

Information Engineering

STRUCTURED SYSTEMS ANALYSIS
DIAGRAMS (DFDS)

B HOLLAND

Process Modeling

Graphically represents the processes that capture, manipulate, store, and distribute data between a system and its environment and among system components

Process modelling can be illustrated by either

- DFDs (Data flow diagrams)
- UML (Unified Markup Language)

This presentation will discuss DFDs

Process Modeling

Data-flow Diagrams (DFD)

- Graphically illustrate movement of data between ,external entities and the processes and data stores within a system.

Process Modeling (continued)

Data flow diagrams can be used in both Analysis and Design phase of the SDLC.

- At analysis phase they will illustrate an existing system.
- At design phase they will illustrate a proposed system.

Process Modeling (continued)

There are actually four different types of data flow diagrams used in the systems development process:

1. A physical model to describe how the existing system works now (the how of now).
2. A logical model to describe what the existing system does now (the what of now).

Process Modeling (continued)

3. A logical model to describe what the new system will do (the what of the new).
4. A physical model to describe how the new system will work (the how of the new).

Of course, the first two steps are unnecessary if a completely new system is to be built. However, all the steps apply if you wish to re-engineer an existing system (including manual systems).

Process Modeling (continued)

Physical DFD's depict these physical elements. In a current physical DFD, process labels include the names of people or their positions or the names of computer systems that might provide some of the system's processing.

That is, the label includes an identification of the technology used to process the data. Similarly, data flows and data stores are often labelled with the names of the physical media on which data flow (e.g. forms) or in which data are stored (e.g. folders).

Process Modeling (continued)

Logical DFD's describe data, processes, etc. differently. They are more abstract than physical DFD's but that difference is important.

If a key worker becomes ill or chooses to leave the company, the work will remain.

Desks, files and computer equipment will change too.

By focusing on the underlying elements – the logical, non-physical aspects – you gain an understanding of the fundamental structure of the system.

A logical DFD is derived from the physical version by -

- All data flow names should be examined to ensure that each name reflects the data flowing between processes, data stores and entities, not the documents that contain them. For example, 'order form' should be changed to perhaps 'order details'. Only data needed to perform the process should be an input to the process.

A logical DFD is derived from the physical version by -

- Remove routing information, that is, take out the names of people or departments that perform the processes. Show the flow between procedures, not between people, offices or locations.

A logical DFD is derived from the physical version by -

- Remove references to devices (for example, folders, file cabinets or pending trays).

A logical DFD is derived from the physical version by -

- Rationalise data stores. Consolidate redundant data stores and turn temporary data stores into data flows.

A logical DFD is derived from the physical version by -

- Remove unnecessary processes, such as those that do not change the data or data flows (e.g. routing, sorting or copying). Consolidate processes to remove duplication of effort.

A logical DFD is derived from the physical version by -

- Finally, the current logical DFD's need to be reviewed. They should be agreed with users, which may lead to further iteration among these tasks.

Process Modeling (continued)

There are different notations to draw DFDs e.g.

- Yourdon & Coad
- Gane & Sarson,

BE CLEAR IN WHICH NOTATION USED
BE CONSISTENT !!!

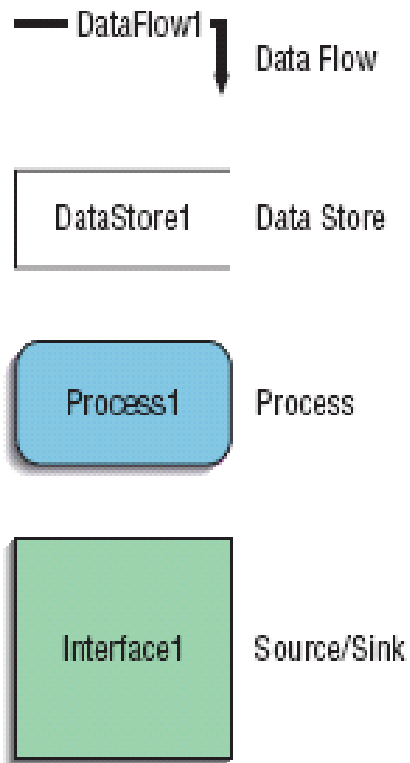


FIGURE 6-2

Gane and Sarson identified four symbols to use in data-flow diagrams to represent the flow of data: data-flow symbol, data-store symbol, process symbol, and source/sink symbol. We use the Gane and Sarson symbols in this book.



Function

Data flow diagram -
[Yourdon/DeMarco](#) notation.



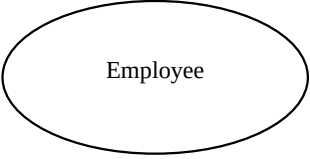
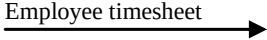
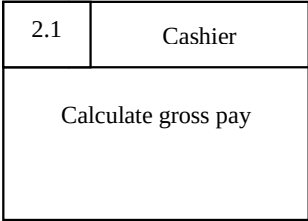
File/Database



Input/Output



Flow

	An external entity
	A data flow
	A process
	A data store

Celko, 1987

Data-Flow Diagramming Mechanics

Data Flow

- The arrow shows movement of data (about a person, place or thing) from one point to another, in the direction of the arrowhead.
- A data flow arrow is labelled with a meaningful description of the data in motion (a noun).

Data-Flow Diagramming Mechanics

- The name represents the aggregation of all individual elements of data moving as part of one packet.
- In a physical DFD the arrow might be labelled with the names of actual documents, e.g. timesheets, order form, etc.,
- In a logical DFD the arrow must be labelled with a term which indicates the nature of the important *data*, e.g. hours worked, order details, etc.

Data-Flow Diagramming Mechanics (continued)

Data Store

- Depicts data at rest
- May represent data in
 - File folder
 - Computer-based file
 - Notebook
- Drawn as a rectangle with the right vertical line missing
- Label includes name of the store as well as the number

Data-Flow Diagramming Mechanics (continued)

Data Store

- A data store might contain data
E.g. customers, customer orders, supplier invoices.
- The data store is always assigned a unique number
 - if a physical DFD, may be prefixed by-
 - M to represent a manual store (e.g. filing cabinet)
 - D to represent a computerised store (e.g. database).

Data-Flow Diagramming Mechanics (continued)

Process

- Depicts work or actions performed on data so that they are transformed, stored, or distributed
- Drawn as a rectangle with rounded corners
- Number of process as well as names are recorded

Data-Flow Diagramming Mechanics (continued)

Process

- Data processes involve data being used or altered.
- The process could be manual or computerised.
- Each process must be given a unique identifying number indicating its level in the diagram hierarchy.
- To the right of the ID number, the person or department conducting the process may be entered (physical DFD's only).
- Inside the lower portion is a name for the process.
 - This should be precise and meaningful, usually following the verb-adjective-noun format.

Data-Flow Diagramming Mechanics (continued)

Source/Sink

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

Data-Flow Diagramming Definitions

Levelled DFDs

The complexity of business systems means that it is impossible to represent the operations of any system by means of a single diagram.

Therefore, in systems analysis we separate a system into its component subsystems or processes, which are 'decomposed' (or broken down) into smaller sub-systems or processes, each decomposition revealing more detail

Data-Flow Diagramming Definitions

Context Diagram

- A data-flow diagram 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

Level-O Diagram

- A data-flow diagram that represents a system's major processes, data flows, and data stores at a higher level

Data-Flow Diagramming Definitions

BE CAREFUL ON THIS POINT....

...Some of the literature uses the terms Context Diagram and Level 0 DFD interchangeably!!! With the level increasing along with the level of detail of the diagram

Data-Flow Diagramming Definitions

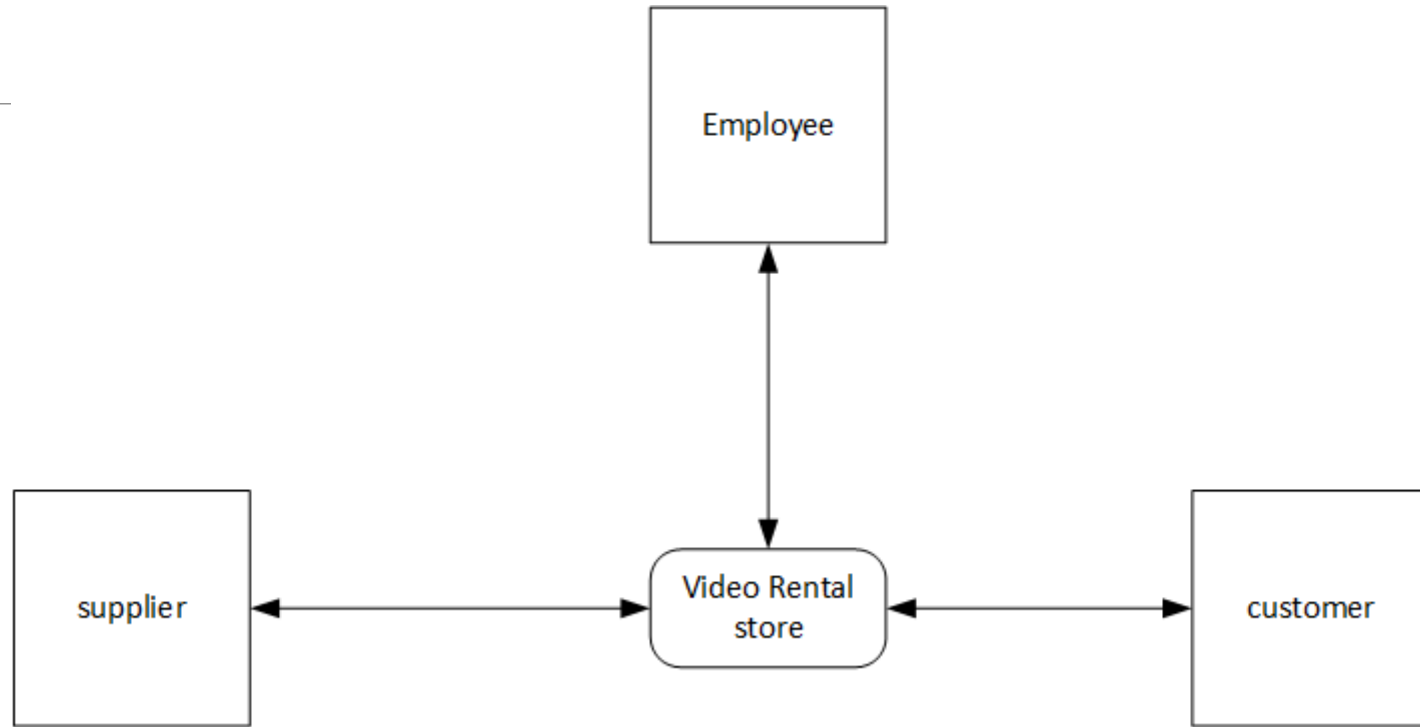
Level 1 and beyond

This is in turn 'exploded' by means of a more detailed data flow diagram, known as a level 1 DFD.

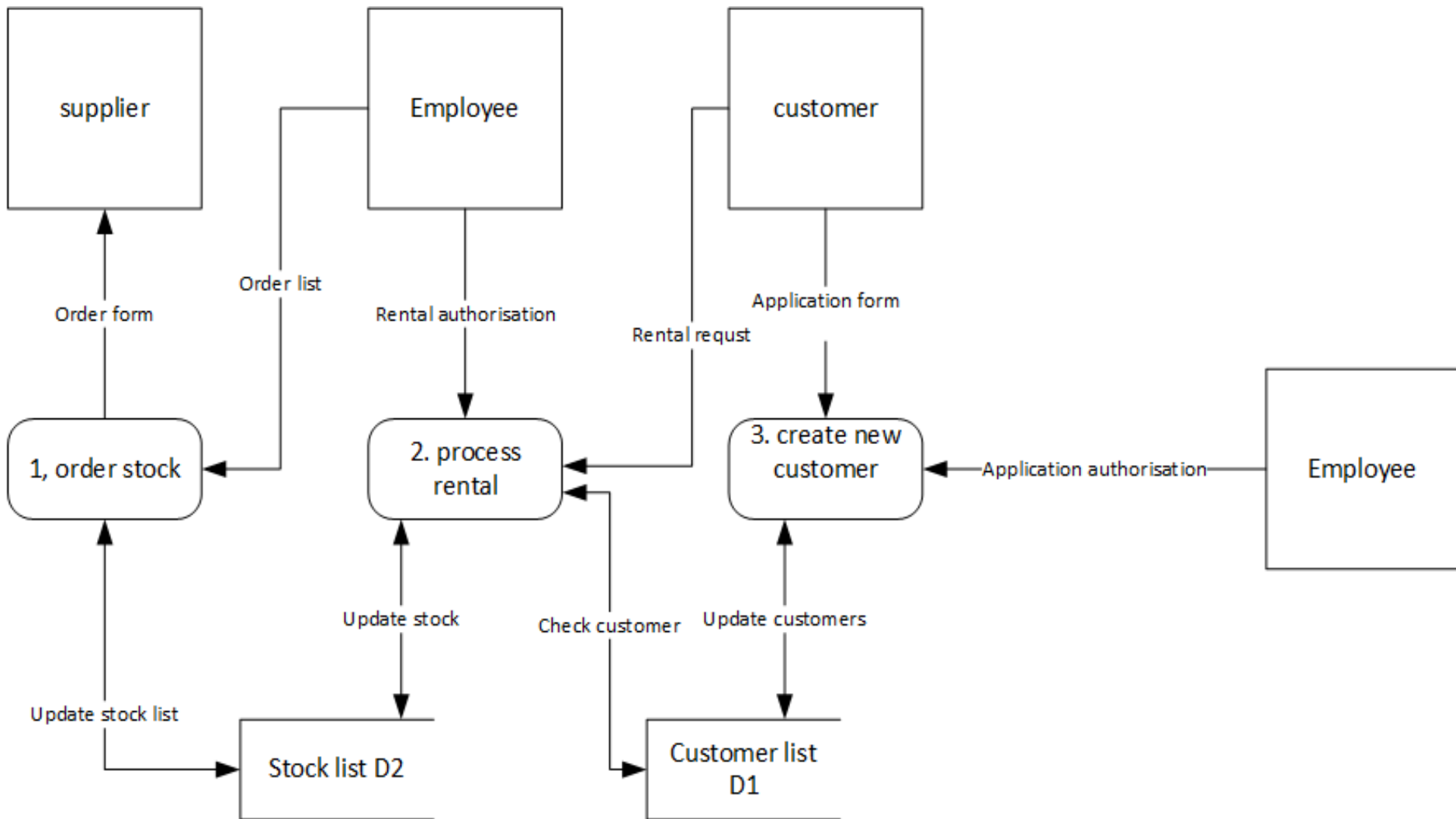
Further detail can be represented on a level 2 DFD, and so on until all individual entities, stores, flows and processes are shown.

The following diagrams illustrate how levels of DFD's are built up.

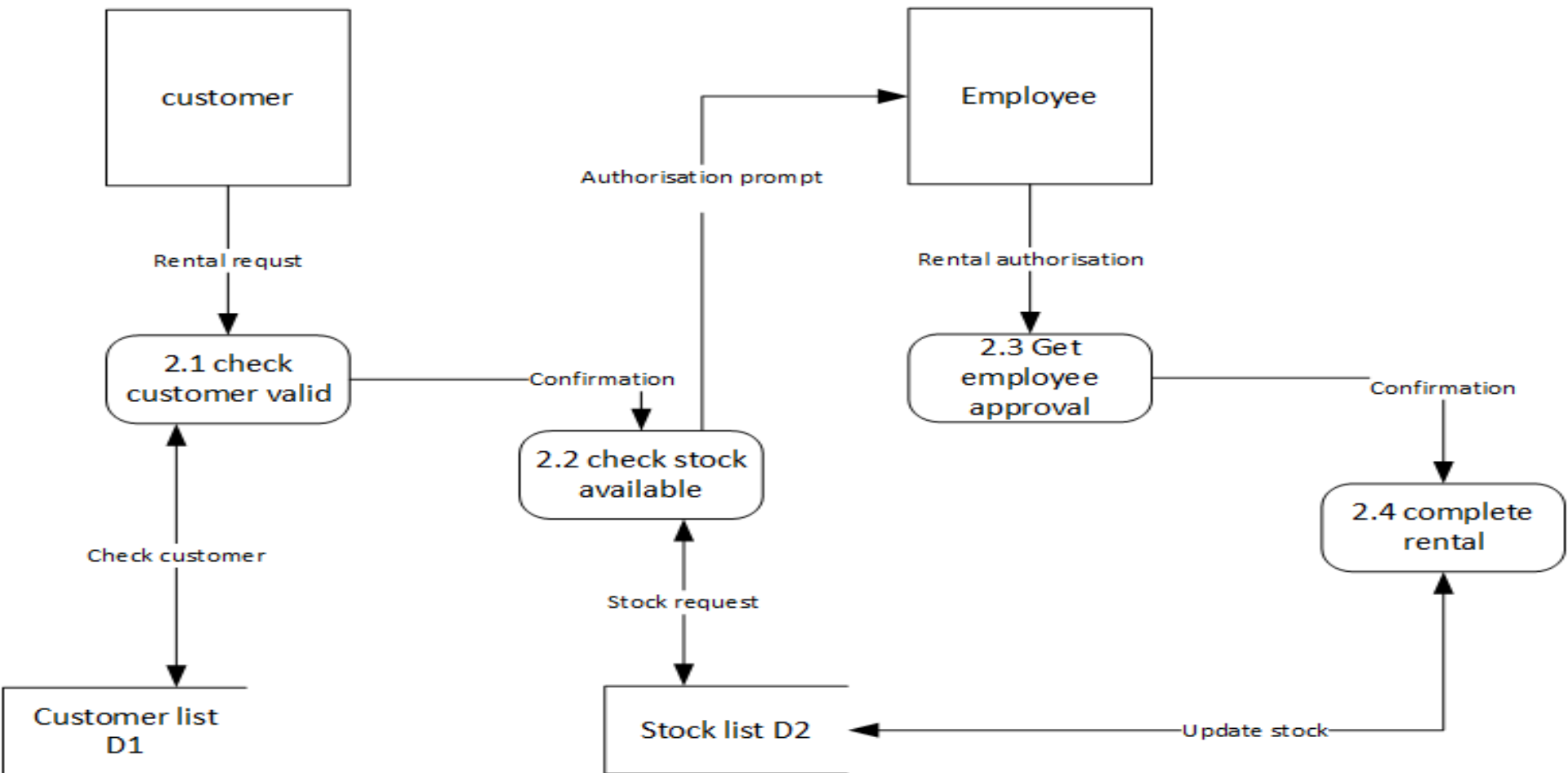
Context Diagram



DFD Level 0



DFD Level 1 (for process 2)



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

DATA FLOW

- A data flow has only one direction of flow between symbols
- A fork means that exactly the same data go from a common location to two or more processes, data stores, or sources/sinks
- 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
- A data flow cannot go directly back to the same process it leaves
- A data flow to a data store means update
- A data flow from a data store means retrieve or use
- A data flow has a noun phrase label

Data-Flow Diagramming Rules

PROCESS

- No process can have only outputs. It is making data from nothing (a miracle). (If an object has only outputs, then it must be a source.)
- No process can have only inputs (a black hole). If an object has only inputs, then it must be a destination (sink).
- A process has a meaningful verb-adjective-noun label which is unique.

Data-Flow Diagramming Rules

DATA STORE

- Data cannot move directly from one data store to another data store. Data must be moved by a process.
- Data cannot move directly from an outside source to a data store. Data must be moved by a process that receives data from the source and places the data into the data store.
- Similarly, data cannot move directly to an external destination (sink) from a data store. Data must be moved by a process.
- A data store has a noun phrase label.

Data-Flow Diagramming Rules

SOURCE DESTINATION

- Data cannot move directly from a source to a destination. It must be moved by a process if the data are any concern to our system. Otherwise, the data flow is not shown on the DFD.
- A source/destination has a noun phrase label.

Data-Flow Diagramming analysis tips

1. Interview everyone (or subset) of individuals within the system.
Find Out-
 - What data they use
 - What they do with it
 - What data stores do they use
 - Which external systems/departments/people do they interact with

Data-Flow Diagrams Where to start

Context level

- Draw and label a ONE CENTRAL PROCESS to represent the whole system (ignore all processes and data stores, and internal data flows).
- Read the narrative, and list all sources and destinations of data (external entities), and their associated data flows (data that enters or leaves the system).
- Draw them onto your diagram (data flow arrowhead pointing into the system from sources, outward pointing for destinations).
- Draw the system boundary (useful to 'balance' different DFD levels).

Data-Flow Diagrams Where to start

Level 0

- Create a list of all processes, and their associated data stores.
- Use your judgement to find some logical way to 'group' processes (at this level we show primary processes only – the detail regarding these 'parent' processes is hidden, and revealed in the level 1 'child' diagram).
- Draw in, and number, the primary processes.
- Draw the external entities (identified at previous level).
- Draw in the data flows that 'join' entities to processes.

Data-Flow Diagrams Where to start

Level 0

- Working your way through the narrative, enter the data flows associated with each process (inputs and outputs).
- At each process, ask yourself 'what data is needed for this process?' This should also help reveal any dependence on data stores. Draw in any data stores that are used by more than one process, ignore others.
- Draw the system boundary, and compare with the context diagram. The data flows crossing the boundary should be the same on both diagrams (balancing).
- Draw a boundary around each primary process.

Data-Flow Diagrams Where to start

Level 1

- Select for decomposition (one at a time), a 'parent' process (one per diagram/page).
- Draw, and number, the processes involved in the parent process.
- Working your way through the narrative, enter the data flows associated with each process (inputs and outputs).
- At each process, ask yourself 'what data is needed for this process?' This should also help reveal any dependence on data stores. Draw in all data stores.
- Draw the sub-system boundary, and compare with the level 1 diagram. The data flows crossing the boundary should be the same on both diagrams (balancing).

Data-Flow Diagramming analysis tips

2. Draw an outline Data Flow Diagram for each person or team

Check back with stakeholders to check that

- all relevant data flows are represented
- All process are represented and appropriately denote the real-life scenario

Data-Flow Diagramming analysis tips

3. Take two diagrams and combine (pay attention to)

Common elements

- Processes
- Data flows
- Stores
- External entities

Data-Flow Diagramming analysis tips

4. Take the next diagram and continue combination

Gradually an overall picture of the system will emerge

If complex then some processes may be sub-processes (produce further level DFDs)