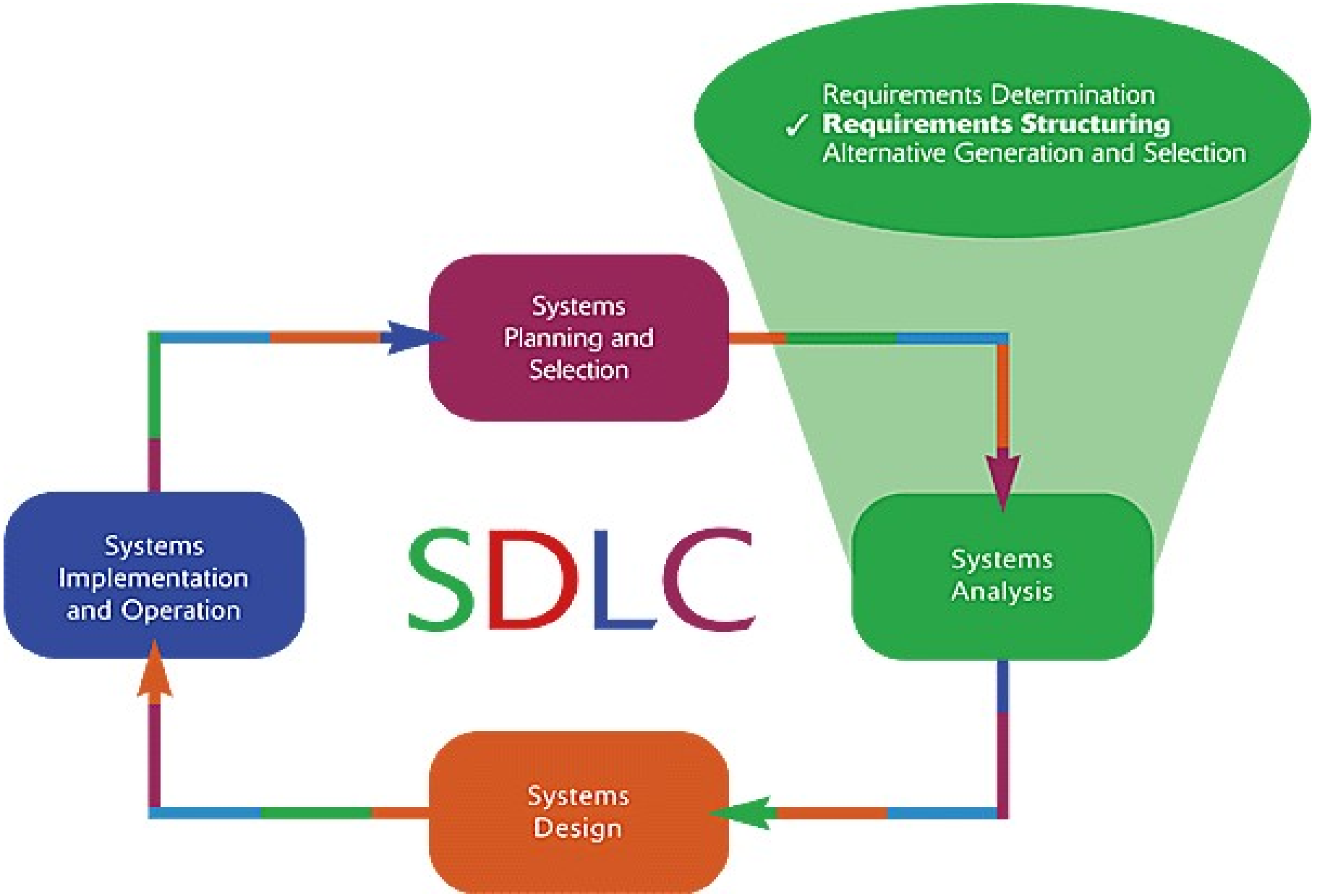
Process Modeling
and
Data Flow Diagrams

Organised by

Software Engineering Team

CSED


Requirements Determination
✓ Requirements Structuring
Alternative Generation and Selection
Systems Planning and Selection
Systems Analysis
Systems Design
Systems Implementation and Operation
SDLC

Learning Objectives

- Understand the logical modeling of processes through studying data flow diagrams
- How to draw data flow diagrams using rules and guidelines
- How to decompose data flow diagrams into lower-level diagrams
- Balancing of data flow diagrams

Process Modeling

- Modeling a system's process
  - Utilize information gathered during requirements determination
  - Structure of the data is also modeled in addition to the processes
- Graphically represent the processes that capture, manipulate, store and distribute data between a system and its environment and among system components
- Data flow diagrams (DFD)
  - Graphically illustrate movement of data between external entities and the processes and data stores within a system

Process Modeling: Deliverables and Outcomes

- Set of coherent, interrelated data flow diagrams
- Context data flow diagram (DFD)
  - Scope of system
- DFDs of current system
  - Enables analysts to understand current system
- DFDs of new logical system
  - Technology independent
  - Show data flows, structure and functional requirements of new system
- Project dictionary and CASE repository

Data Flow Diagramming Mechanics

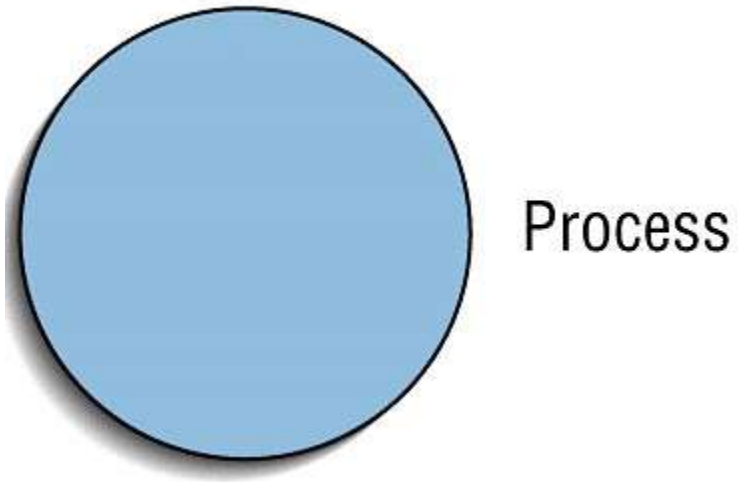
- Drawn as an arrow
- Depicts data that are in motion and moving as a unit from one place to another in the system.
- Select a meaningful name to represent the data

Data Flow Diagramming Mechanics: Data Store

- Drawn as two horizontal parallel lines
- 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

Data Flow Diagramming Mechanics: Process

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



Data Flow Diagramming Mechanics: Source/Sink

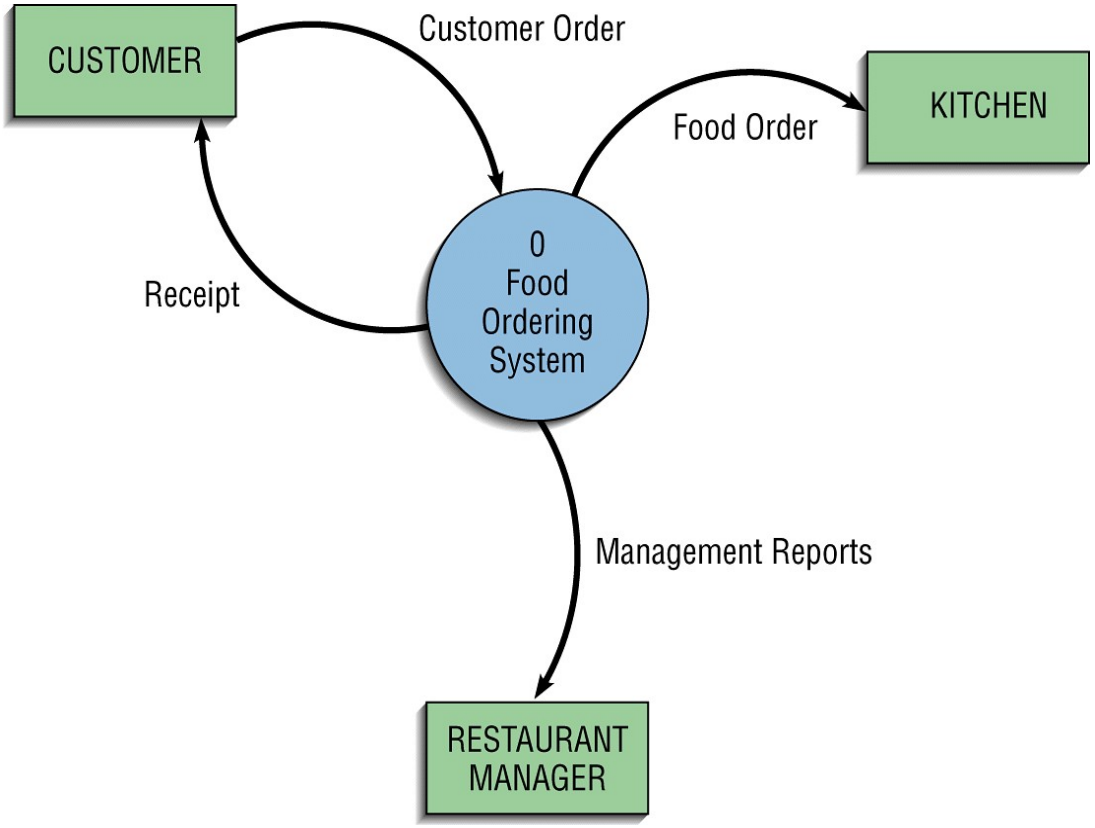
- Drawn as a square symbol
- Depicts the origin and/or destination of the data
- Sometimes referred to as an external entity
- Because they are external, many characteristics are not of interest to us

Source/Sink

Data Flow Diagramming Definitions: Context Diagram

A data flow diagram (DFD) of the scope of an organizational system that shows the

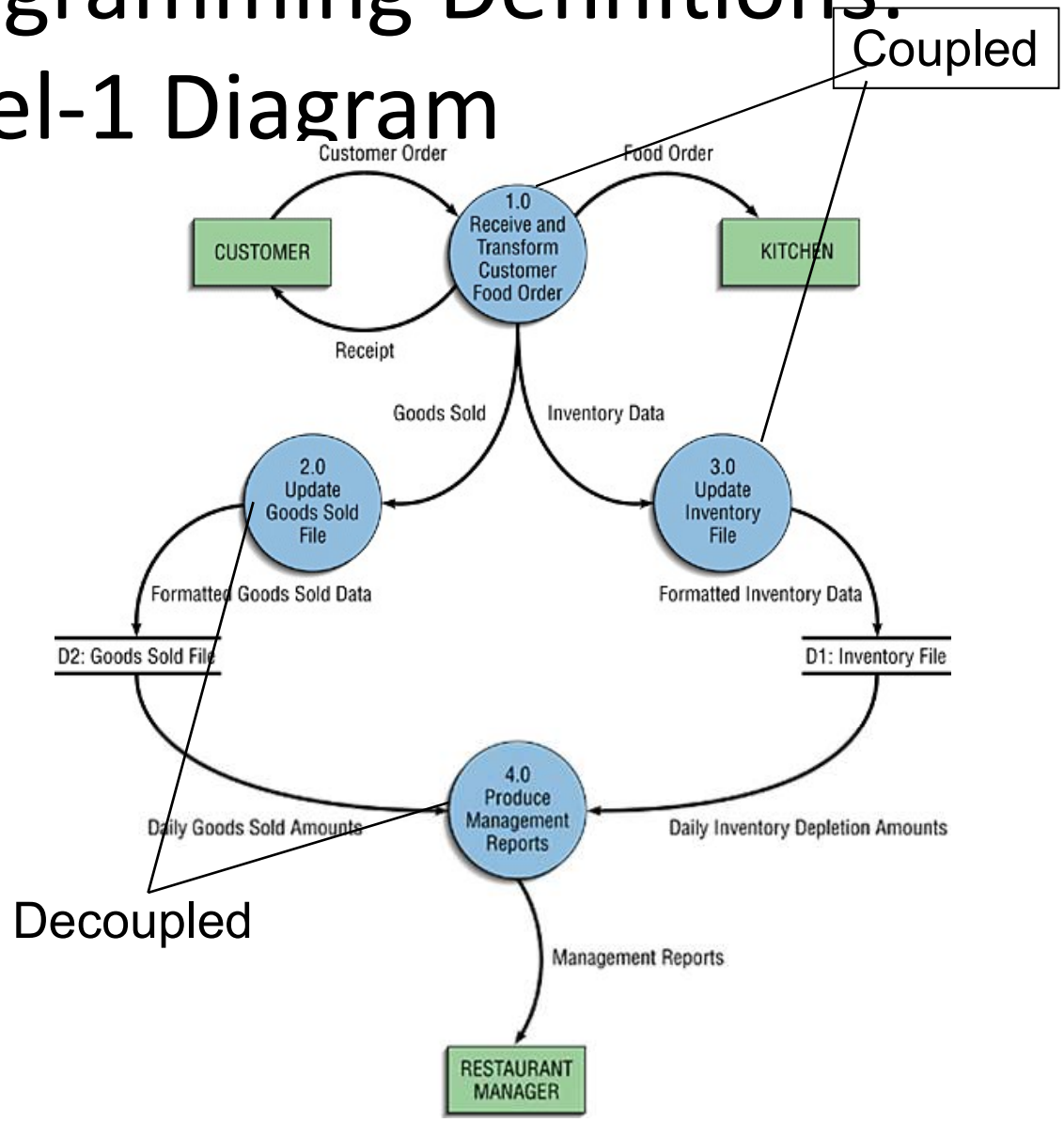
- system boundaries
- external entities that interact with the system and
- the major information flows between the entities and the system



Data Flow Diagramming Definitions: Level-1 Diagram

A data flow diagrams (DFD) that represents

- system's major processes
- data flows
- data stores at a higher level

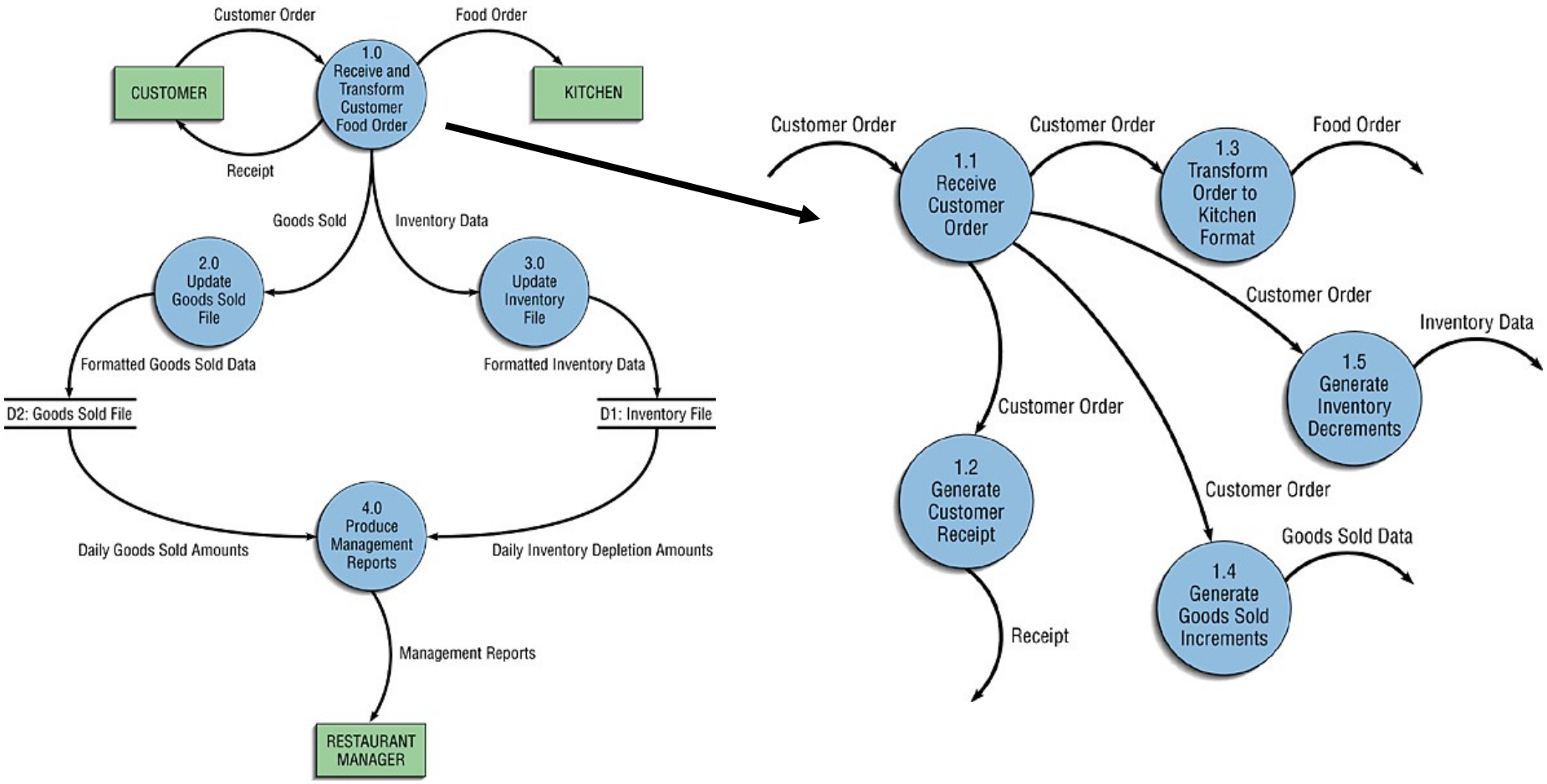


Decomposition of DFDs

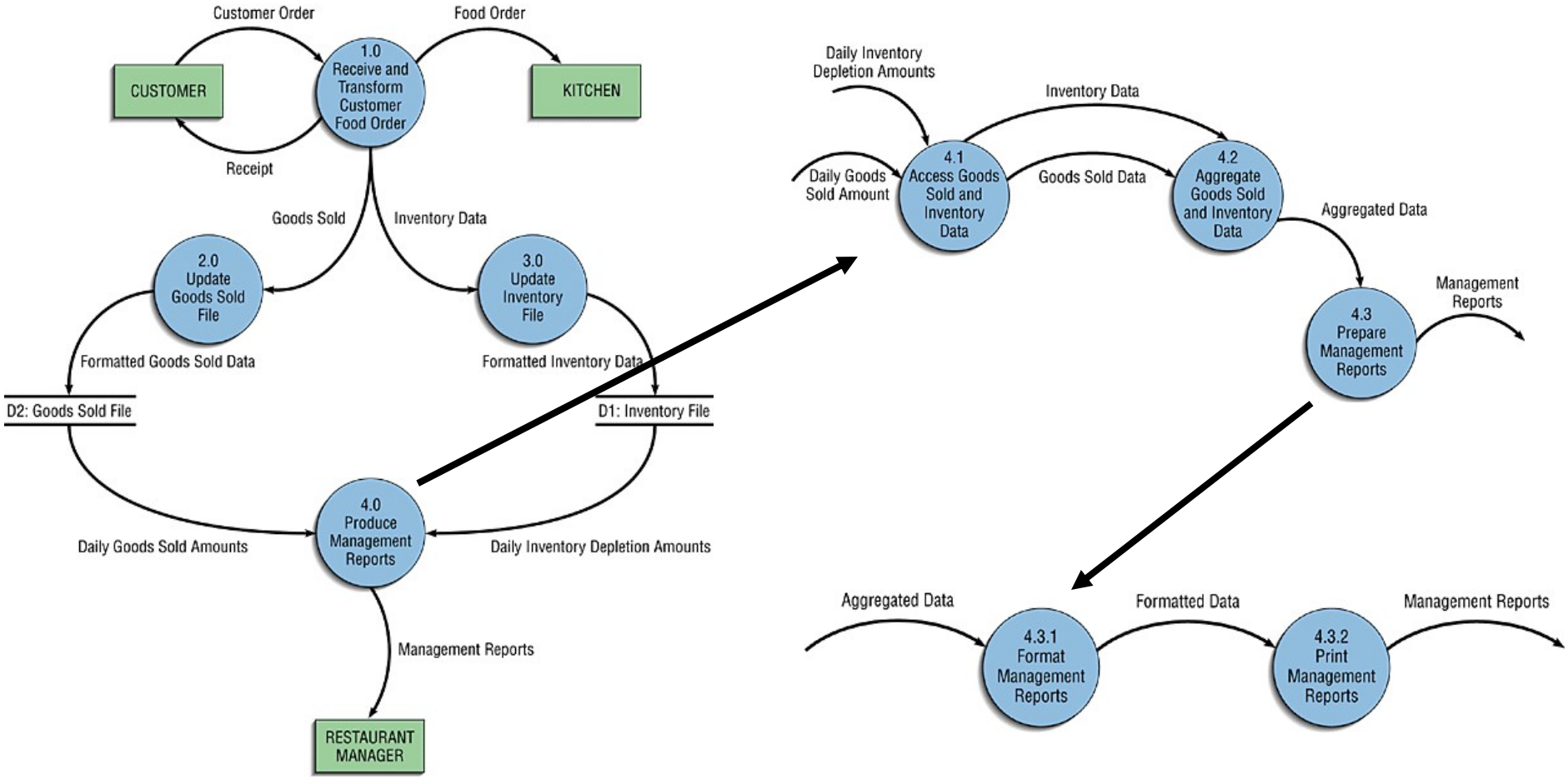
- Functional decomposition
  - Act of going from one single system to many component processes
  - Repetitive procedure
  - Lowest level is called a primitive DFD

- Level-N Diagrams
  - A DFD that is the result of *n* nested decompositions of a series of subprocesses from a process on a level-0 diagram

Decomposition of DFDs

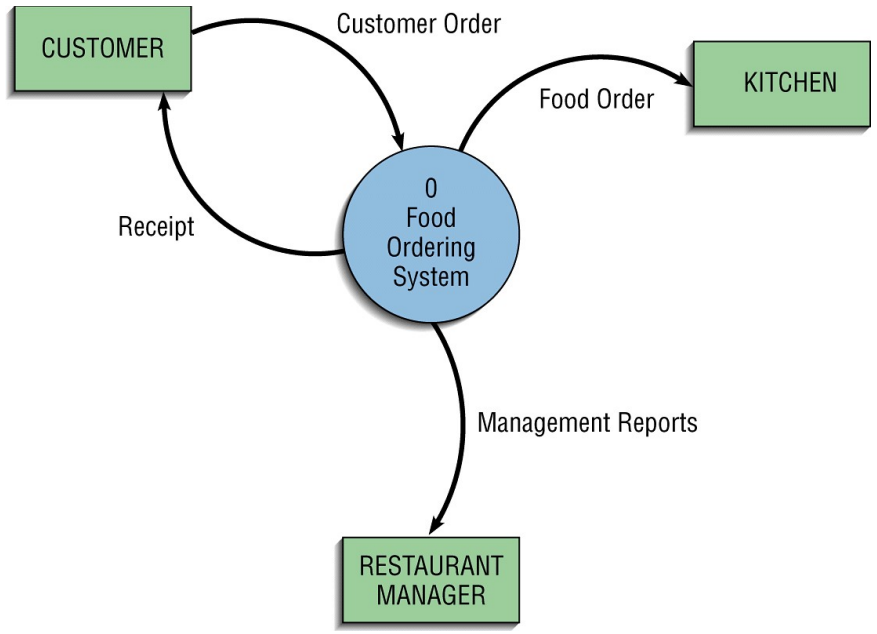


Decomposition of DFDs



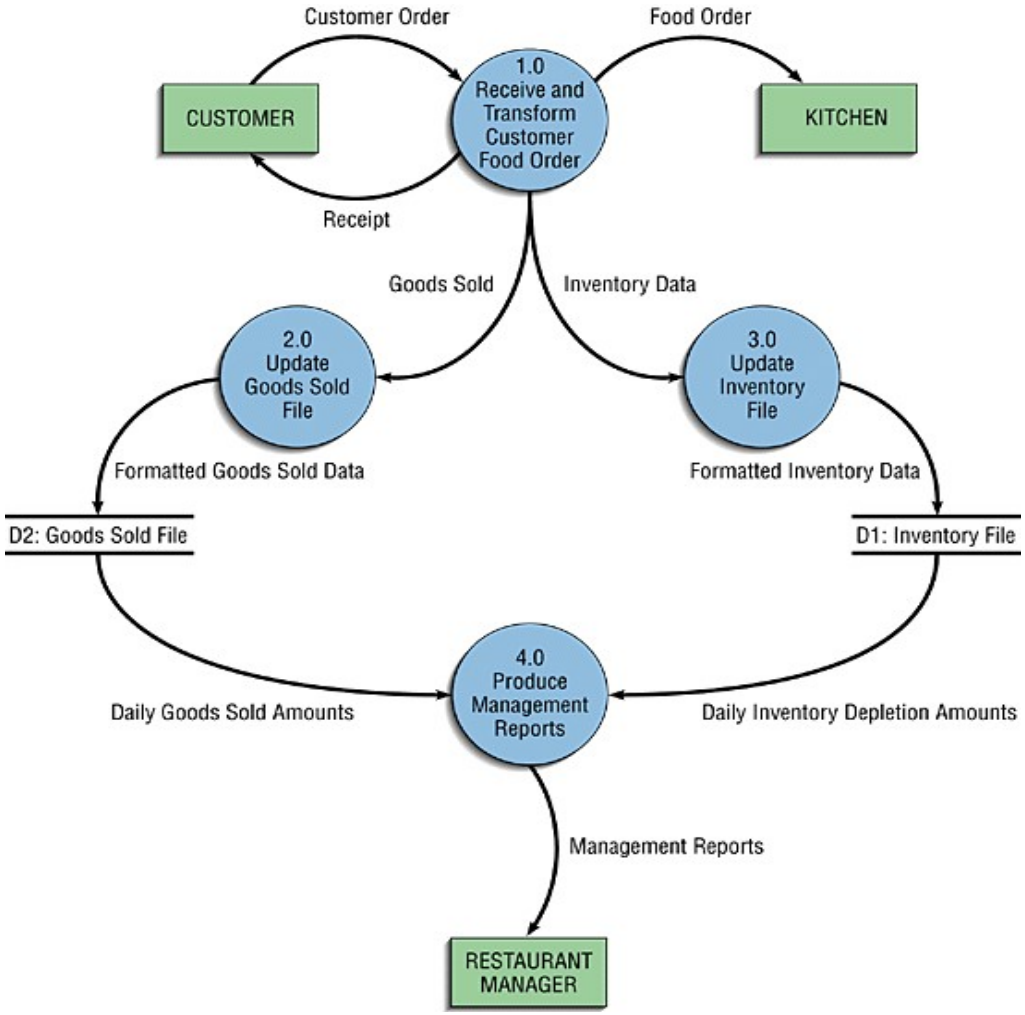
Balancing DFDs

- When decomposing a DFD, you must conserve inputs to and outputs from a process at the next level of decomposition
- This is called balancing
- Example: King's Burgers
  - One input to the system
    - Customer order
  - Three outputs:
    - Customer receipt
    - Food order
    - Management reports



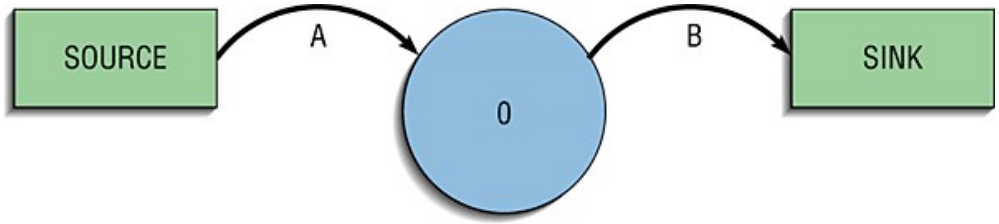
Balancing DFDs

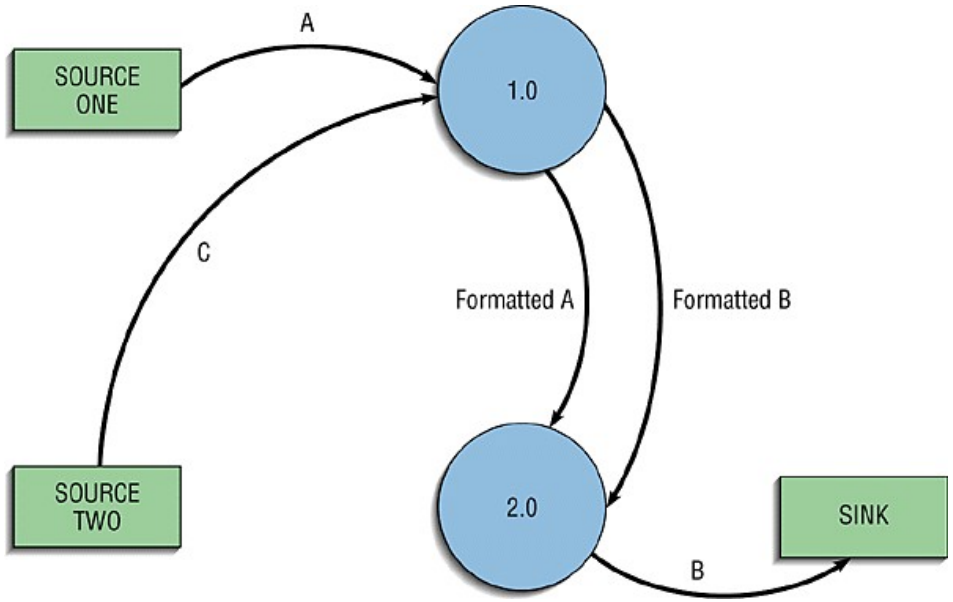
- Here we have the same inputs and output, that is no new inputs or outputs have been introduced
- We can say that the context diagram and level-1 DFD are balanced



Balancing DFDs: An unbalanced example

- In context diagram, we have one input to the system, A and one output, B
  - Level 1 diagram has one additional data flow, C



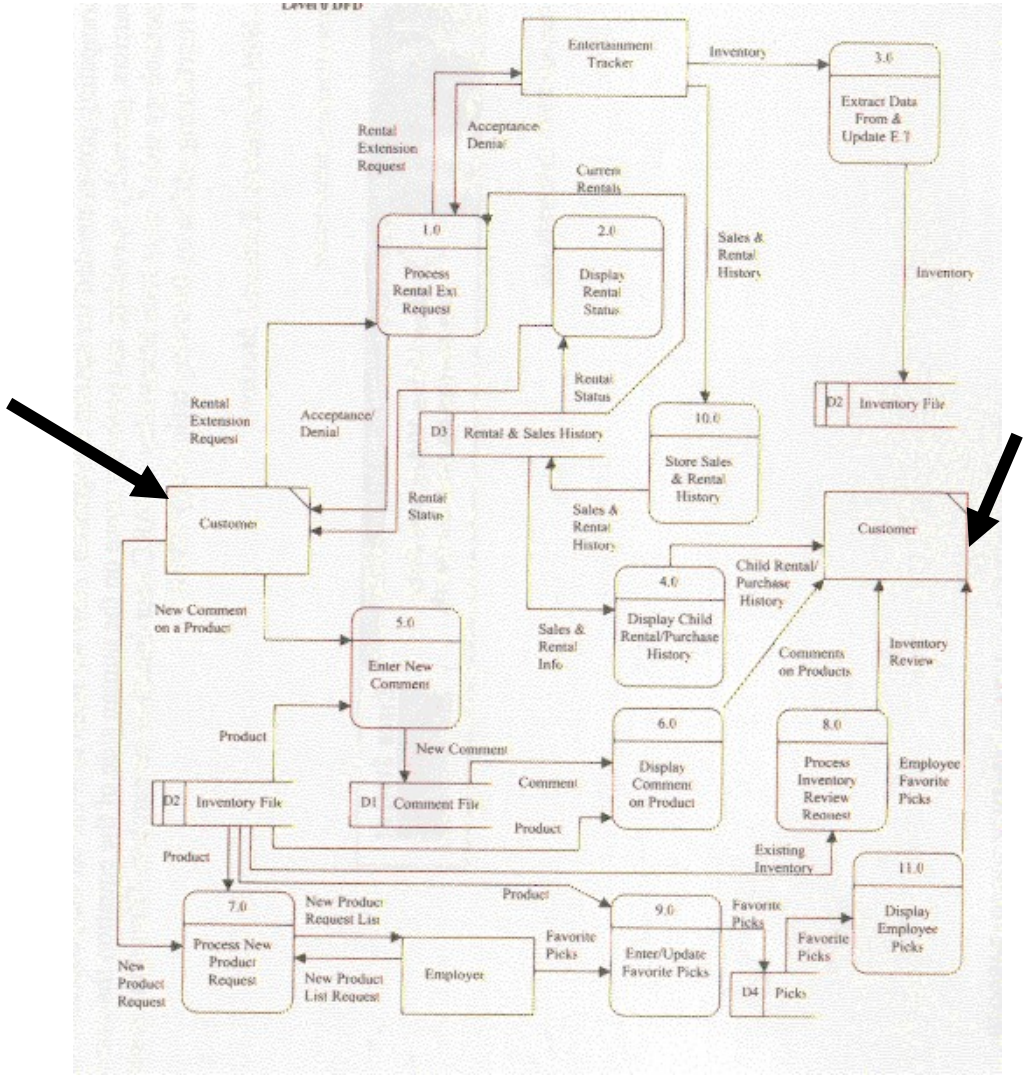


Data Flow Diagramming Rules

Basic rules that apply to all DFDs

- Inputs to a process are always different than outputs

- Objects always have a unique name
  - In order to keep the diagram uncluttered, you can repeat data stores and data flows on a diagram

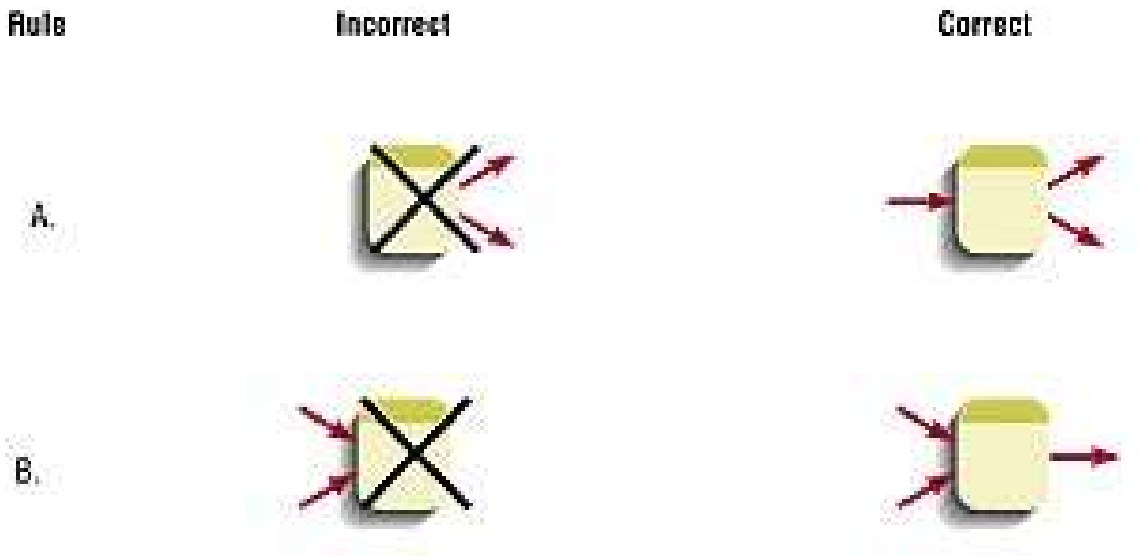


Data Flow Diagramming Rules: Process

A. No process can have only outputs (a miracle)

B. No process can have only inputs (black hole)

C. A process has a verb phrase label (except for context diagram)



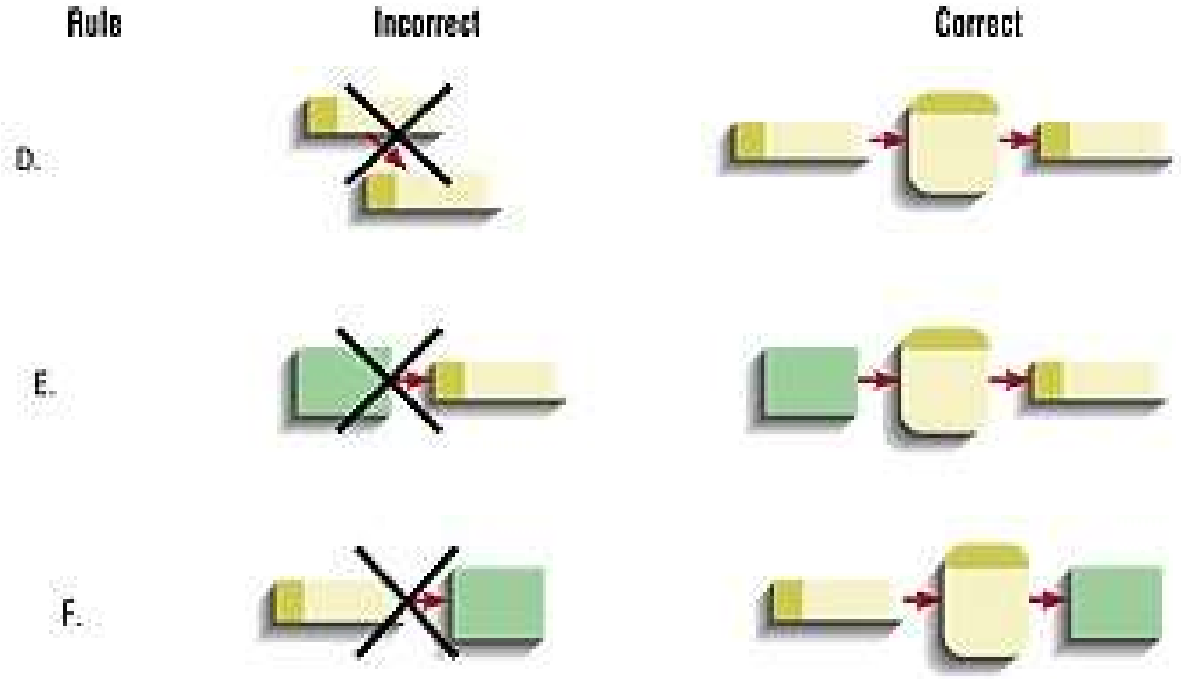
Data Flow Diagramming Rules: Data Store

D. Data cannot be moved from one store to another.

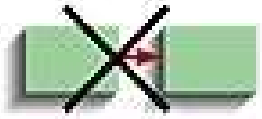
E. Data cannot move from an outside source to a data store

F. Data cannot move directly from a data store to a data sink

G. Data store has a noun phrase label



Data Flow Diagramming Rules: Source/Sink

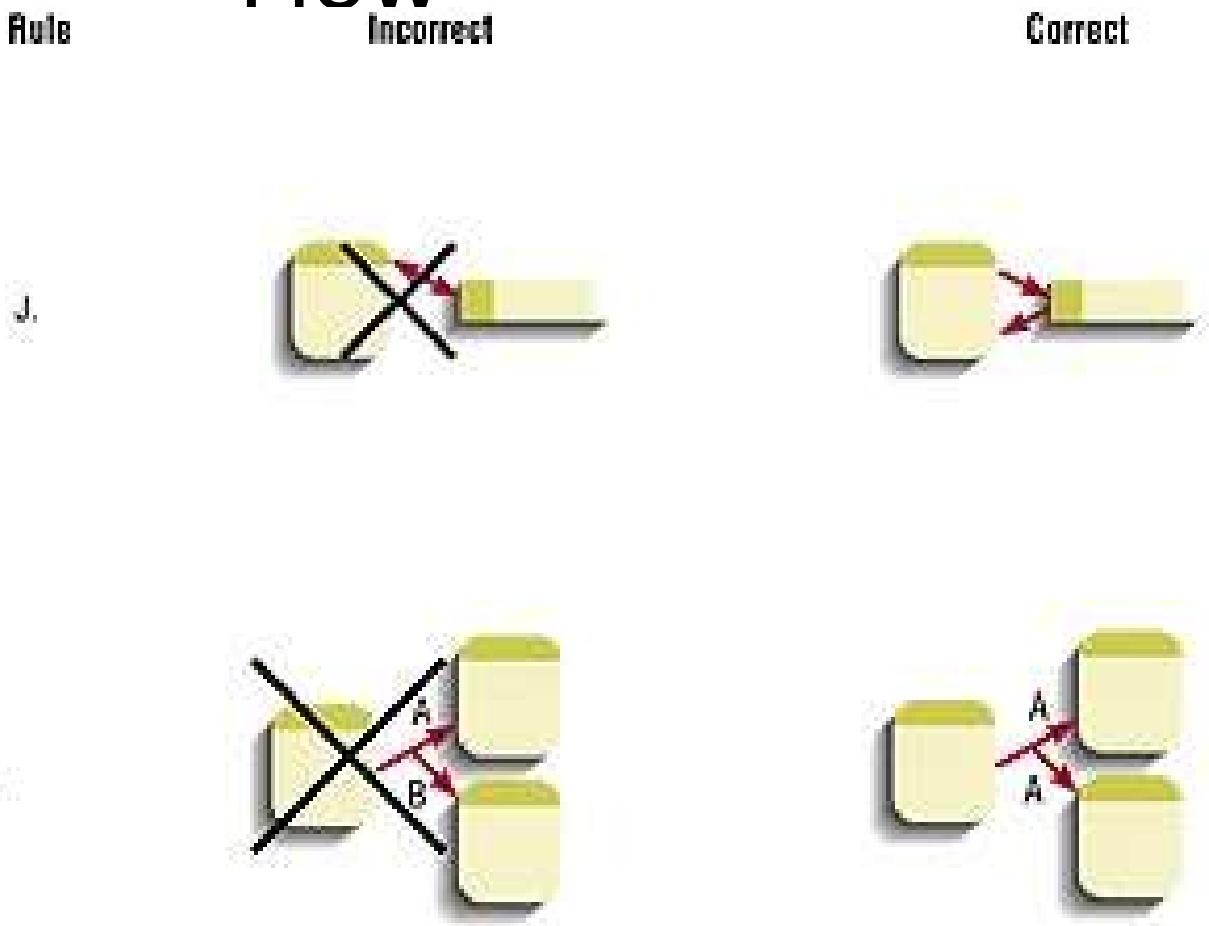
Rule	Incorrect	Correct
H. Data cannot move directly from a source to a sink	H. 	

I. A source/sink has a noun phrase label

Data Flow Diagramming Rules: Data Flow

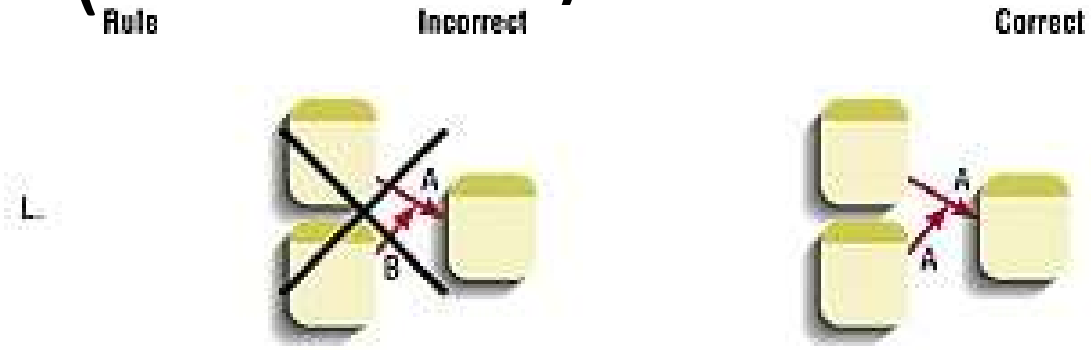
J. A data flow has only one direction of flow between symbols.

K. A fork means that exactly the same data go from a common location to two or more processes, data stores or sources/sinks



Data Flow Diagramming Rules: Data Flow (Continued)

L. A join means that exactly the same data come from any two or more different processes, data stores or sources/sinks to a common location



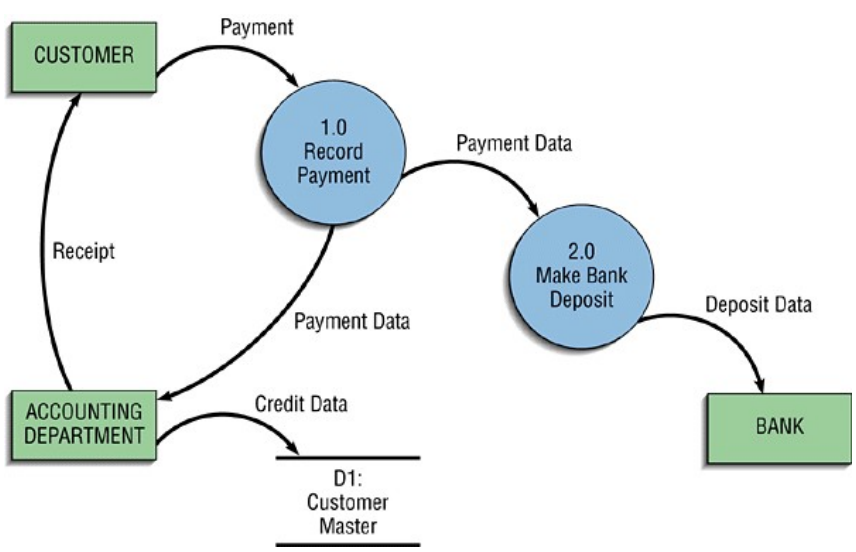
M. A data flow cannot go directly back to the same process it leaves

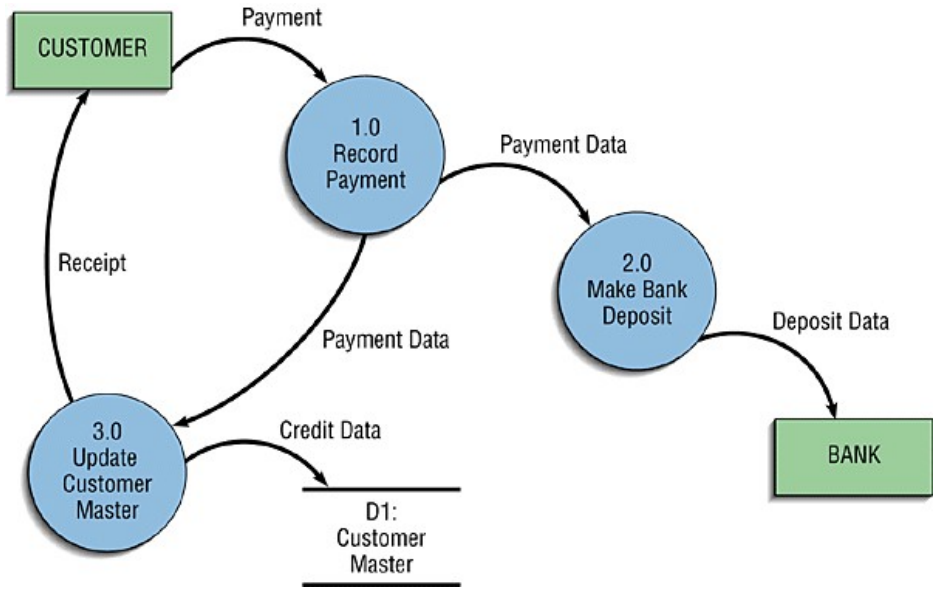
N. A data flow to a data store means update (delete or change)

O. A data flow from a data store means retrieve or use

P. Data flow has a noun phrase label

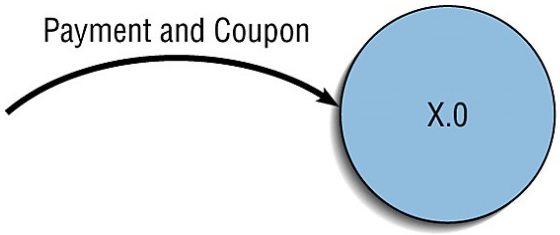
DFD Errors: Representing process as sink/source

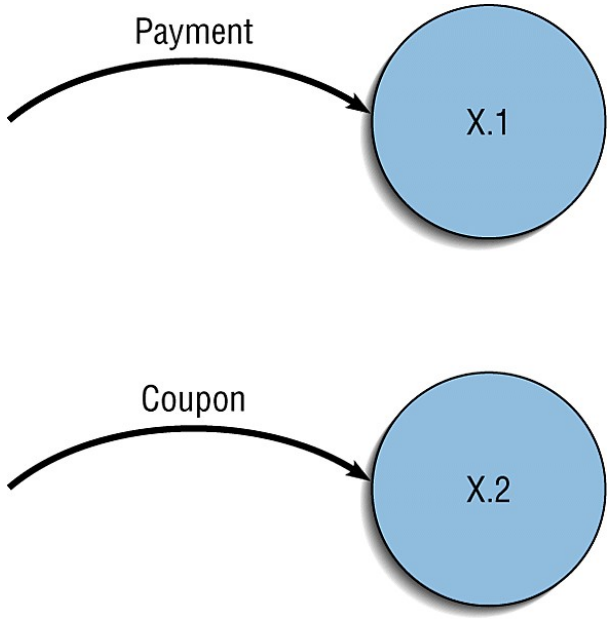




Data Flow Diagramming Rules: Advanced Rules

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 sub-flows.





Data Flow Diagramming Rules: Advanced Rules

R. The inputs to a process must be sufficient to produce the outputs from the process.

S. At the lowest level of DFDs, new data flows may be added to represent data that are transmitted under exceptional conditions (e.g., error messages).

T. To avoid having data flow lines cross each other, you may repeat data stores or sources / sinks on a DFD.