regardless of the type of system implemented



45

Physical DFD

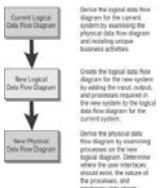
- ✓ details physical system requirements
- ✓ contains construction specifications (software, hardware, people)



46

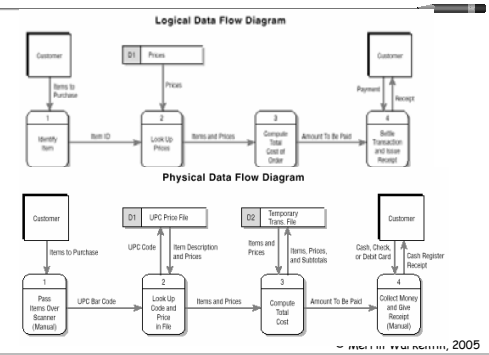
Logical to Physical DFD

- ✓ create a logical DFD of the current system
- ✓ add all data and processes not in the current system which must be present in the new system
- ✓ derive the physical data flow diagram for the new system
- ✓ logical DFD of the current system will lead to appropriate logical DFD of the new system
- ✓ physical DFD will follow naturally if the logical DFD is accurate



47

Logical to Physical DFD



48

Logical DFD - Advantages

- ✓ better communication
- ✓ more stable systems
- ✓ better understanding
- ✓ flexibility and maintenance
- ✓ elimination of redundancies

49

Physical DFD - Advantages

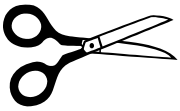
- ✓ clarifying manual & automated processes
- ✓ detailed process description
- ✓ sequencing processes
- ✓ identifying temporary data stores
- ✓ specifying name of files and printouts
- ✓ adding controls

50

Partitioning Data Flow Diagrams

The process of separating activities into groups based on automation, manual, and user needs

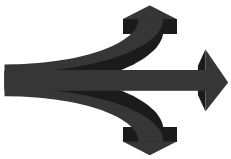
- ✓ identify separate user groups
- ✓ process timing
- ✓ similar task
- ✓ efficiency
- ✓ data consistency
- ✓ security



51

Reasons for Partitioning

- ✓ different user groups
- ✓ timing
- ✓ similar tasks
- ✓ efficiency
- ✓ consistency
- ✓ security



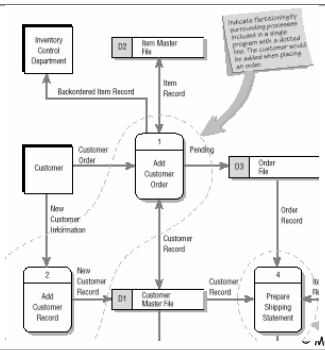
52

Partitioning the DFD

Look . . .

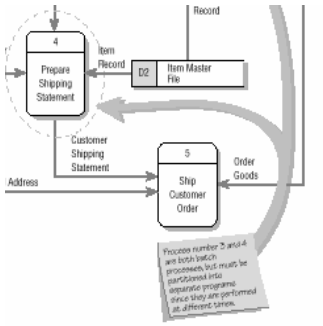
53

Indicate with dotted line



54

Separate batch programs



55

Practice Question - THINK

Which of the following is NOT a reason for partitioning processes into separate programs?

56

Summary

- ✓ standard communication tool for design team
- ✓ promotes better understanding of system
- ✓ ensures that all components are included
- ✓ leads to effective logical design

57