

Highlights

- System Algorithms
- Role of an Analyst
-

Inside

-
-
-

SA LEC 02

*Bolger 'n' Slaters
Systems
Analysis Course
Notes
Issue 2*

The Systems Analysis

 LECTURE

SA

The Role of a Systems Analyst

Analyst Tools and Techniques

Thinking like a Systems Analyst

Before you set out on the course that will (hopefully) turn you into a Systems Analyst / Designer, just take a minute or two to see just how good (or how bad) a analyst/designer you are to start with.

Worked Activity

Write down the algorithm needed (ie the steps to be carried out) in order fill a bucket with water.

Its more than likely (and acceptable at this stage of your development) that your algorithm would looks something like the one below give or take a few lines:

1. **Get bucket**
2. **Switch on tap**
3. **Fill bucket**
4. **Switch off tap**

However with a little training and Systems Analysis/Design experience you would if asked the question perhaps have written out a little more detailed design that took into account more user requirements, similar perhaps in content and complexity to the one below.

1. **Get a bucket**
2. **Ask for amount of water to put in bucket**

3. **Get user reply and remember as AMOUNT**
4. **Ask how fast the water is put into bucket**
5. **Get user reply and remember as SPEED**
6. **Switch on Tap**
7. **Set speed of flow to desired SPEED**
8. **Place bucket under tap**
9. **Repeat following step(s)**
 10. **Put water in bucket**
11. **Until Amount in bucket = AMOUNT asked for**
12. **Switch off tap**
13. **Remove bucket**

No doubt with even more thought and analysis to the problem the algorithm would become even more refined and improved to take account of other issues such as asking the user which size bucket should be used, or what actual test would be made in order to determine the correct amount of water was put into the bucket.

Being able to undertake Analysis and Design correctly only comes with a great deal of

- practice,
- study and
- thought.

You will be given plenty of the first ingredient, however only you can provide plenty of the second and third ingredients.

Worked Activity

1. Produce an Algorithm to put petrol in a Car
2. Produce an Algorithm to provide a ' brew ' from a vending machine
3. List the steps / tasks undertaken in order to become a student at the college

The Role Of The Systems Analyst

The Systems analyst has a key role to play in developing computer systems. It is their job to

- Analyse the data processing requirements of the organisation
- Decide whether computerisation should be introduced, or the current computer system modified or changed
- Specify how the new computer system should work, and what the hardware and software requirements will be
- Be responsible for implementing the new system and ensuring that it works efficiently

The systems analyst therefore has to have a good understanding of the nature of the business or organisation by whom they are employed, and a thorough understanding of how computers can be used.

They need to be an excellent communicator, capable of extracting the required information about the current system from people in the user departments without alienating them.

Analysis Techniques And Tools

In describing a system the system analyst must cope with two aspects of the system

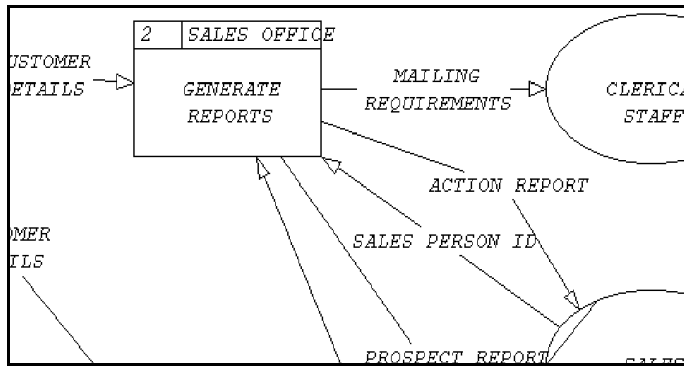
1. **Its structure and function - how the various bits fit together and how they work.**
2. **Its data - the "raw material" processed by the system.**

The first of these is fairly straightforward. The analyst will use structure charts, block diagrams and description, while the way the system works (its logic) is shown using flowcharts and decision tables.

Data flow diagrams show the data working within the system. This is important since, after all, it's the data that the system is all about. The data flow diagram (which we will refer to as a DFD from now on) shows the flow of data between the various processes into which the system divides.

A DFD is a network diagram. Other, perhaps more familiar examples of network diagrams are electrical wiring or circuit diagrams, or even the London Underground map.

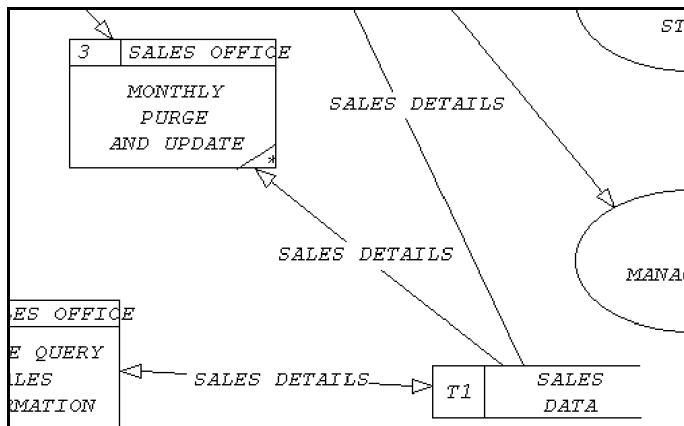
The use of DFD's is fairly recent to systems analysis, and occurs most frequently within structured analysis and design. However, the idea is not new - system flowcharts show data flow, while bubble chart, which have been around a long time are a similar idea.



The rest of the box describes what is happening in the process. the rule of thumb here is to keep the description as terse and meaningful as possible. Use an imperative verb and object, but make the verb specific. 'Process' and 'Update' are TOO VAGUE and give little clue as to what is meant. 'Calculate', 'Add' and 'Validate' give a clearer picture of what is happening.

Data Flow Lines

DATA FLOW LINES, represent a flow of data, which might be a single data item, a whole form, information from a file, a report, and so on.



Data flows represent any passage of data into the system, out of the system or between elements inside the system. It is represented by an arrow between the source and recipient of that data flow. In the world, it may take the form of a standard document with fixed content or a telephone call. It may be an enquiry, a functional document or a memo.

Wherever traceable data is passed, it must be shown by an arrow. At the highest level DFD, one arrow may represent several data flows, which may

be decomposed into the individual data flows at the lower levels.

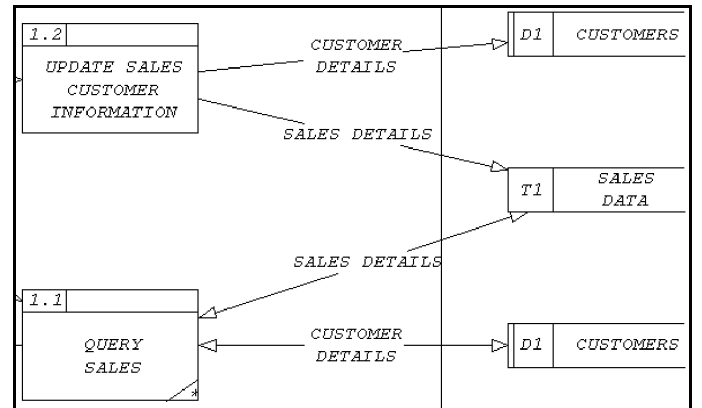
There are some validation rules about where data flows may or may not travel:

- **Data stores may not be linked by data flows: flows must travel from one to another via a process.**
- **External entities may neither send nor receive data flows directly to or from a data store: they must communicate via a process.**

Data cannot be generated by a process, nor be swallowed by a process; documents may be swallowed or generated, but there must be output that is related directly to all inputs to the process.

Data Stores

DATA STORE BOXES, represent the source of data or its destination. A data store is a place where data comes to rest. It may be a filing cabinet, or an in-tray, a card index, a reference book or a computer file.

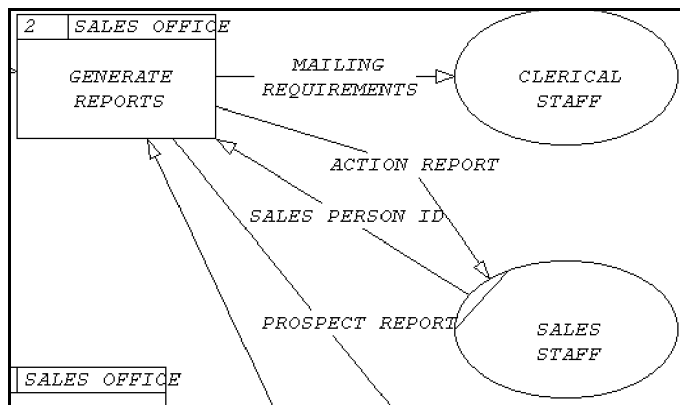


Anywhere that data is stored and retrieved is called a data store. The notation is simple: a long open ended rectangle, with a box at the left-hand end. The box is labelled with an Alpha prefix and a number. The alpha could be D (for an automated data store) or M (for a manual/card data store) or T (for a Temporary Transaction store).

If for the sake of tidiness in the diagram, you wish to show the data store in more than one part of the diagram, draw a bar beside the left-hand box. Each occurrence of the data store will display that bar.

External Entities

EXTERNAL ENTITIES are denoted by an ellipse shaped symbol which represents an external entity. External entities are those bodies outside the system boundary which interact with the system. They may be external to the whole company, such as Customers, Accountant, Supplier, Inland Revenue, or just external to the application area under investigation.



So if we were modelling a sales office system, accounts and despatch areas would be shown as external entities. Each external entity communicates in some way with the system, so there is always a flow of data between a process in system and an external entity.

The entities are labelled with a singular noun describing the role of the entity, eg Accounts, VAT Office, Credit Manager. Above the label will be an alphabetic character, again for reference purposes only.

As with data stores, it may be desirable for the sake of clarity to duplicate an external entity on the diagram, rather than have arrows from all points converging on one entity. If that is the case, put a small line along the top of the ellipse.

An important rule about drawing DFD's. This rule says that flowlines should indicate only data essential to the system, and should avoid cluttering up the diagram with inessential data-flow (you may not agree).

Preparation and Analysis needed in order to undertake Data Flow Modelling

Before trying to undertake Data Flow modelling, considerable analysis and study must be carried out in order to identify all the data used and produced by the system, all the processes performed by the system, and all the data repositories used by the system.

In preparation for being able to follow the Banking system case study, it is important that you read through the accompanying scenario and perform the following tasks:

1. Write down or describe as many jobs as you can identify from the scenario
2. Group related jobs together under one heading
3. Group together related groups of jobs under an even more general heading
4. Identify sources of information
5. Identify destinations of information

Practical Class Activity - A typical banking system

Read through the following description of part of a typical banking system

Among the many jobs undertaken within a typical Bank are **Maintaining Account Details**, **Maintaining Transaction Information** and the **Production Of Various Reports** as requested by both Bank Staff and Customers. Maintaining Account Details involves overseeing the opening of New Accounts, Editing or updating Existing Account information and also the closure of an account. Maintaining transaction Information involves a customer being able to make a deposit and/or a withdrawal and later on the actual processing of transactions (that is the updating of the various accounts which is done in a batch and is not an immediate task).

The job of processing reports involves amongst others Producing Reports for the bank and for the customer. Bank hierarchy require various reports including a list of customers in credit and/or overdrawn and a list of transaction between two given dates. The reports required by customers include a statement and an account enquiry slip.

1. Draw a Data Flow Diagram to illustrate processes and Data of the Banking System Described