

Dataflow Diagrams

Data Flow Diagrams are a network representation of a system. Referred to as Bubble Charts, Bubble Diagrams, Process Model, Work Flow Diagrams or Function Models. DFDs are tools for modeling the functions (processes) in a system. Data flow diagrams do not show decisions or timing of events

The Four symbols used to represent any system at any level of detail are:

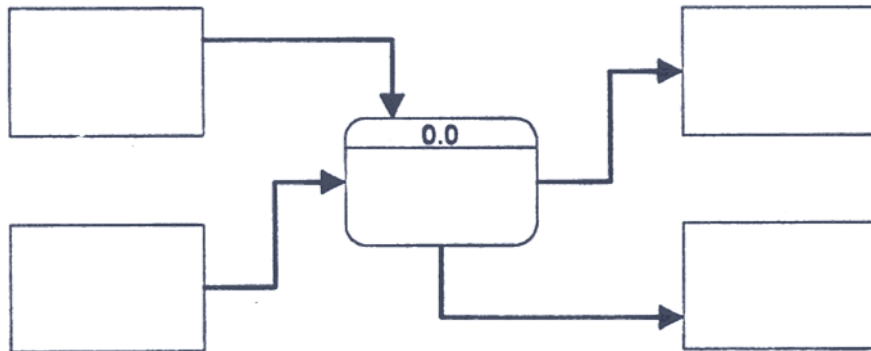
1. Data flows -movement of data in the system
2. Data stores -data repository for the data that is not moving
3. Processes -transforms incoming data flow(s) to outgoing data flow(s)
4. External entities -sources or destinations outside the specified system boundary

Multi-Level DFD's

Most dataflow diagrams are too complicated to draw the entire diagram on one page, so what you do is "explode" the diagram into multiple levels (multiple pages).

The highest level is called a context diagram (page one), this diagram contains ONE process symbol (the system) and illustrates the relationship between the system and external entities. This process is labeled 0.0.

(Context Example)



You may then explode the context diagram into a system diagram (label these processes 1.0) 2.0 and so on., . If you need to continue exploding these processes use decimals to show this explosion (i.e. 1.1, 1.2, or 1.1.1, 1.1.2.

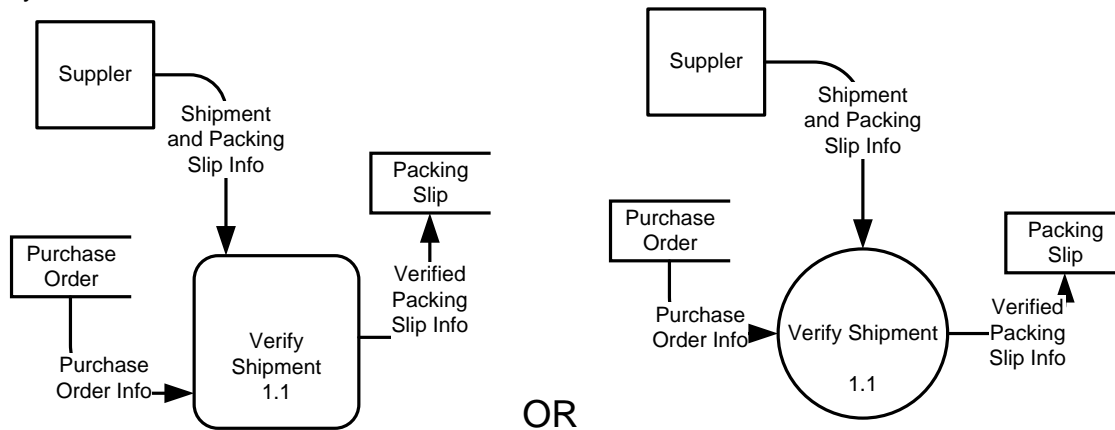
Dataflow Diagrams

The Process

common synonyms are: bubble, function, transformation

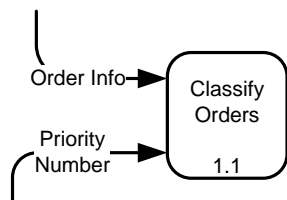
- Transforms inputs into outputs
- Shows how one or more inputs are changed into an output
- Graphically represented by a circle or a rectangle with rounded corners
- Named or described with a single word, phrase or simple sentence
- Process name describes what the process does

Symbols:

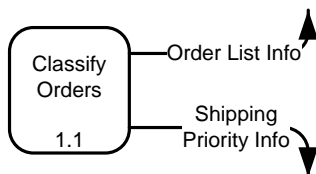


Rules: Avoid “Black Holes”, “Miracles” and “Grey Holes”

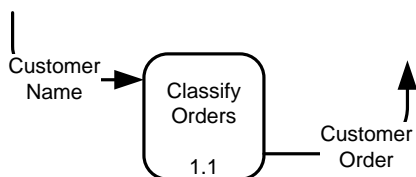
Black Hole: Process has Input but no Output Order Data



White Hole or Miracle: Process has Output but no Input.



Gray Hole: Process has insufficient Input for the Output it generates

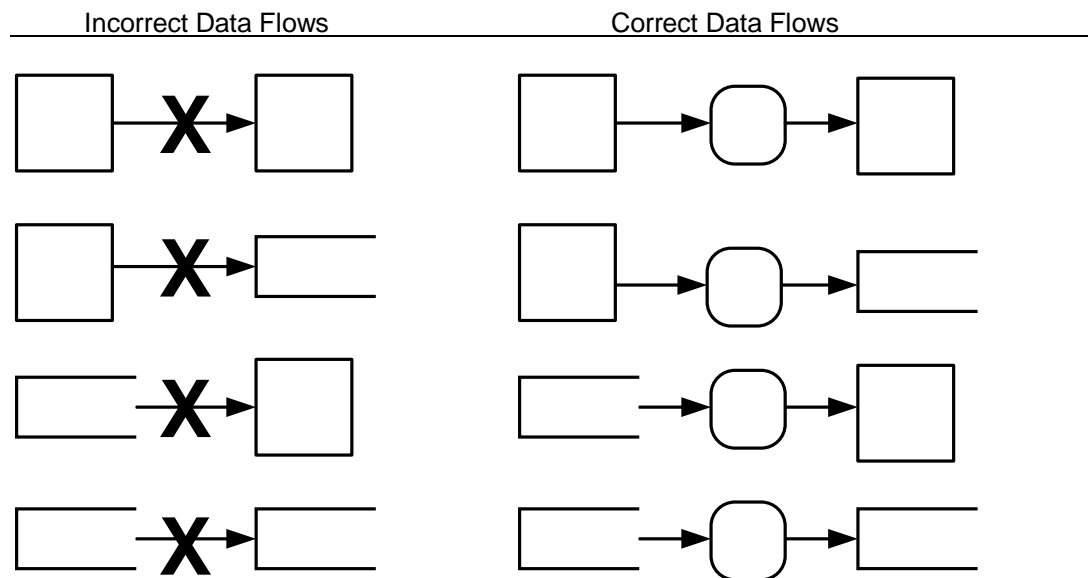
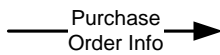


Dataflow Diagrams

The Data Flow

- Represented graphically by an arrow into or out of a process
- Used to describe the movement of packets of information from one part of the system to another
- Represents data in motion
- Name represents the meaning or content of the data that is moving along the flow
- Carries only one type of data
- Show direction; an arrowhead indicated the direction the data is traveling
- Show data movement in and out of stores, processes, external entities
- Can diverge -duplicate copies of a packet of data are sent to different parts of the system or a complex packet of data is being split into several more elementary data packets, each being sent to different parts of the system
- Dataflows can converge -several elementary packets of data are joining together to form a more complex, aggregate packet of data

Symbols:

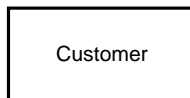


Dataflow Diagrams

The Data Store

- Used to model a collection of data packets at rest
- Name chosen is the plural of the data packet name
- Shown as a rectangle with an open end
- Can be a file, database, file folder, filing cabinet
- Stores are connected by flows to processes
- Flow from a store -interpreted as a read
- Flow to a store - interpreted as a write or an update, or a delete
- The store is changed as a result of the flow entering the store. It is the process connected to the other end of the flow that is responsible for making the change to the store

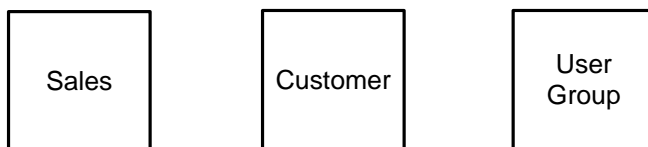
Symbol:



The External Entity

- Also referred to as a Source or Sink
- Drawn as a rectangle
- Can be a person, group of people, an outside agency, a department, can be another system with which your system communicates
- Must be outside the control of the system being modeled
- Generally easy to identify externals
- Connected to processes by flows this represents the interface between the system and the outside world
- Relationships between external entities are not shown. These are not part of the system. If they are, then show the entities as processes

Symbol:



Dataflow Diagrams

Guidelines For Constructing DFDs:

- 1) Choose meaningful names for processes, flows, stores, and entities
 - a) Label the process to identify the function the system is carrying out
 - b) Choose an active verb and an appropriate object to form a descriptive phrase
 - c) Avoid verbs such as DO, HANDLE, PROCESS
 - d) Avoid specific abbreviations and acronyms familiar to the user or to us
- 2) Number the processes
 - a) Numbering schemes should be consistent
 - b) Numbering schemes may imply a certain sequence of operation
 - c) Numbers are a convenient way of referencing processes
 - d) Numbers become the basis for a hierarchical numbering scheme
- 3) Avoid overly complex DFDs
 - a) Should be easy to understand, easy to absorb, easy to read and pleasing to the eye
 - b) Should fit comfortably on an 8.5 by 11 inch sheet of paper
 - c) Generally not more than 6 -8 processes per page
 - d) Context diagram shows one process that represents the entire system and highlights the interfaces between the system and the external entities
- 4) Redraw the DFD as many times as necessary for aesthetics
 - a) Draw, redraw, redraw
 - b) Should be technically correct
 - c) Should be acceptable to the user
 - d) Should be neat
- 5) Make sure your DFD is logically consistent
 - a) Avoid processes that have no outputs -"black holes"
 - b) Avoid processes that have no inputs -"miracles"
 - c) Beware of unlabelled flows and unlabelled processes
 - d) Beware of read-only or write-only stores
 - e) Make sure that dataflows coming into and going out of an entire level correspond to the dataflows coming into and going out at the next lower level which describes that process
 - f) A store at the highest level serves as an interface between two or more processes; show it at EVERY lower-level that further describes those processes
 - g) Do not show local stores for a lower level at the higher level