Table of COntents

[1. Introduction 2](#_Toc390979646)

[1.1. Development and purpose of DFDs 2](#_Toc390979647)

[2. Components of a Data Flow Diagram 4](#_Toc390979648)

[2.1. Components 4](#_Toc390979649)

[2.1.1. External entities 5](#_Toc390979650)

[2.1.2. Processes 5](#_Toc390979651)

[2.1.3. Data stores 6](#_Toc390979652)

[2.1.4. Data flows 7](#_Toc390979653)

[3. Developing Data Flow Diagrams 7](#_Toc390979654)

[3.2 Context diagram 8](#_Toc390979655)

[3.3 Level 1 DFD 9](#_Toc390979656)

[3.4 Check list 10](#_Toc390979657)

[4. Categories of Data Flow Diagrams 10](#_Toc390979658)

[5. Example Case Study – Pizza Supreme 10](#_Toc390979659)

List of Figures

[Figure 1: Document Flow Diagram 4](#_Toc388558313)

[Figure 2: Examples of external entities 5](file:///D:\Personal\BA\Training%20Material\Final\Reading%20Material\DFD.docx#_Toc388558314)

[Figure 3: Process Template 6](file:///D:\Personal\BA\Training%20Material\Final\Reading%20Material\DFD.docx#_Toc388558315)

[Figure 4: Data store template 6](file:///D:\Personal\BA\Training%20Material\Final\Reading%20Material\DFD.docx#_Toc388558316)

[Figure 5: Data store examples 6](file:///D:\Personal\BA\Training%20Material\Final\Reading%20Material\DFD.docx#_Toc388558317)

[Figure 6: Data flows - examples 7](file:///D:\Personal\BA\Training%20Material\Final\Reading%20Material\DFD.docx#_Toc388558318)

[Figure 7: DFD Levels 8](#_Toc388558319)

[Figure 8: Sales system - context diagram 9](file:///D:\Personal\BA\Training%20Material\Final\Reading%20Material\DFD.docx#_Toc388558320)

# Introduction

This unit deals with one of the major techniques for recording the requirements of a user for a new computer application. An initial diagram is constructed to show the processes which are being implemented in an existing system. The diagram helps to show how information is used to produce the functions that are required by the current system. It also shows what information is provided to the system and what information is provided form the system. Other benefits include the documentation of who is using the system and what data will be stored. By careful construction of the DFDs (data flow diagrams) the boundaries of the system to be built may be clearly identified. This helps to clarify what will and what will not be constructed. It will also show the interaction that may be required with other systems.

The data flow diagrams should also have some associated documentation. This is necessary as the diagrams are meant as a visual representation of the way in which information is processed. There is limited space on the diagrams so that documentation to explain, refine and describe further details of what is shown need to be kept somewhere in the proposed system documentation. The data flow diagrams and the associated documentation together combine to form a data flow model. This is also commonly called a process model.

The user requirements when complete are used as a basis for the development of the system. Later when the system has been developed it can be tested against the initial requirements to see whether the user’s needs have been met.

# Development and purpose of DFDs

This is a change. Before attempting to construct an initial DFD it is necessary to gather and digest information that helps us to understand how data is processed in the current system. Fact-finding techniques are used for this purpose and are discussed in another Unit. As the DFD is constructed a systems analyst will often come across areas of doubt where the precise way to model the system is unclear. This is a natural part of the development and should not be regarded with alarm. In fact, it is expected and it is a consequence of attempting to model the current situation that questions will be asked to clarify the exact processes which are taking place. Sometimes the analyst will make an assumption and then check this with the user at a subsequent meeting.

Results of interviews, documents, reports, questionnaires etc. will all play a part in helping the analyst to gain an insight into the current processes. Where a system is being developed from scratch the analyst will work with the user to develop the proposed DFDs.

When all the information about the current system is gather it should be possible to construct the DFDs to show:

* the information that enters and leaves the system
* the people/roles/other systems who generate and/or receive that information
* the processes that occur in the system to manipulate the information
* the information that is stored in the system
* the boundary of the system indicating what is (and what is not) included

As a starting point, it is sometimes useful to construct a document flow diagram. This diagram shows how ‘documents’ are passed around an organization to fulfill the current requirements of the system under investigation. The term’ documents’ is interpreted very loosely and usually translates to information.

Sometimes before beginning to produce the set of data flow diagrams a document flow diagram is generated. This helps to establish the system boundary so we can decide which parts of the system we are modeling and which parts we are not. The document flow diagram shows the different documents (or information sources in the system and how they flow from a source to a recipient.

Here is an example of a document flow diagram shown below.

Here ellipses represent either the source or recipient of the ‘documents’ and named arrows show the direction of transfer and the nature of the information being exchanged. This kind of diagram is a first step in understanding what information is in the system and how it is used to perform the required functions.

Another useful feature of the construction of this type of diagram is that it enables a sensible discussion of where the system boundary should lie. In other words it is important to establish what is to be included in the proposed information system and what is not. To indicate this, a system boundary line is constructed on the document flow diagram.

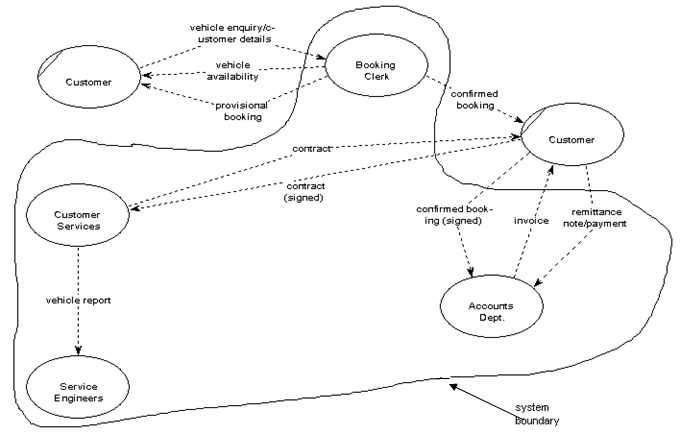


Figure 1: Document Flow Diagram

The DFDs are used to:

* discuss with the user a diagrammatic interpretation of the processes in the system and clarify what is currently being performed
* determine what the new system should be able to do and what information is required for each different process that should be carried out
* check that the completed system conforms to its intended design

# Components of a Data Flow Diagram

The components of a Data flow Diagram are always the same but there are different diagrammatic notations used. The notation used here is one adopted by a methodology known as SSADM (Structured Systems Analysis and Design Methods)

# Components

Luckily there are only four different symbols that are normally used on a DFD. The elements represented are:

* External entities
* Processes
* Data stores
* Data flows

# External entities

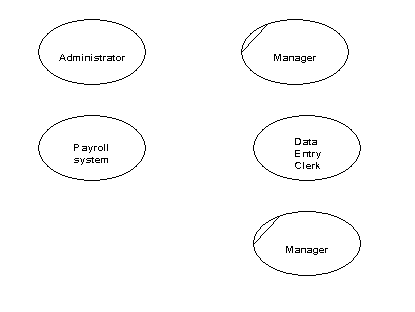
External entities are those things that are identified as needing to interact with the system under consideration. The external entities either input information to the system, output information from the system or both. Typically they may represent job titles or other systems that interact with the system to be built. Some examples are given below in Figure 1. Notice that the SSADM symbol is an ellipse. If the same external entity is shown more than once on a diagram (for clarity) a diagonal line indicates this.

Figure : Examples of external entities

# Processes

Processes are actions that are carried out with the data that flows around the system. A process accepts input data needed for the process to be carried out and produces data that it passes on to another part of the DFD. The processes that are identified on a design DFD will be provided in the final artifact. They may be provided for using special screens for input and output or by the provision of specific buttons or menu items. Each identifiable process must have a well chosen process name that describes what the process will do with the information it uses and the output it will produce. Process names must be well chosen to give a precise meaning to the action to be taken. It is good practice to always start with a strong verb and to follow with not more than four or five words.

Examples of good process names would be:

* Enter customer details
* Register new students
* Validate sales orders

Try to avoid using the verb ‘process’, otherwise it is easy to use this for every process. We already know from the symbol it is a process so this does not help us to understand what kind of a process we are looking at.

The process symbol has three parts as shown in Figure 2.

Figure : Process Template

Where/who

Process identifier

Process name

# Data stores

Data stores are places where data may be stored. This information may be stored either temporarily or permanently by the user. In any system you will probably need to make some assumptions about which relevant data stores to include. How many data stores you place on a DFD somewhat depends on the case study and how far you go in being specific about the information stored in them. It is important to remember that unless we store information coming into our system it will be lost.

The symbol for a data store is shown in Figure 4 and examples are given in Figure 5.

Figure : Data store template

Data store identifier

Data store description

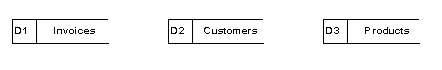


Figure : Data store examples

As data stores represent a person, place or thing they are named with a noun. Each data store is given a unique identifier D1, D2 D3 etc.

# Data flows

The previous three symbols may be interconnected with data flows. These represent the flow of data to or from a process. The symbol is an arrow and next to it a brief description of the data that is represented. There are some interconnections, though, that is not allowed.

These are:

* **Between a data store and another data store:** This would imply that one data store could independently decide to send some of information to another data store. In practice this must involve a process.
* **Between an external entity and a data store:** This would mean that an external entity could read or write to the data stores having direct access. Again in practice this must involve a process.

Also, it is unusual to show interconnections between external entities. We are not normally concerned with information exchanges between two external entities as they are outside our system and therefore of less interest to us. Figure 6 shows some examples of data flows.

Figure : Data flows - examples

Applicant’s name

Employee record

Payment

Customer details

# Developing Data Flow Diagrams

Data flow diagrams usually occur in sets. The set consist of different levels. To start with a context diagram is drawn. This shows the people and/or systems that interact with the system under construction. By interaction we mean putting information in or taking information out of our system. This diagram gives an overview of the information going in and coming out of the system.

The system is represented by a box. In this DFD (level 0 DFD) no processes or data stores are shown.

Another diagram (level 1 DFD) is now developed which shows the processes taking place to convert the inputs shown in the context diagram to the outputs. In this DFD detail is given to show which processes are responsible for accepting the different inputs and producing the different outputs. Any process shown that is complicated by a number of data flows and requires further refinement is shown on another diagram (level 2 DFD) where sub-processes are shown together with any necessary extra data stores. Figure 8 shows how these levels of detail are related to one another

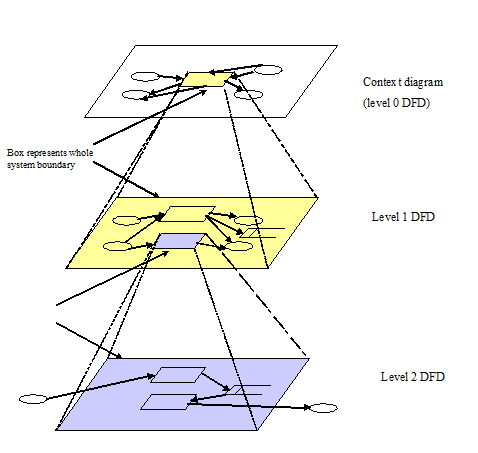


Figure 7: DFD Levels

# Context diagram

This DFD provides an overview of the data entering and leaving the system. It also shows the entities that are providing or receiving that data. These correspond usually to the people that are using the system we will develop. The context diagram helps to define our system boundary to show what is included in, and what is excluded from, our system.

The diagram consists of a rectangle representing the system boundary, the external entities interacting with the system and the data which flows into and out of the system. Figure 9 gives an example of a context diagram.

# Level 1 DFD

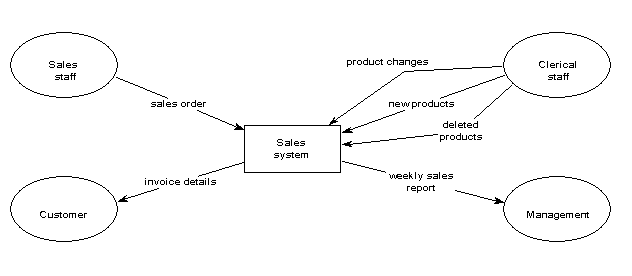
Now we wish to develop further a model of what the system will do with the information the external entities will supply to it. We construct a diagram which is, in effect, taking a magnifying glass to the system boundary represented by the rectangle in the context diagram. We will be looking inside the rectangle and describing what is done with the data inputs in order to provide the data outputs required. The level 1 DFD we construct is a ‘child diagram’ of the context diagram. We should see all the inputs outputs and external entities that are present in the context diagram. This time we shall include processes and data stores. When students come to construct these level 1 DFDs for the first time they are often unsure as to how many processes to show. It somewhat depends on the case study. However, as a guide it is probable that you would not want more than eight or nine processes

Figure : Sales system - context diagram

as more than this would make the diagram too cluttered. In addition the process names should be chosen so that there are at least three. Fewer than this would mean that the system was unrealistically simple.

Note the following features:

* Every data flow on the context diagram, to or from an external entity, is also shown on the Level 1 DFD.
* Each process has a good strong verb describing what the process is doing with the information received.
* Some data stores appear more than once. This is indicated by a double vertical line on the left hand side of the symbol.
* Each process has access to the relevant information to be able to produce the required output
* Each process has at least one data flow input and one output.
* Information input to the system is always stored somewhere and so never lost
* At this stage the invoice data store has an input data flow but no output. This would indicate that the invoice data is never used. Later on in the development of this system we might define a process for checking the invoices have been paid and so this information in the invoice data store would then be used.
* One of the processes (process 8) has no asterisk in the bottom right hand corner although all the others do. This is explained in the next section.

# Check list

There are many errors that may occur when drawing data flow diagrams. Here is a check list to help you avoid some of the major difficulties.

* External entities must be people or systems that send information to or accept information from the system to be engineered
* Data flows must always be labeled with the data they contain. Do not put verbs in the data flow description as this implies a process
* Parent and child diagrams should be consistent. Do not show a data flow coming from or to an external entity on a level 1 DFD that isn’t shown on the context diagram (and vice versa)
* Check the direction of data flows to and from data stores
* Make sure each process has at least one input and one output
* Each data store should have at least one input and one output on the DFDs somewhere
* Each process name should start with a verb
* Where a process has only two data flows (one input and one output) then check it. Usually a data flow has been omitted.

# Categories of Data Flow Diagrams

Data flow diagrams may be categorized as either physical or logical. A logical diagram shows how the business operates but not how the system will be constructed. The logical diagram omits details of how the more physical aspects are implemented. The logical model is more to do with the business whilst the physical model shows the whole system. The physical model deals with the way that information is stored, the manual procedures used, the necessity for temporary data collections etc.

# Example Case Study – Pizza Supreme

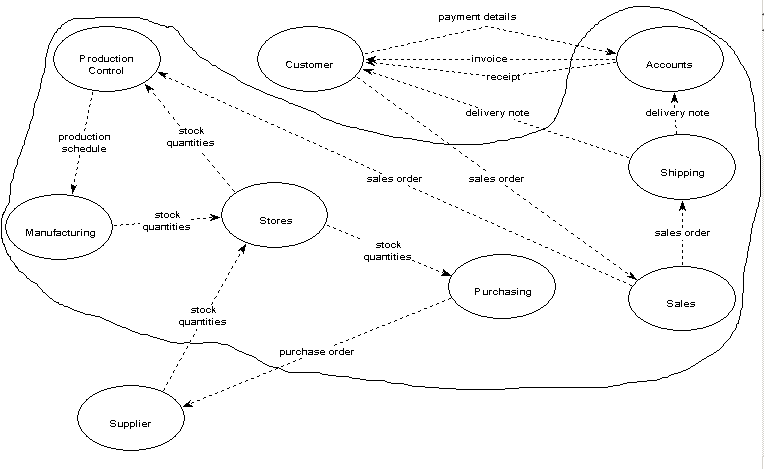
*A large pizza business makes pizzas and sells them. The pizzas are manufactured and kept in cold storage for not more than two weeks.*

*The business is split into a number of functional units. There is Production Control, Manufacturing, Stores, Accounts, Sales, Shipping and Purchasing. Production Control is responsible for organizing which pizzas to produce in what order and in what quantity. They need to schedule the production of the pizzas according to the current and expected sales orders together with the number of pizzas already in stores. Manufacturing take the raw materials from the stores and manufacture pizzas returning the completed goods to the stores. Accounts deal with the payments for the pizzas when delivered to the customer and the payment to the suppliers of the raw materials. Sales deal with customer orders whilst Purchasing organize the buying of raw material from suppliers. Shipping manages the packing and delivery of the goods to the customer with a delivery note.*

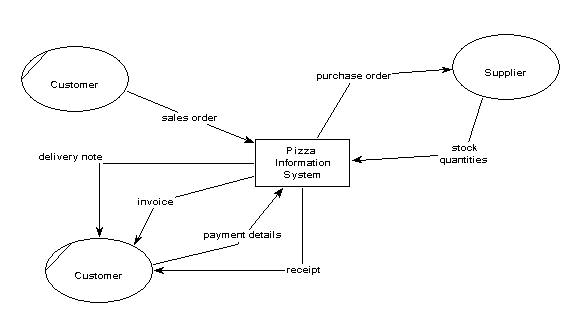
*When a sales order is received by sales they record what is being ordered and by whom. They also record the details of the expected date of delivery. Production Control access this information and make sure that, if required, pizzas are produced by Manufacturing and are ready in Stores for when the delivery needs to be made.*

*After the delivery is made Accounts make sure that the customer receives an invoice and that payment for the invoice is received at which time a receipt is issued. Purchasing look at the current stock of raw materials and by using current stock levels, supplier turnaround times and quantity to be ordered decide what needs to be ordered on a daily basis. Their aim is never to run out of an ingredient but to minimize the amount of raw material kept in stock.*

Sometimes a document flow diagram may be drawn to show the way in which information flows around the system. Here is one shown below.



Some of the details here have been assumed and would need to be checked with the user to make sure they are a true reflection of what is happening. By producing this diagram the analyst is gaining an understanding of how the current system is working in terms of the information that is being used and how it is being passed around the system. Here we assume the system to be produced will include all the functions provided by the different functional areas but the suppliers and customers are regarded as being outside the system.

Form this document flow diagram we may construct a context diagram. This will show the information entering and leaving the system. It also shows which external entities are supplying or receiving the information.

We may now develop a level 1 data flow diagram from the context diagram. The level 1 data flow diagram must show the same data flows from and to the external entities as appear on the context diagram. If this is not the case there would be an inconsistency. We need to decide which processes are handling the information input to the system and which are producing the information outputs that are shown on the context diagram. We also need to think about the process names involved and which data stores are required. Remember if data is not stored away it will be lost.

Looking at this level 1 diagram we see that process ‘Order raw materials’ has a number of incoming and outgoing data flows and to explain what is happening probably needs some decomposition. This then is a candidate for a level 2 DFD. Before looking at the level 2 DFD for this process examine the level 1 DFD and make sure you understand why all the data flows are as they are. Check the diagram against the check list given in section 3.4. You should find this raises at least one problem that will need to be resolved !

Check the level DFD is consistent with the level 1 DFD (its parent diagram).

This completes the example showing how Data Flow Diagrams may be constructed. We have produced an outline of the processes that are currently occurring in the physical system. We could now continue to generate a set of current logical DFD’s and from there build a set of required logical DFD’s. Of course the required processes differ from the current processes in so much as when a new system is produced it should try to solve any of the current problems with the old system and also take advantage of any new innovations that may now exist due to the introduction of new technology.

