Princess Noura University
Department of Computer Science

IS321

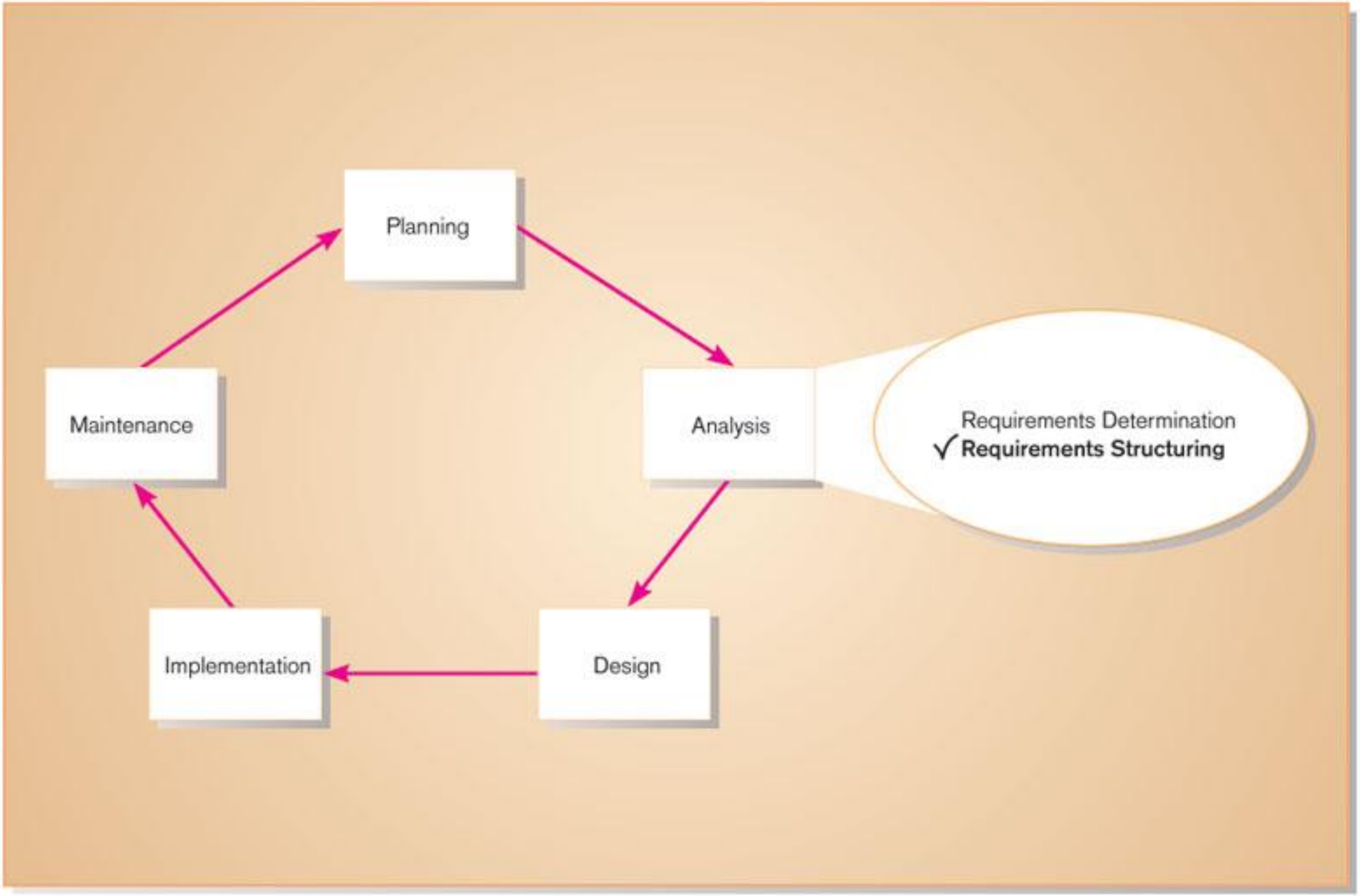
Chapter 7PartA

Analyzing System Process Requirements

Learning Objectives

- ✓ Understand logical process modeling via data flow diagrams (DFDs).
- ✓ Draw DFDs of well structured process models.
- ✓ Decompose DFDs into lower-level diagrams.
- ✓ Balance high-level and low-level DFDs.
- ✓ Use DFDs for analyzing information systems.

Figure 7-1 Systems development life cycle with the analysis phase highlighted



Process Modeling

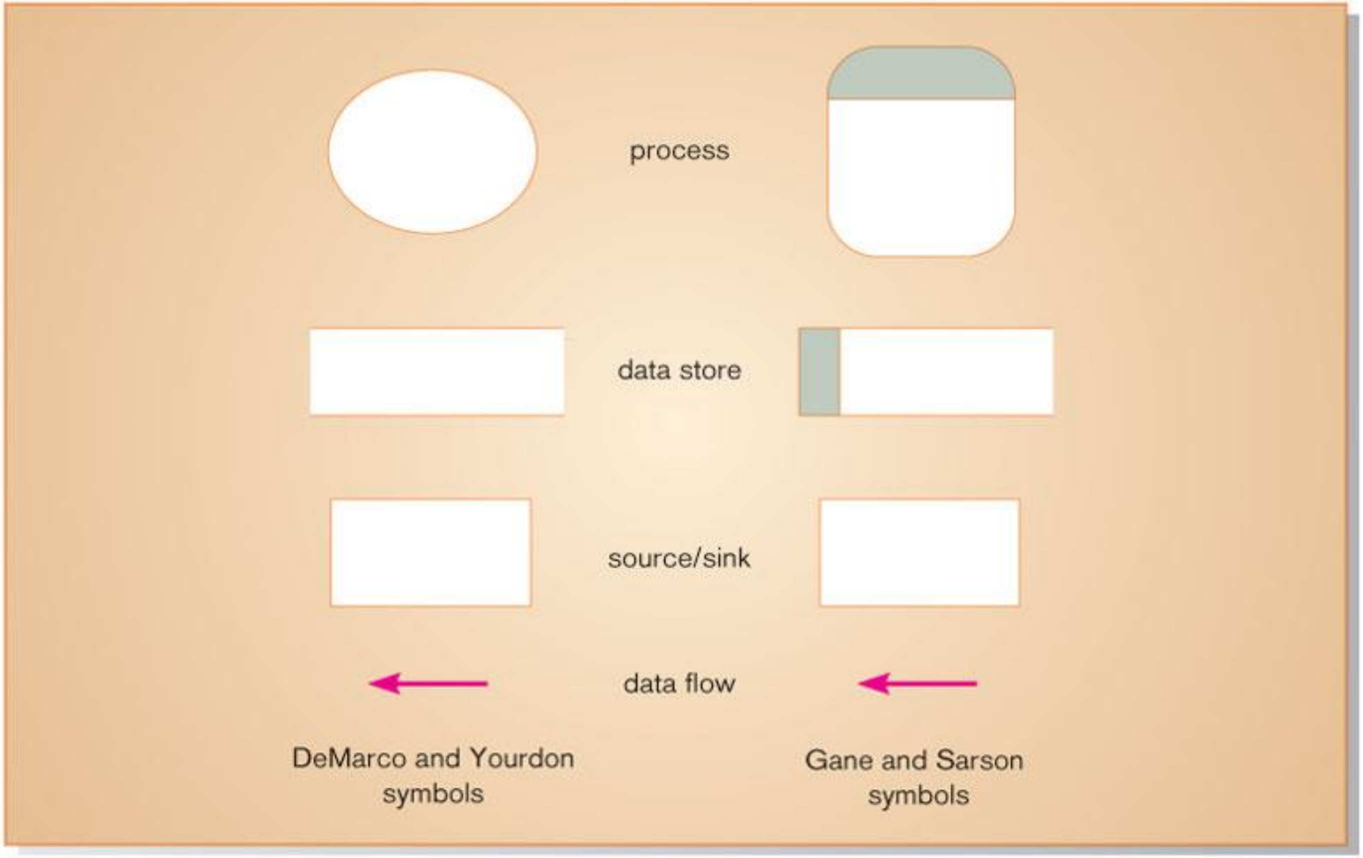
- Graphically represent the processes that capture, manipulate, store, and distribute data between a system and its environment and among system components
- Use information gathered during requirements determination
- Processes and data structures are modeled

Process Modeling (cont.) Deliverables and Outcomes

- Context data flow diagram (DFD)
  - Scope of system
  - Indicating which elements are inside and which elements are outside the system.
- DFDs of the system
  - Specify which process move and transform data, accepting inputs and producing outputs.
- Detailed description of each DFD component
  - Project dictionary.

DFD Symbols

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



DFD Symbols (cont.)

- Data Flow
  - Depicts data that are in motion and moving as a unit from one place to another in the system.
  - Drawn as an arrow
  - Select a meaningful name to represent the data

DFD Symbols (cont.)

- Data Store
  - Depicts data at rest
  - May represent data in
    - File folder
    - Computer-based file
    - Notebook
  - The name of the store as well as the number are recorded in between lines

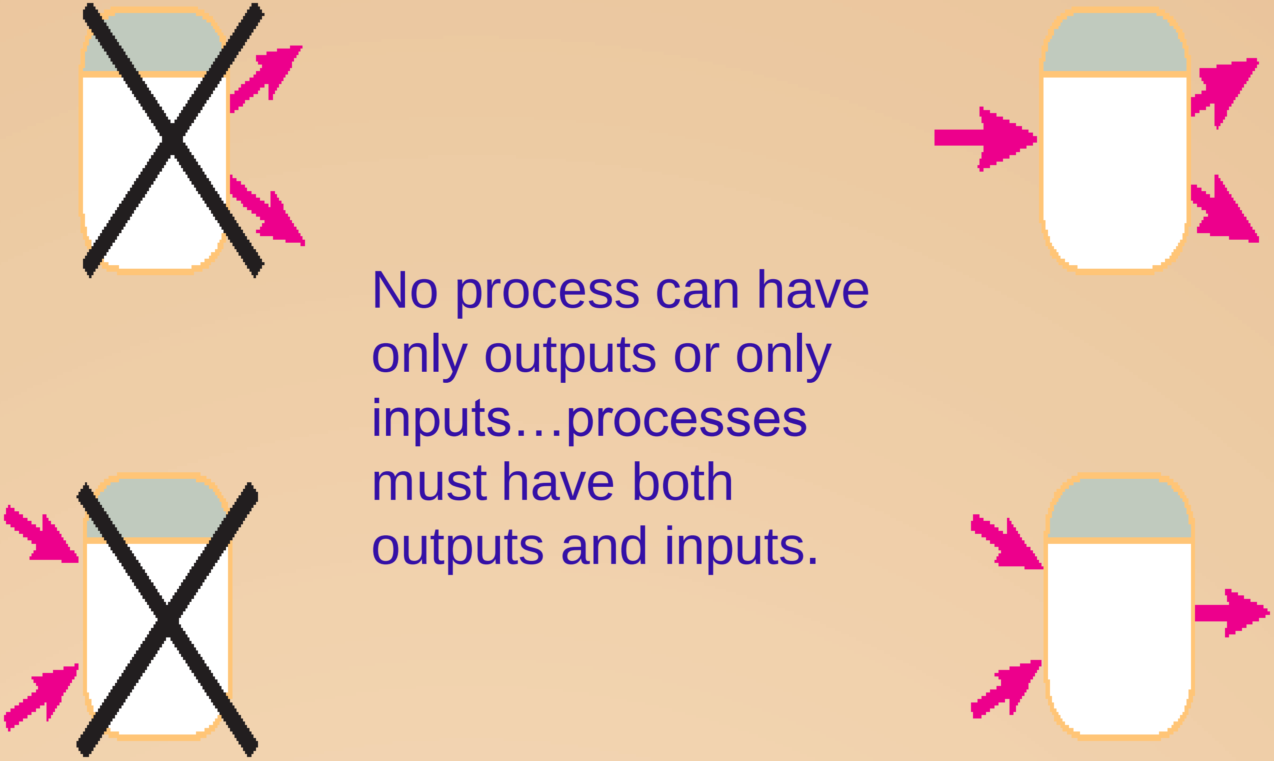
DFD Symbols (cont.)

- Process
  - Depicts work or action performed on data so that they are transformed, stored or distributed
  - Number of process as well as name are recorded

DFD Symbols (cont.)

- Source/Sink
  - Depicts the origin and/or destination of the data
  - Sometimes referred to as an external entity
  - Drawn as a rectangle symbol
  - Because they are external, many characteristics are not of interest to us

DFD Diagramming Rules
Process

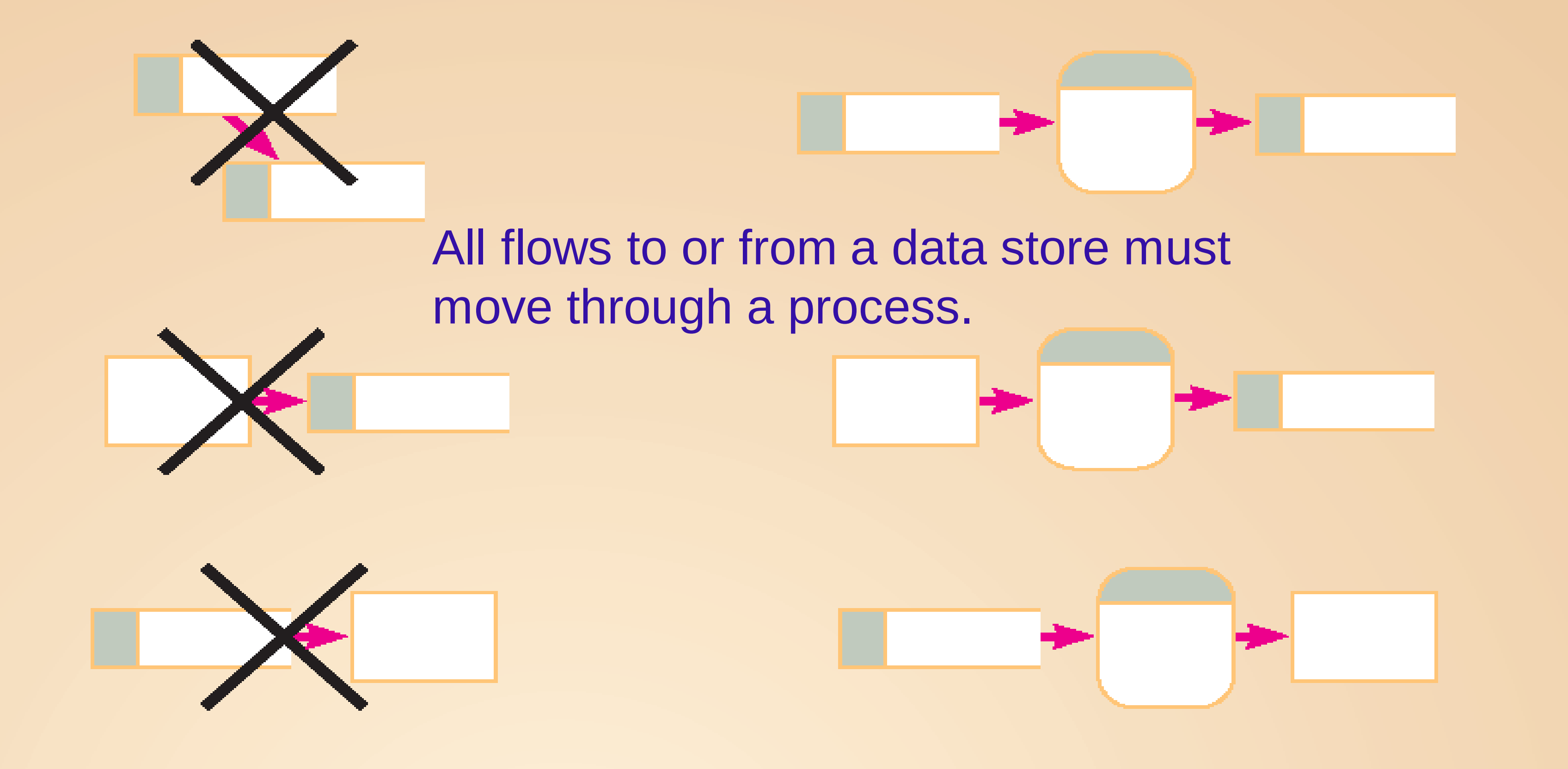


No process can have only outputs or only inputs…processes must have both outputs and inputs.

Process labels should be verb phrases.

Process labels are the same as verbs used in programming (ex. Merge, Sort, Read..).

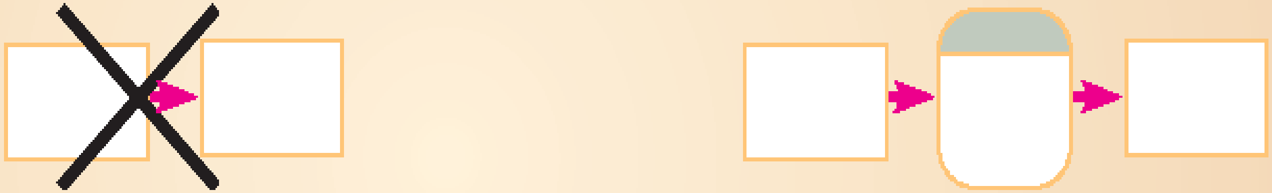
DFD Diagramming Rules
Data Store



All flows to or from a data store must move through a process.

Data store labels should be noun phrases.

DFD Diagramming Rules
Source/Sink

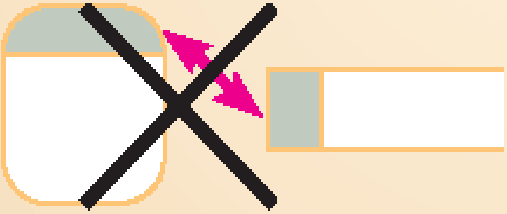


No data moves directly between external entities without going through a process.

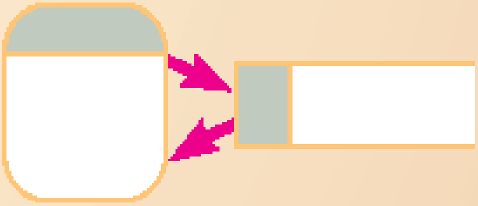
Interactions between external entities without intervening processes are outside the system and therefore not represented in the DFD.

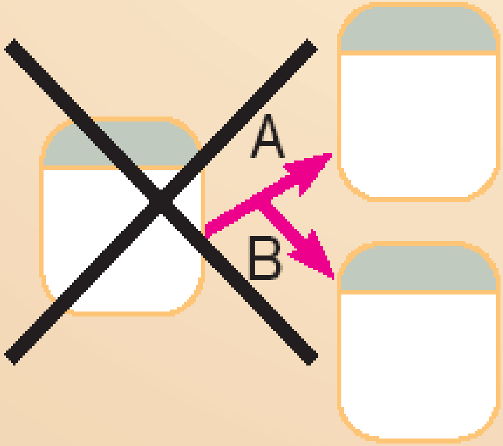
Source and sink labels should be noun phrases.

DFD Diagramming Rules
Data Flow

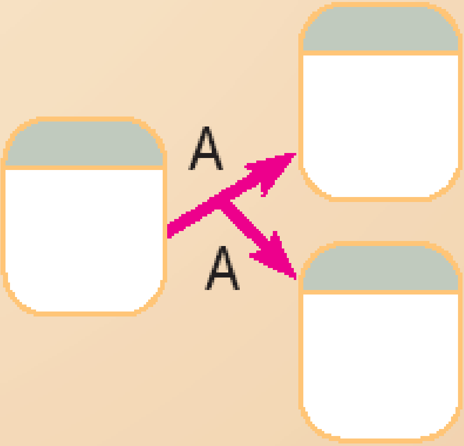


Bidirectional flow between process and data store is represented by two separate arrows.

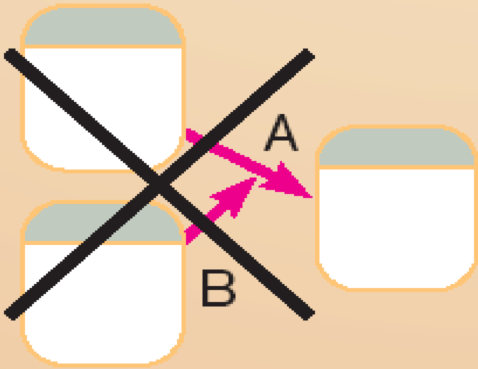




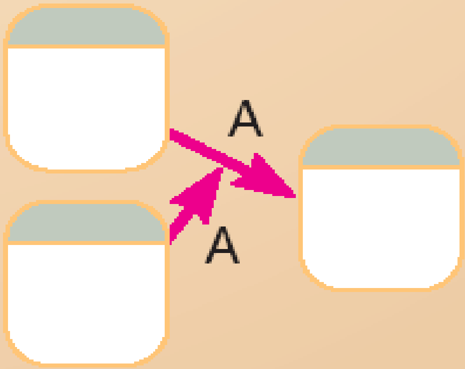
Forked data flow must refer to exact same data item (not different data items) from a common location to multiple destinations.

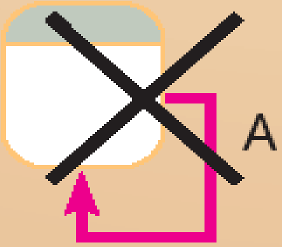


DFD Diagramming Rules
Data Flow (cont.)

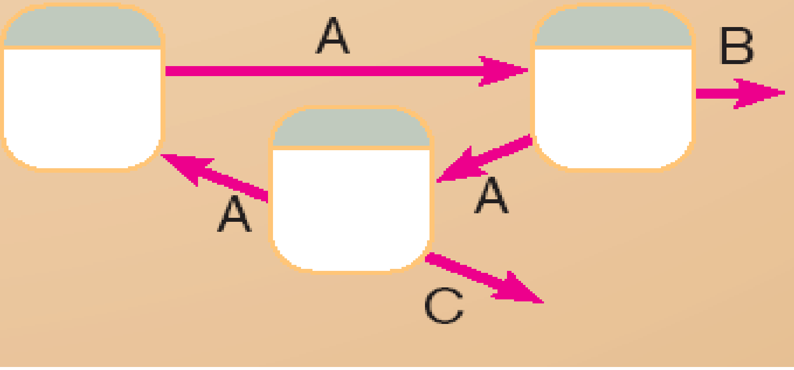


Joined data flow must refer to exact same data item (not different data items) from multiple sources to a common location.





Data flow cannot go directly from a process to itself, must go through intervening processes.



DFD Diagramming Rules
Data Flow (cont.)

- Data flow from a process to a data store means update (insert, delete or change).
- Data flow from a data store to a process means retrieve or use.
- Data flow labels should be noun phrases.
- The inputs to a process are different from the outputs.
- Objects on DFD have unique names.
- You may repeat data stores and sources/sinks.

An Example

Students send in an application form containing their personal details, and their desired course. The university checks that the course is available and that the student has necessary academic qualifications.

If the course is available the student is enrolled in the course, and the university confirms the enrolment by sending a confirmation letter to the student.

If the course is unavailable the student is sent a rejection letter.

1. Read the problem description carefully looking for:

- people/organisations/things that supply information to or use information from the system => external entities (EE)
- actions/doing words/verbs => Processes (P)
- movement/exchange of information/data between external entities to processes, and processes to processes => data flows (DF)
- store/record information/data => data stores(DS)

2. Walk through the system in its logical sequence.

- A **student** (EE) sends in an *application form* (DF) containing their personal details, and their desired course
- The university checks (P) that the course is available.
- If the course is available the student is enrolled (P) in the course, and the university confirms (P) the enrolment by sending a *confirmation letter* (DF) that they are registered for the course to the student.
- Or if the course is unavailable the student is sent a *rejection letter* (DF).

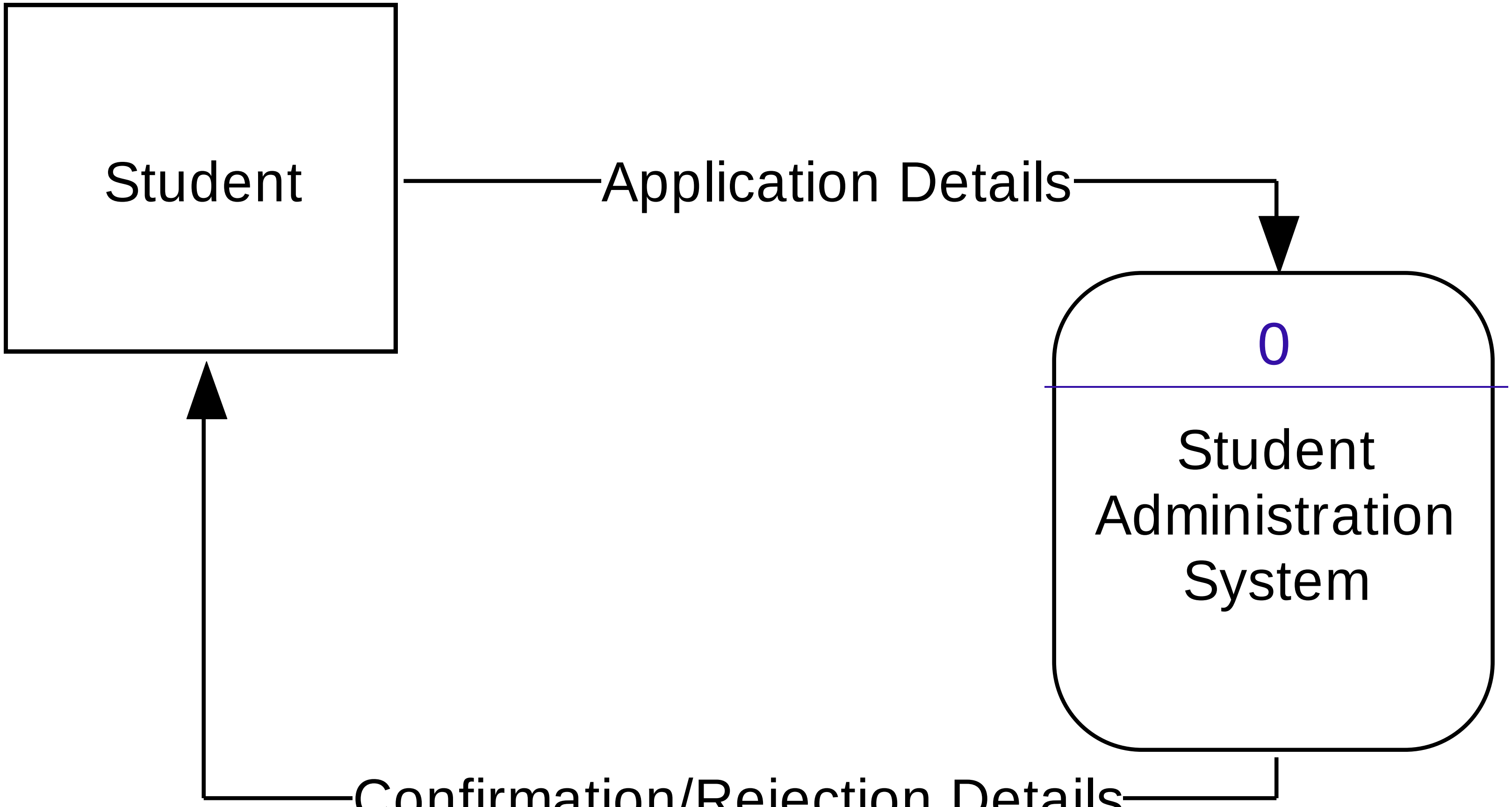
Context diagram

- Highest level DFD.
- Has data flows, external entities, one process (system in focus) and no data stores.
- Shows the system boundary and interactions with external entities.

In this case:

External entity	- Student
Process	- Student Administration process application
Data Flows	- Application Form, - Confirmation/Rejection Letter

Context diagram



Student

Application Details

0

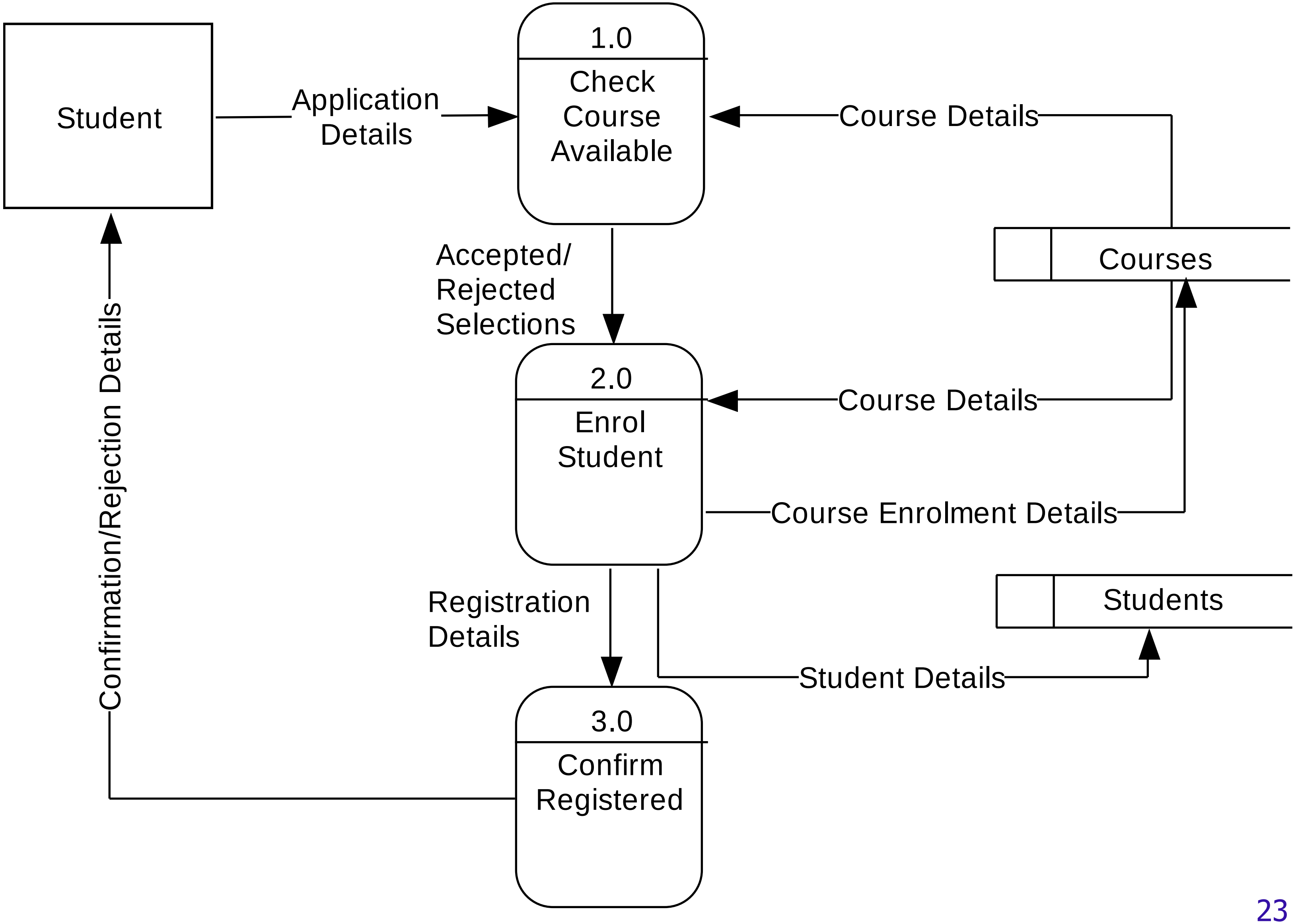
Student Administration System

Confirmation/Rejection Details

System/Level 0 DFD

External entity	- Student
Processes	- Check available
	- Enrol student
	- Confirm Registration
Data Flows	- Application Form
	- Course Details
	- Course Enrolment Details
	- Student Details
	- Confirmation/Rejection Letter
Data Stores	- Courses
	- Students

System/Level 0 DFD



Hoosier Burger's food ordering system

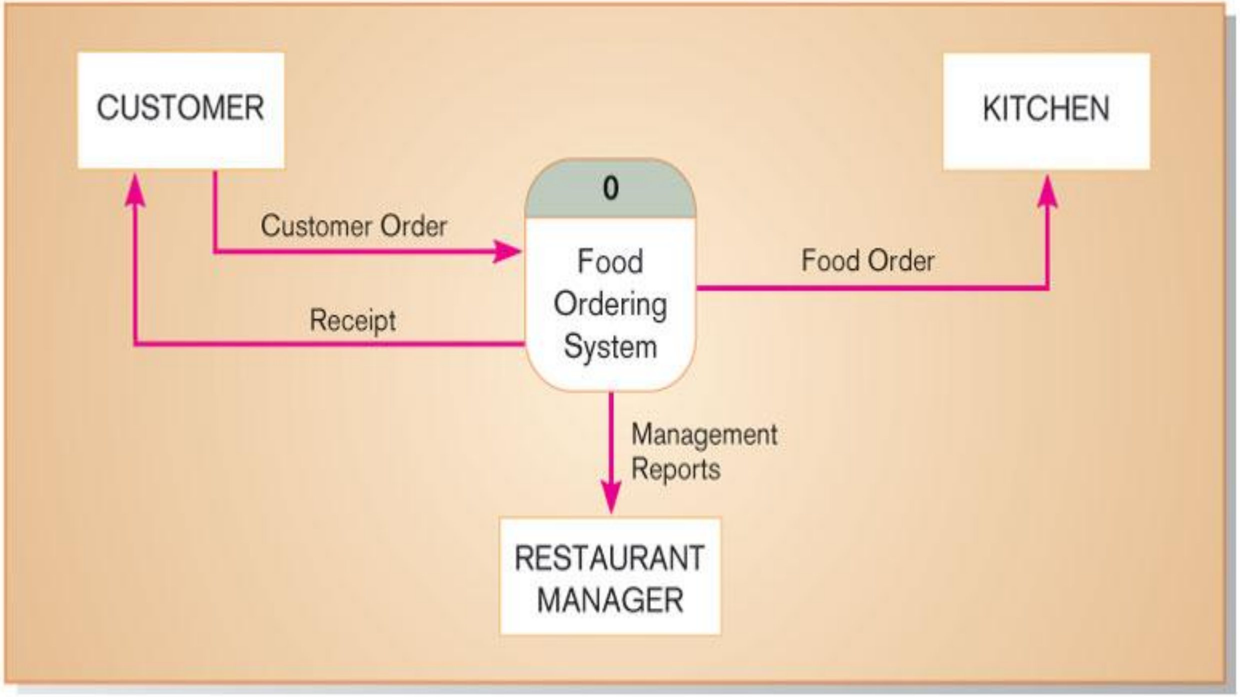
- The restaurant uses an information system that takes customer orders, sends the order to the kitchen, monitors the goods sold and inventory, and generates reports for management

Hoosier Burger's food ordering system

- The restaurant uses an information system that takes customer orders, sends the order to the kitchen, monitors the goods sold and inventory, and generates reports for management

Context Diagram

Figure 7-4 Context diagram of Hoosier Burger's food ordering system

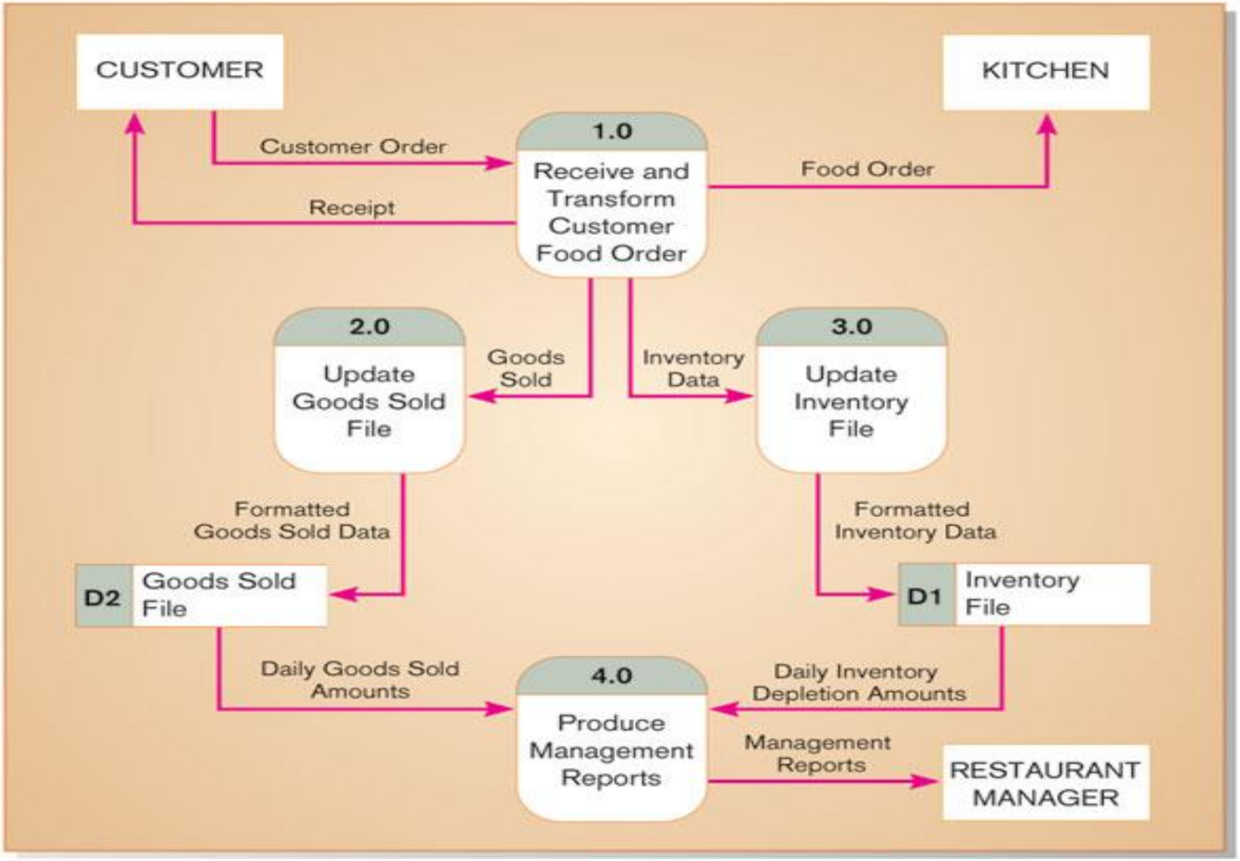


Context diagram shows the system boundaries, external entities that interact with the system, and major information flows between entities and the system.

NOTE: only one process symbol, and no data stores shown.

Level-0 DFD

Figure 7-5 Level-0 DFD of Hoosier Burger's food ordering system



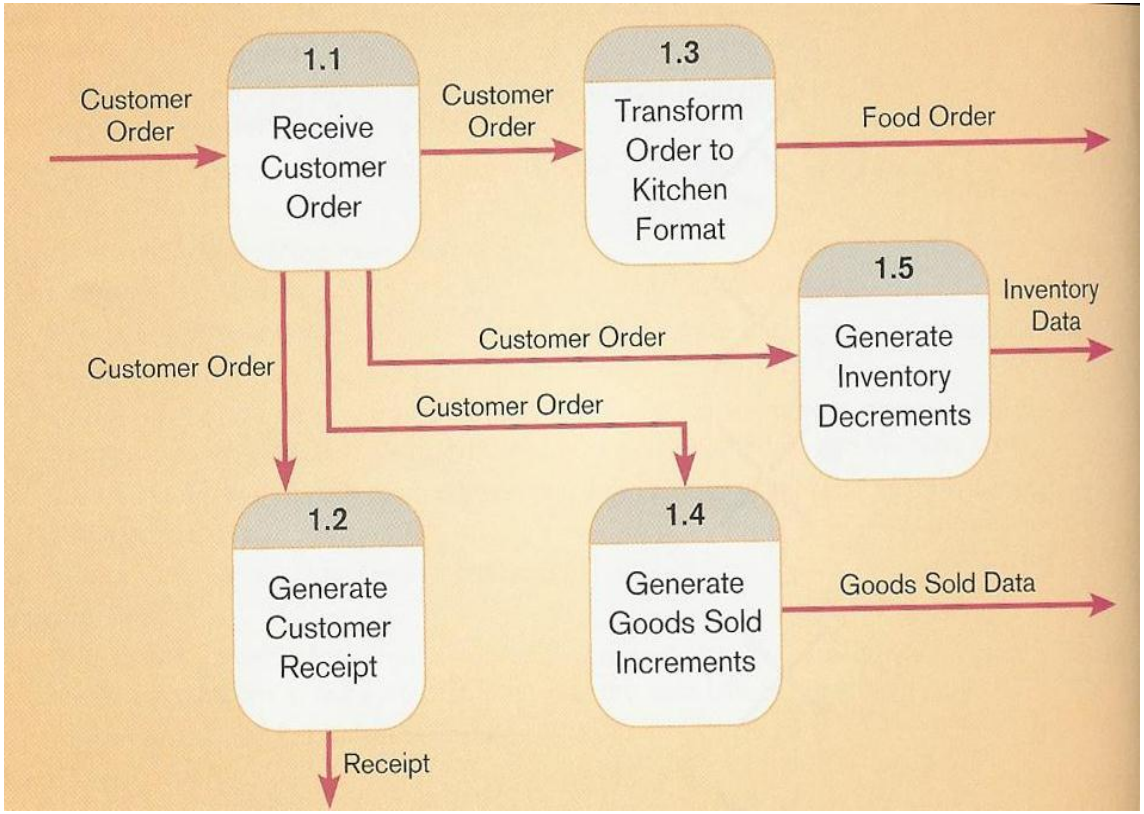
Level-0 DFD shows the system's major processes, data flows, and data stores at a high level of abstraction.

Processes are labeled 1.0, 2.0, etc. These will be decomposed into more primitive (lower-level) DFDs.

Functional Decomposition

- An iterative process of breaking a system description down into finer and finer detail
- High-level processes described in terms of lower-level sub-processes
- DFD charts created for each level of detail

Level-1 DFD of Receive & Transform Customer Food Order



Level-1 DFD shows the sub-processes of one of the processes in the Level-0 DFD.

This is a Level-1 DFD for Process 1.0.

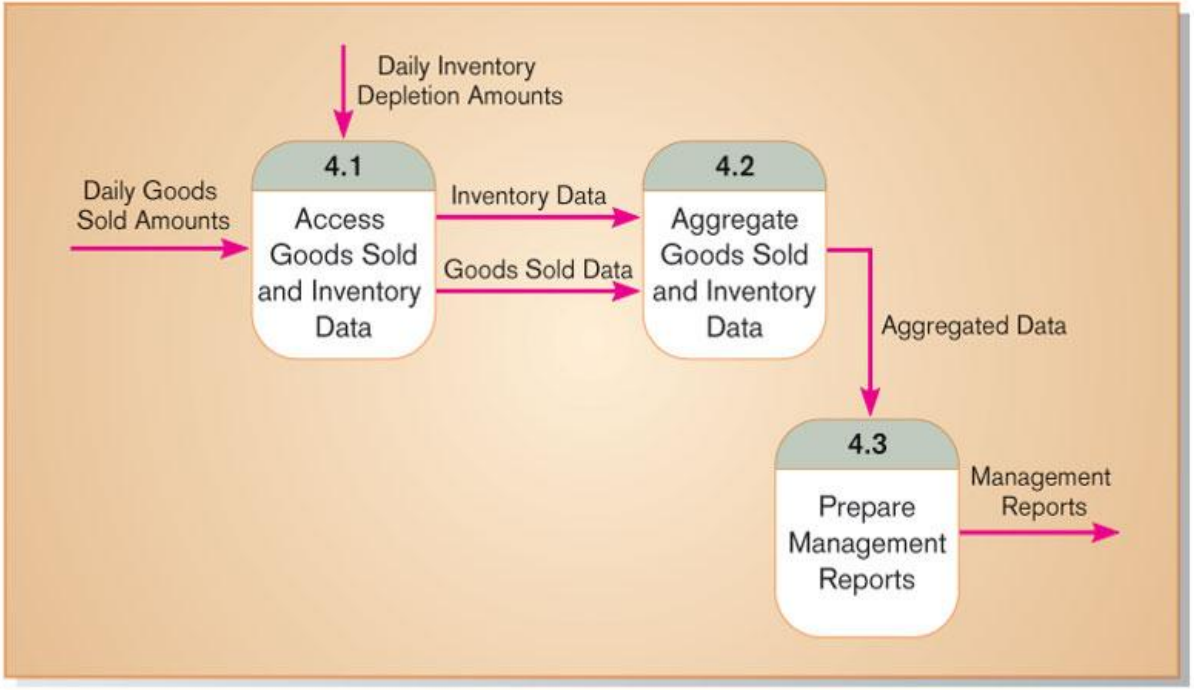
No sources or sinks Are represented.

Processes are labeled 1.1, 1.2, etc. These can be further decomposed in more primitive (lower-level) DFDs if necessary.

Decomposition of DFD

Level-1 DFD

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


Daily Inventory Depletion Amounts
Daily Goods Sold Amounts
4.1
Access Goods Sold and Inventory Data
Inventory Data
Goods Sold Data
4.2
Aggregate Goods Sold and Inventory Data
Aggregated Data
4.3
Prepare Management Reports
Management Reports

Level-1 DFD shows the sub-processes of one of the processes in the Level-0 DFD.

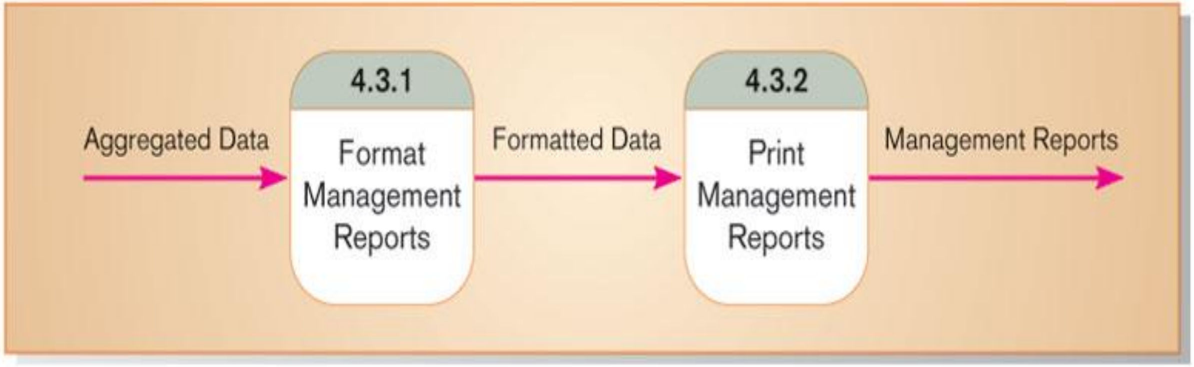
This is a Level-1 DFD for Process 4.0.

No sources or sinks Are represented.

Processes are labeled 4.1, 4.2, etc. These can be further decomposed in more primitive (lower-level) DFDs if necessary.

Level-*n* DFD

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



Level-*n* DFD shows the sub-processes of one of the processes in the Level *n-1* DFD.

This is a Level-2 DFD for Process 4.3.

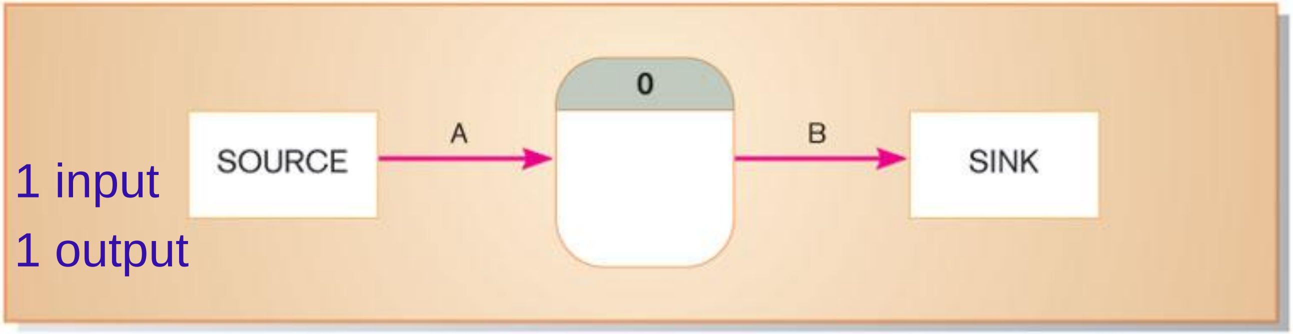
Processes are labeled 4.3.1, 4.3.2, etc. If this is the lowest level of the hierarchy, it is called a *primitive DFD.*

DFD Balancing

- The conservation of inputs and outputs to a data flow process when that process is decomposed to a lower level
- Balanced means:
    - Number of inputs to lower level DFD equals number of inputs to associated process of higher-level DFD
    - Number of outputs to lower level DFD equals number of outputs to associated process of higher-level DFD

Unbalanced DFD

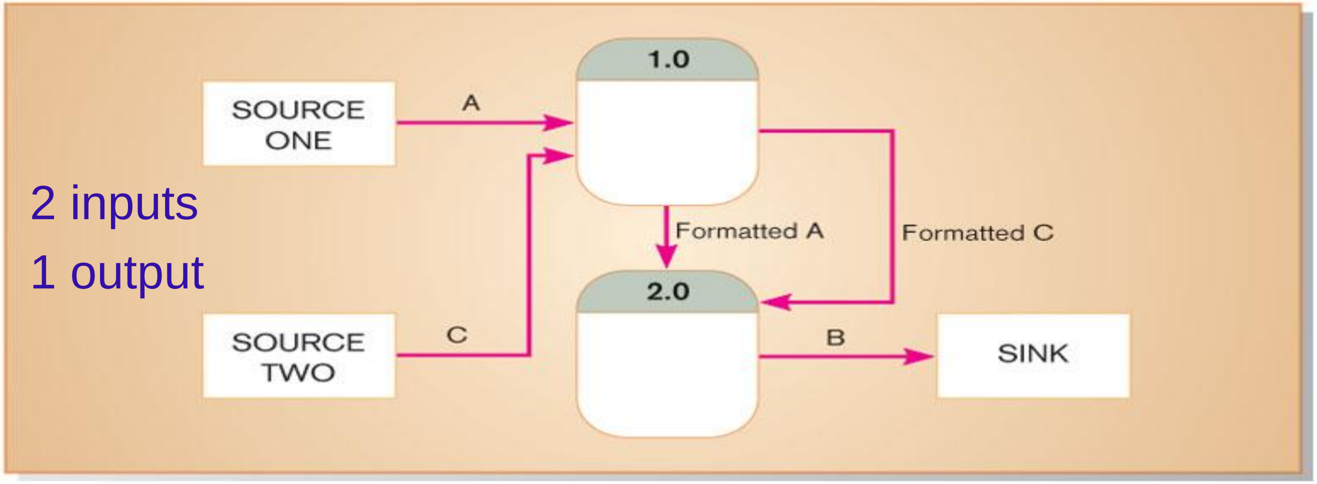
Figure 7-10a An unbalanced set of data flow diagrams - Context diagram



1 input

1 output

Figure 7-10b An unbalanced set of data flow diagrams - Level-0 diagram



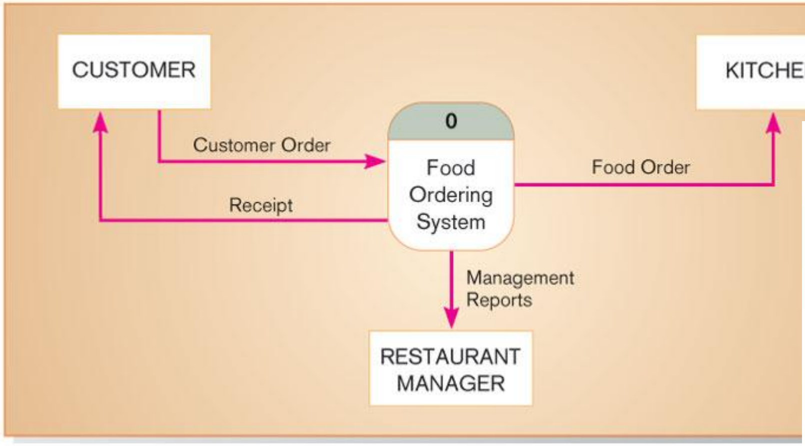
2 inputs

1 output

This is unbalanced because the process of the context diagram has only one input but the Level-0 diagram has two inputs.

Balanced DFD

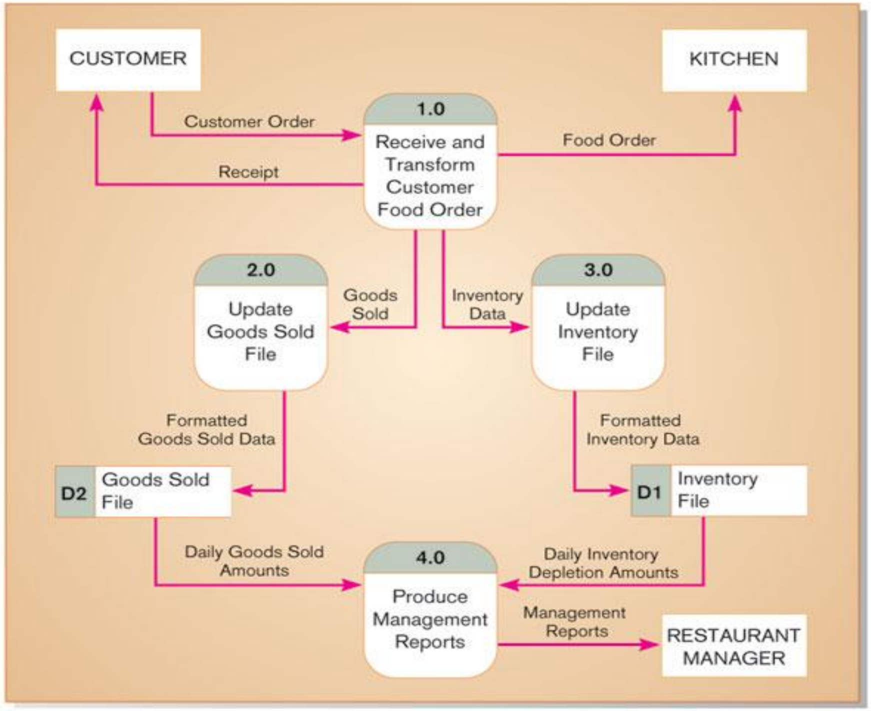
Figure 7-4 Context diagram of Hoosier Burger's food ordering system



1 input

2 outputs

Figure 7-5 Level-0 DFD of Hoosier Burger's food ordering system

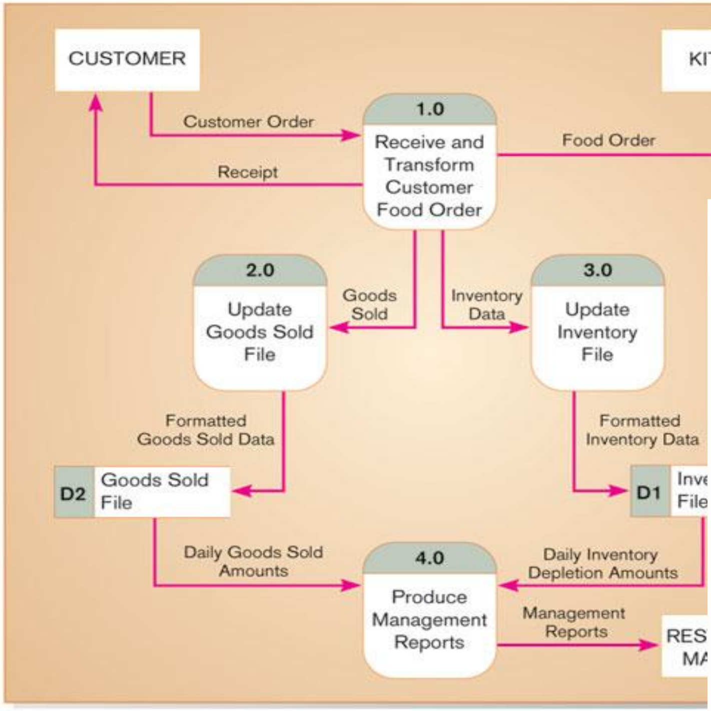


These are balanced because the numbers of inputs and outputs of context diagram process equal the number of inputs and outputs of Level-0 diagram.

Balanced DFD (cont.)

These are balanced because the numbers of inputs and outputs to Process 1.0 of the Level-0 diagram equals the number of inputs and outputs to the Level-1 diagram.

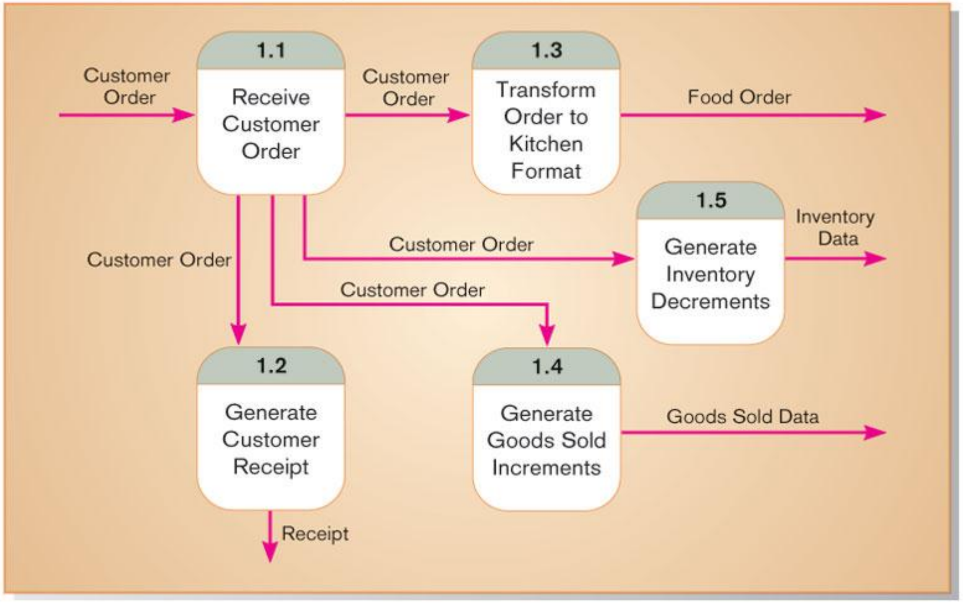
Figure 7-5 Level-0 DFD of Hoosier Burger's food ordering system



1 input

4 outputs

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



Data Flow Splitting

Figure 7-11a Example of data flow splitting - Composite data flow

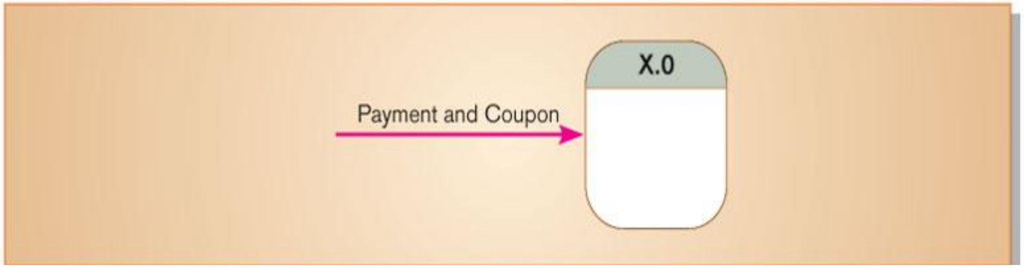
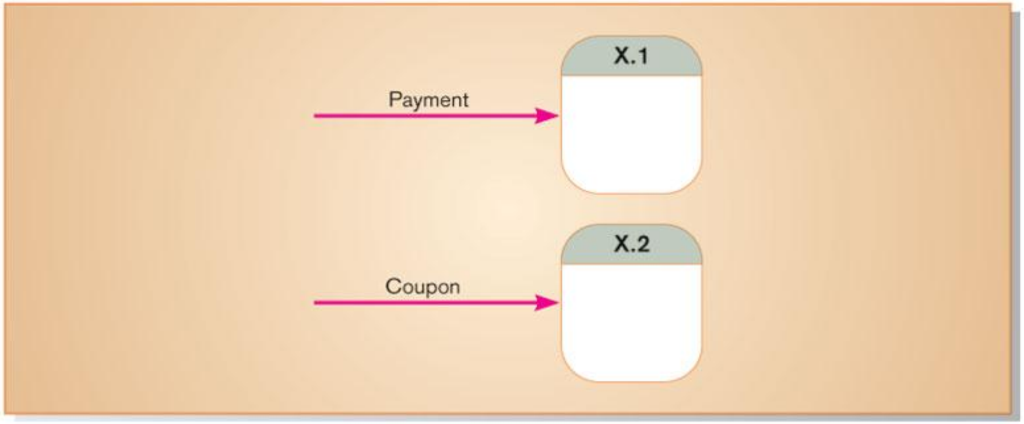


Figure 7-11b Example of data flow splitting - Disaggregated data flows



A composite data flow at a higher level may be split if different parts go to different processes in the lower level DFD.

This remains balanced because the same data is involved, but split into two parts.

More DFD Rules

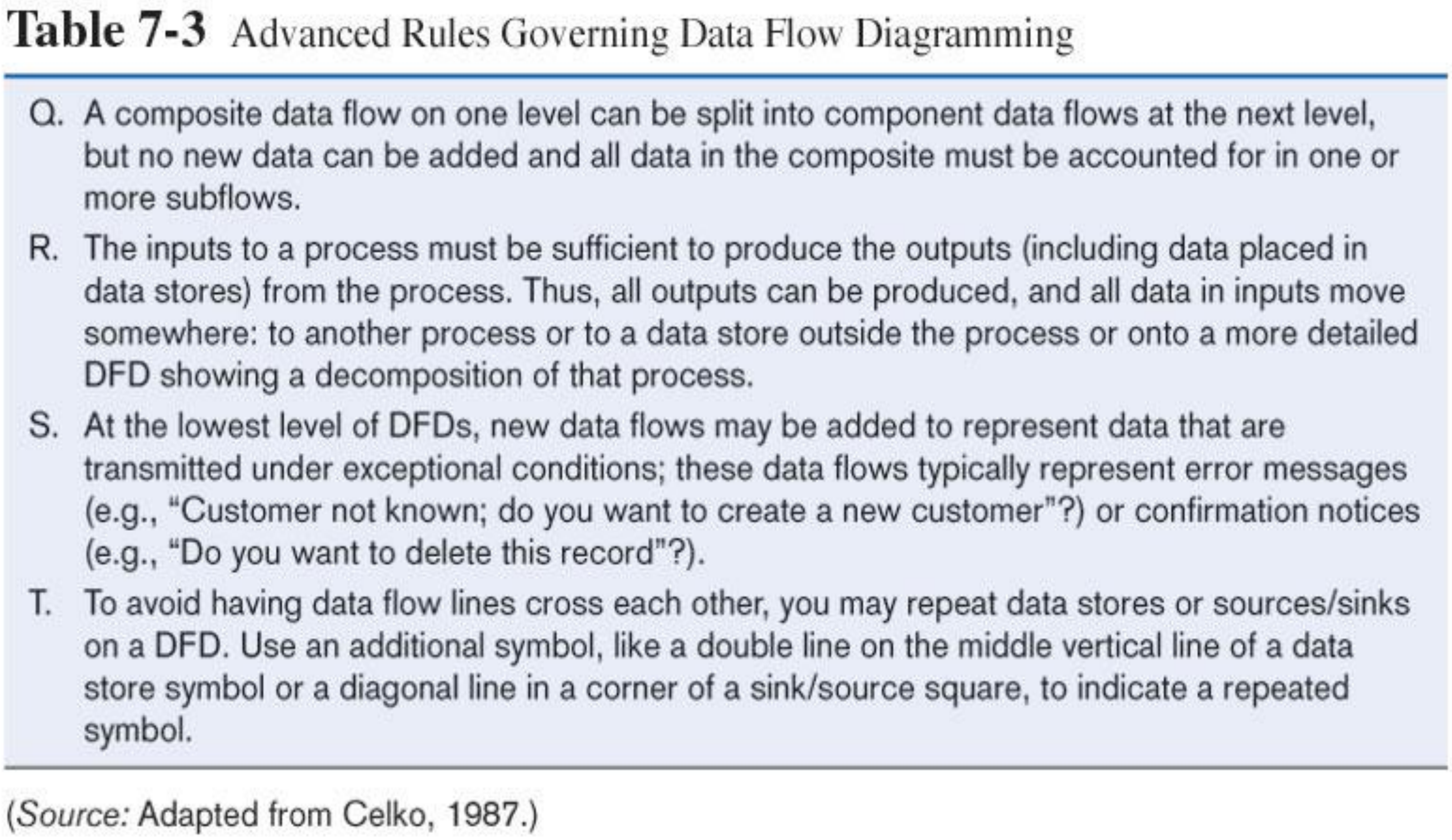


Table 7-3 Advanced Rules Governing Data Flow Diagramming

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:* Adapted from Celko, 1987.)

Four Different Types of DFDs

- **Current Physical**
  - Process labels identify technology (people or systems) used to process the data.
  - Data flows and data stores identify actual name of the physical media.
- **Current Logical**
  - Physical aspects of system are removed as much as possible.
  - Current system is reduced to data and processes that transform them.

Four Different Types of DFDs (Cont.)

- **New Logical**
  - Includes additional functions.
  - Obsolete functions are removed.
  - Inefficient data flows are reorganized.
- **New Physical**
  - Represents the physical implementation of the new system.

Guidelines for Drawing DFDs

- Completeness
  - DFD must include all components necessary for system.
  - Each component must be fully described in the project dictionary or CASE repository.
- Consistency
  - The extent to which information contained on one level of a set of nested DFDs is also included on other levels.

Guidelines for Drawing DFDs (cont.)

- Timing
  - Time is not represented well on DFDs.
  - Best to draw DFDs as if the system has never started and will never stop.
- Iterative Development
  - Analyst should expect to redraw diagram several times before reaching the closest approximation to the system being modeled.

Guidelines for Drawing DFDs (cont.)

- Primitive DFDs
  - Lowest logical level of decomposition
  - Decision has to be made when to stop decomposition

Guidelines for Drawing DFDs (cont.)

- Rules for stopping decomposition
  - When each process has been reduced to a single decision, calculation or database operation (ex. Retrieve, create, delete…).
  - When each data store represents data about a single entity (ex. Customer, employee, order…).
  - When the system user does not care to see any more detail or when the analyst documented sufficient detail.

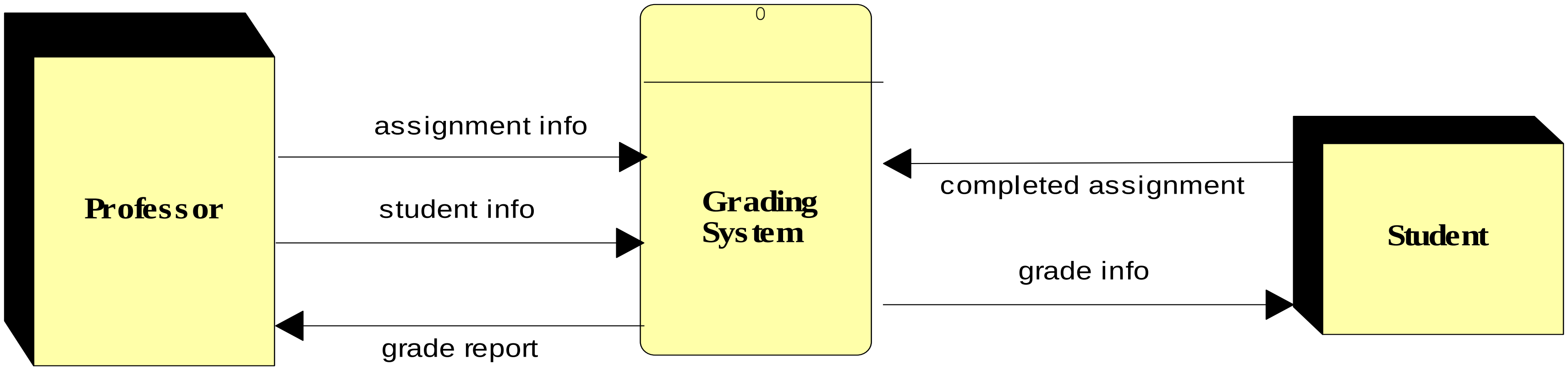
Guidelines for Drawing DFDs (cont.)

- Rules for stopping decomposition (continued)
  - When every data flow does not need to be split further to show that data are handled in various ways
  - When you believe that you have shown each business form or transaction, online display and report as a single data flow
  - When you believe that there is a separate process for each choice on all lowest-level menu options

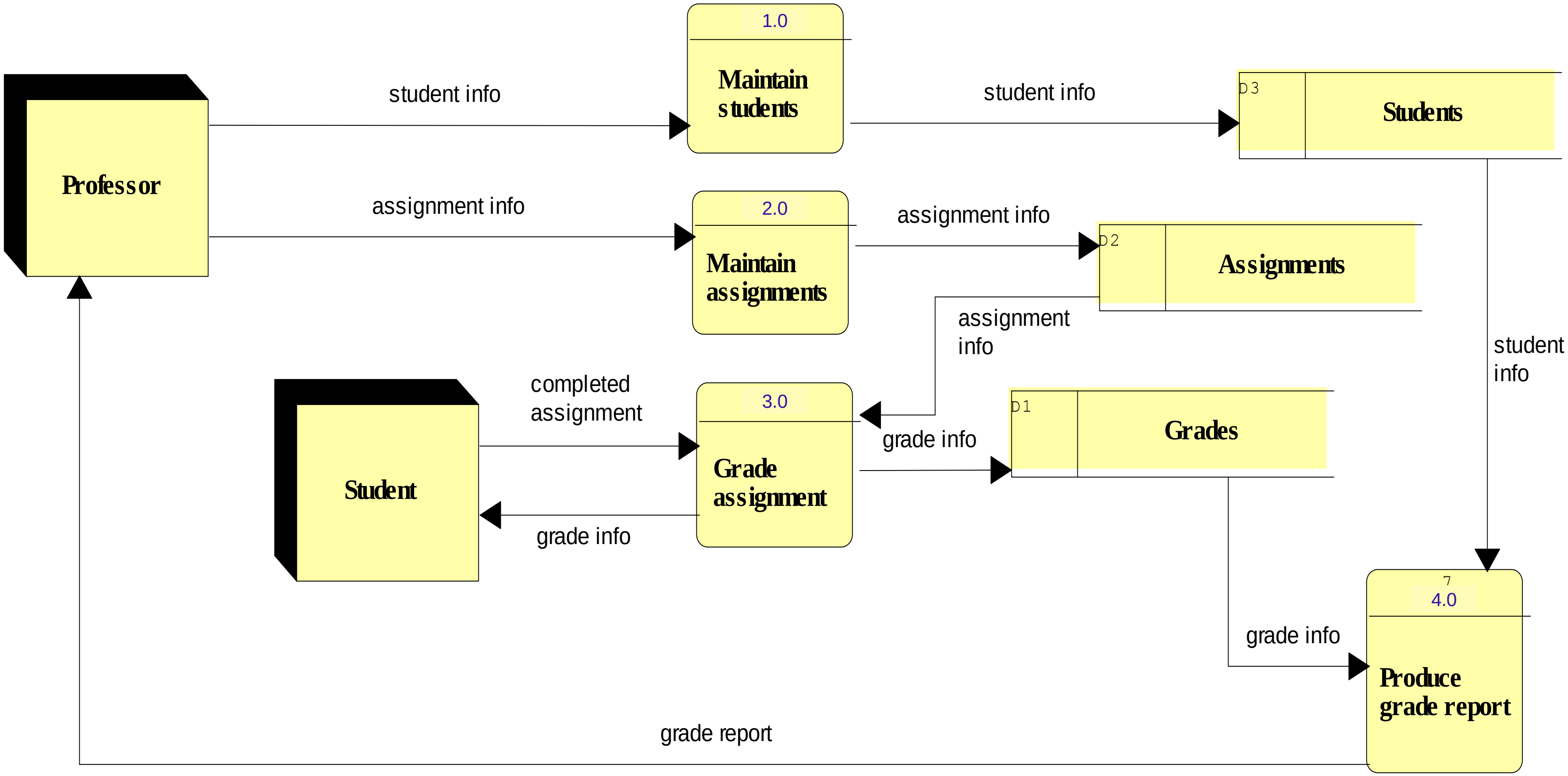
Example:Grading System

- Al Burns is the professor for the class. He gives the student information to Ms. Grader to create a file of them. He also gives a list of assignments, a description for each assignment, and a grading key for each one to Ms. Grader to create a file of them.
- Students complete the assignments and give them to Ms. Grader for grading. Ms. Grader grades the assignments based on a key provided by Al Burns, and assigns a score (grade) from 0 to 5. She then records the scores for the students on the grades file and gives each student his\her grade.
- Finally, Ms. Grader prints a report for Al Burns contains students with their grades.
- **Draw a context diagram and level 0 diagram for the Grading System.**

Context diagram



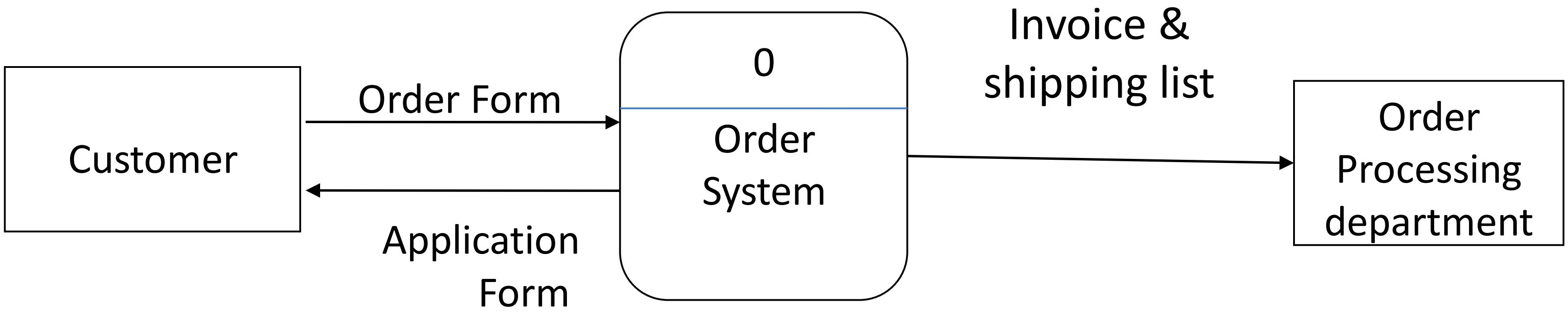
Level 0


Professor
student info
assignment info
1.0
Maintain students
student info
D 3
Students
2.0
Maintain assignments
assignment info
D 2
Assignments
assignment info
student info
Student
completed assignment
grade info
3.0
Grade assignment
grade info
D 1
Grades
7
4.0
Produce grade report
grade info
grade report

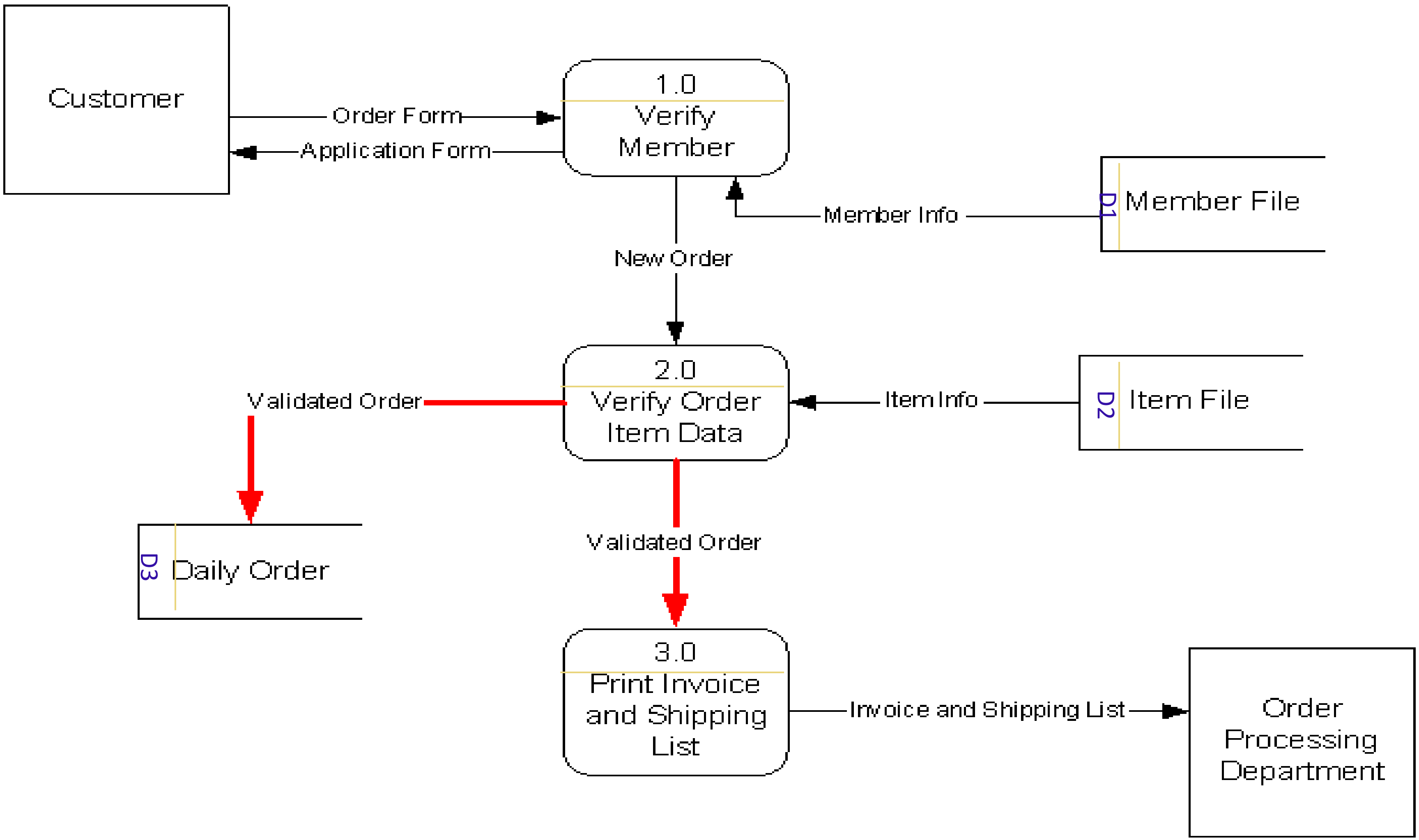
Example: FOX company

- FOX company is a mail order company that distributes CDs and tapes at discount prices to members.
- When a customer place an order form, the clerk verifies that the sender is a member by checking the MEMBER FILE. If the sender is not a member, the clerk returns the order along with a membership application form. If the customer is a member, the clerk verifies the order item data by checking the ITEM FILE. Then the clerk enters the order data and saves it to the DAIILY ORDERS FILE. At the same time the clerk also prints an invoice and shipping list for each order, which are forwarded to the order processing department for processing there.

Context Diagram



Level-0 DFD


Customer
Order Form
Application Form
1.0
Verify Member
D1
Member File
Member Info
New Order
2.0
Verify Order Item Data
Item Info
D2
Item File
Validated Order
D3
Daily Order
Validated Order
3.0
Print Invoice and Shipping List
Invoice and Shipping List
Order Processing Department