LOGICAL SYSTEM MODELLING

DEVELOPING A LOGICAL MODEL

In the procedures of Structured Analysis (see Fig. 01) the first step is to derive the current system physical description consisting of DFD's, Mini-Specs, and a DD.

This description is likely to have many physical characteristics, both in the DFD's and the DD i.e. it describes how things are done as well as why they are done, the way things are done now rather than how they will be done in the new system.

The next step is to derive a logical system description i.e. to take the current system physical description as input and produce a current system **logical** description as output.

The third step is to derive the new system logical model by taking the current system logical description as input and adding to it the requirements for the new system.

The purpose of producing logical models is to design a new system which is based on the **real** business problem. Logical models help the analyst to:

* get a clear understanding of the purpose of a system;
* avoid re-implementing obsolete technology and procedures;
* get an unbiased model for making new technology decisions.

Stages in Analysis and Design
The processes of analysis and design to which the structured methods apply

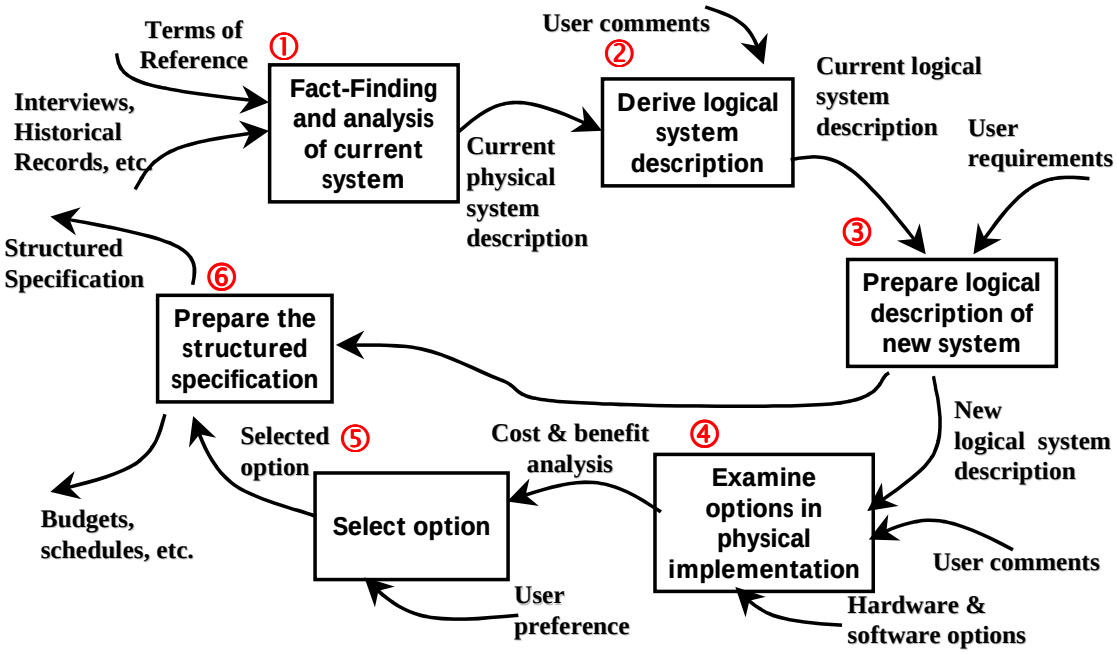


Figure 01

DERIVING THE LOGICAL SYSTEM DESCRIPTION

The transformation from current physical description to current logical description involves removing physical characteristics.

Removing these characteristics has the following effects:-

* DFD's may be altered;
* Composition of data flows may be modified;
* Data store definitions will be restructured to show logical composition rather than physical structure.

RECOGNISING AND REMOVING PHYSICAL CHARACTERISTICS

Work from the lowest level of the DFD set - the functional primitives, to remove physical characteristics.

The most obvious physical characteristics in a DFD appear as Dept./section names, physical locations, personal names, form names. These are usually easy to spot. As a first step, replace all physical names or processes, data flows and data stores with logical names.

Then look for features which fall into one or more of these four

* historical;
* political;
* procedural;
* equipment-based.

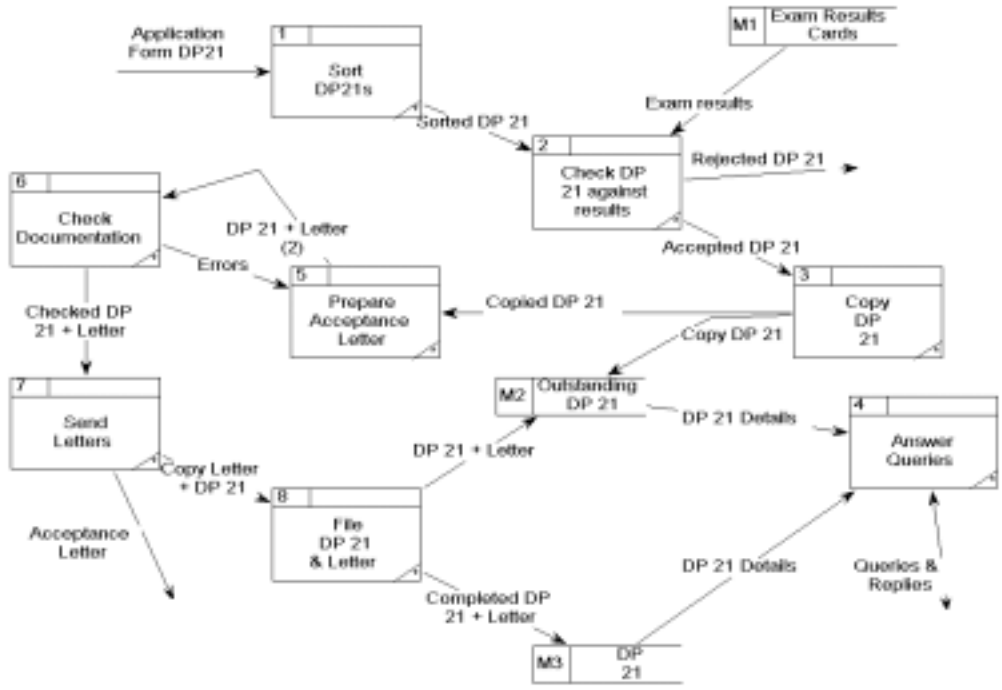
Historical: doing things a certain way solely because they have 'always' been done that way e.g. sending multiple copies of forms to various sections some of whom may no longer have any use for it, keeping the current part price list in the customer file because that is where it has always been kept.

Political: procedures carried out or responsibilities allocated solely for reasons of office politics e.g. the sales office incorporating the personnel section because the Sales Manager is empire building, sales representatives expense sheets routed to Accounts via the Production Manager because he/she is 'keeping tabs' on the Sales Manager.

Procedural: relates to data flows where the composition is determined by the physical medium e.g. a particular form with ten pieces of information on it being sent to a section which only requires one piece of information.

Equipment Based: where functions are described in terms of the equipment used to carry them out e.g. 'type invoices', 'batch orders for ABC computer.

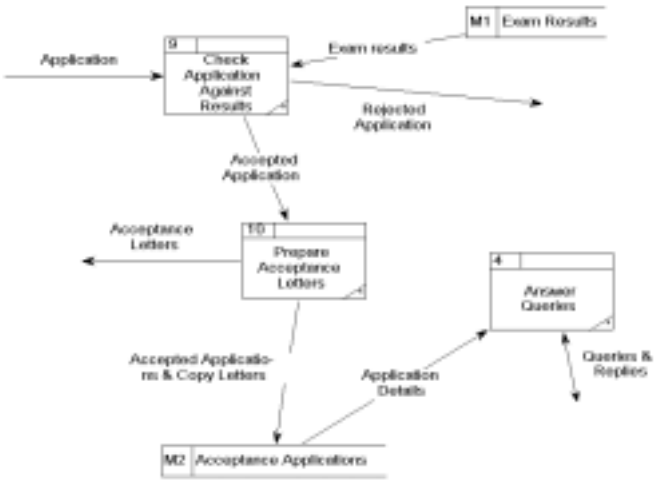
Remove any processes or data flows which are purely physical and therefore redundant on a logical DFD (sorting, copying, batching and geographic processing are physical). Figure 02 shows a DFD with redundant processes and Figure 03 shows the logical DFD.



Physical DFD
Figure 02

Replace physical processes which incorporate logical functions with the logical equivalent.

Question the purpose of each process - is it an essential business function?



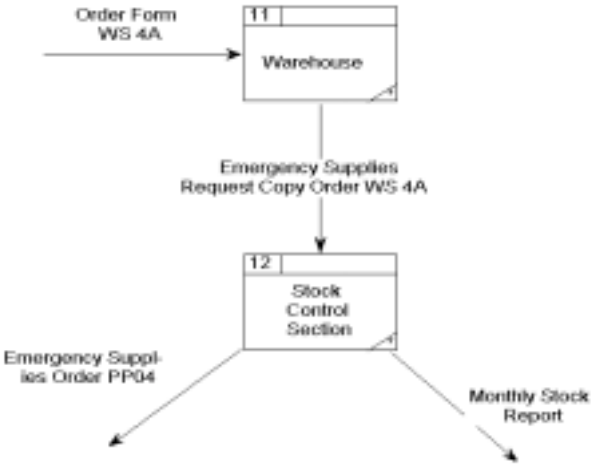
Conversion to Logical - Logical DFD
Figure 03

Partitioning

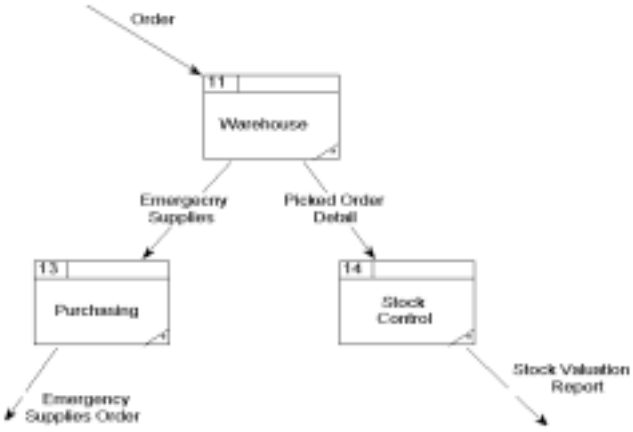
When drawing DFD's of the existing system, activities are usually grouped or partitioned to reflect the existing organisation particularly in the higher levels of the DFD set. For the logical model, the groupings of activities should be reconsidered so that each grouping has a clearly defined function and is as independent of other groupings as possible.

Working from the bottom level of the DFD set, repartition if necessary, e.g. functions may appear in different processes on a physical DFD simply because they were carried out in different physical locations, or vice versa.

Fig. 04 and 05 show an example of repartitioning a physical DFD into a DFD with logical groupings.



(The Purchasing Clerk works in the Stock Control Section)
Physical Current DFD
Figure 04



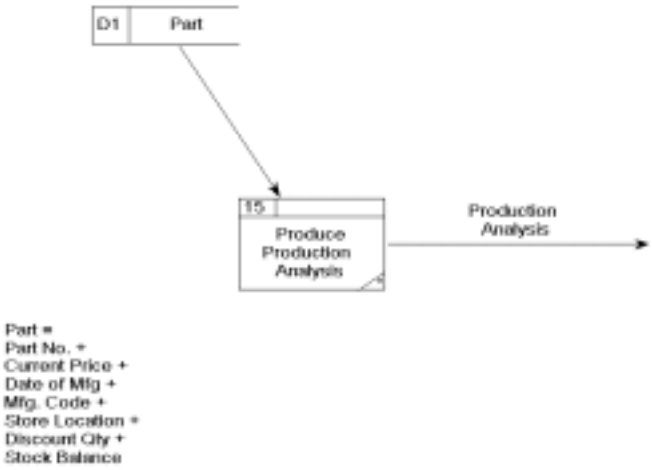
Conversion to Logical
Figure 05

Data Flows

The contents of logical data flows should be both necessary and sufficient i.e. the data flows into a process must contain all the data required to carry out the process and nothing more.

Fig. 06 and 07 show an example of a new logical data flow replacing a physical flow.

In making data flows more logical it may be necessary to make new entries in the DD to define new logical flows.



Physical DFD
Figure 06

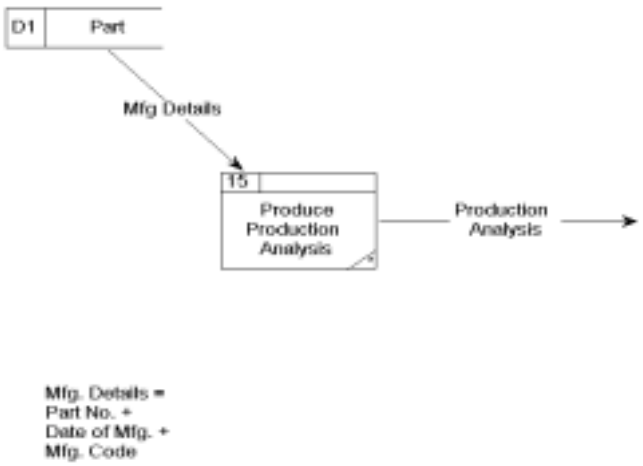


Figure 07

Data Stores

Since data analysis proceeds in parallel or slightly in advance of the process of producing logical DFD's the physical data stores are subjected to a process of logical simplification -

normalisation. The third normal form data structures replace the physical data stores on the logical DFD's.

There is usually a large number of such structures but some of these can be logically grouped together. These are usually very obvious e.g. ORDER and ORDER-LINE can be grouped as ORDER, STORE and STORE-ROOM can be grouped as STORE. Look for entities/structures which may be described by a single name or term, and/or are closely related, and/or are created together.

Mini-Specs

For each logical functional primitive a mini-spec must be done. If a process has not changed in the physical to logical transformation the mini-spec may not need to be changed but it must be examined to ensure that it is a logical statement of policy and that the data flows and data stores referred to are those which appear on the logical DFD's.

RECONSTITUTING THE DFD SET

When the physical characteristics have been removed from the lowest levels of the DFD, reconstitute the DFD set from the bottom up. Partitioning in at least some of the higher levels will change because the lower level processes have changed.

Data flows and data stores at the higher levels will be the new flows and structures taken from the lowest levels.

A characteristic of logical DFD's is that processes communicate with each other for the most part via data stores rather than directly via data flows. This is because such elements as physical media, physical boundaries, multiple copies and private files have been removed in the process of logicalisation.

WALKTHROUGH WITH USERS

Logical DFD's are not solely for the consumption of system analysts. They must be agreed with the users, since the analyst must be sure that in removing physical characteristics he/she has not removed an essential business function.

When walking through the DFD's with the users the physical details should be filled in orally, so that the model becomes recognisable to the users.

NEW SYSTEM DESCRIPTION

When the current system logical description has been completed the new system must be invented i.e. some or all of the deficiencies in the existing system must be corrected and removed and any new requirements added into the model.

Improvements required to the current system might be such as (a) 'improve servicing of outstanding orders' or (b) 'reduce delays in updating the stock file'.

New features might be (c) 'handle multiple items per order' (in a situation where at present

each item requires a separate order), or (d) 'provide sales analysis on demand' (no analyses done at the moment), or (e) 'store and sell by product grade rather than by overall category' (in a scheme where product is bought from the supplier on a grade basis but sold only on a broader category basis).

During fact finding the analyst will have discovered many deficiencies in the current system and users will have described problems. It is vital for the analyst to develop specific definitions of requirements. Improvements to the existing system, in particular, are often expressed in vague terms e.g. in (a) above, what specifically constitutes an improvement?, in (b) is there a level of delay which is acceptable or must the file be updated simultaneously with the stock movement?

The requirements of a system are closely associated with the critical success factors for the business as a whole.

CRITICAL SUCCESS AND PERFORMANCE FACTORS

In some organisations a formal approach is taken in analysing what are the critical success factors for the organisation i.e. the areas in which it is essential to be effective, and this approach is also applied at a lower level in determining what are the key performance factors for particular systems.

The factors vary depending on the nature of the business. For an organisation such factors might be 'profitability', 'productivity' etc.

For a particular system the factors to be considered are those which influence the achievement of objectives in the application area. Each factor identified will be used to set system performance targets and detailed information requirements. Each factor must have a measure of performance i.e. specific criteria which will enable a judgement to be made as to whether the objectives are met.

For a system involving sales to customers such factors and measures might include:-

Performance Factor	**Measure**	**Target**
Billing Procedure	%age billed within two days of despatch	100%
Accuracy of Bills	age of errors	None

An analysis of critical success factors and/or system performance factors may have been carried out during a Feasibility Study, if such a study was undertaken.

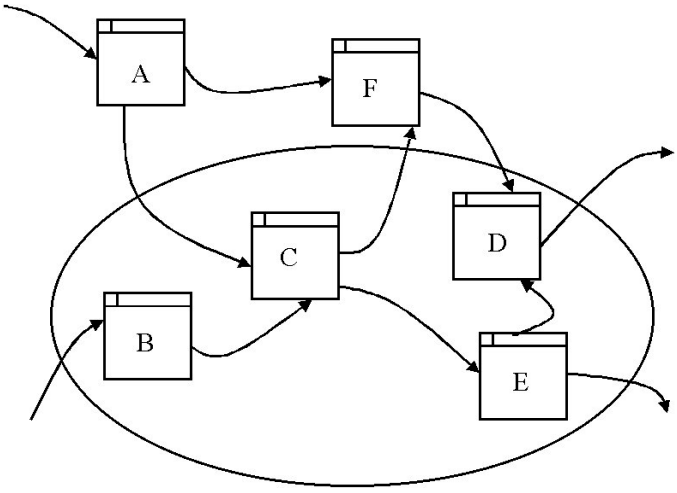
Whether formally developed or not, the critical success factors for the business and the key performance factors for the application must be translated by the analyst into a specific set of requirements for the new system, with criteria established to judge whether system objectives are being met.

DOMAIN OF CHANGE

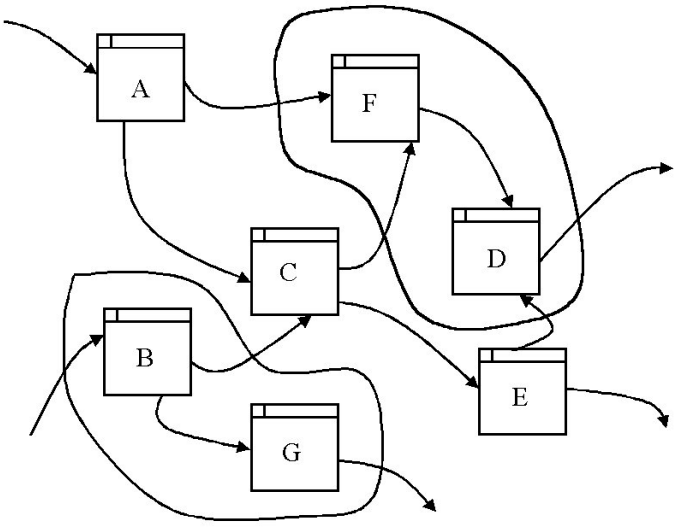
The current system logical model almost always contains processes which will not be affected i.e. will not require to be changed, by the requirements for the new system. The analyst must now determine which processes will be affected and which will not.

Work from the lowest level of the DFD set and examine each mini-spec to see whether the process will be changed in the new system. Draw up a list of processes which will be affected and a list of those which will not. Then, on the DFD's, draw a line around those processes which will be changed, as shown in Fig. 08.

This line is the system boundary and the area inside the line is known as the Domain of Change. The domain of change need not necessarily be only one area (see Fig. 09).



Domain of Change
Figure 08

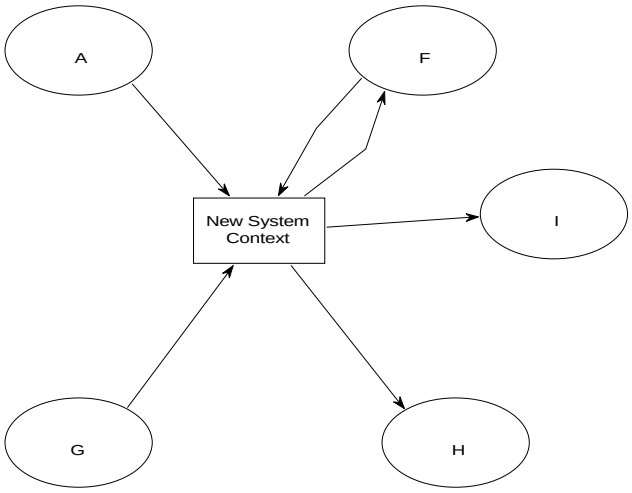


Domain of Change
Figure 09

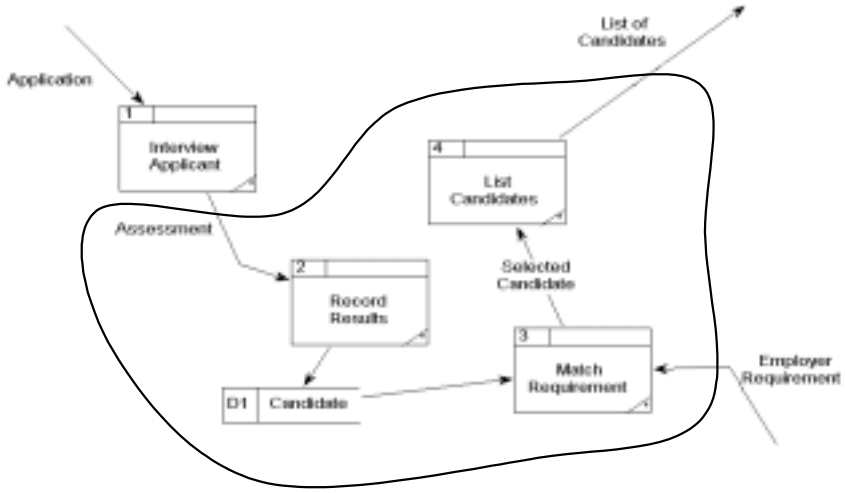
The DFD with this boundary line is actually the context diagram for the new system (Fig.10).

Fig. 11 shows an example of a Domain of change marked on a DFD.

The lines crossing the boundary are the inputs and outputs of the new system.



Context Diagram for New System
Figure 10



Domain of Change - Recruitment
Figure 11

CREATING THE NEW LOGICAL SYSTEM

Once the previous steps have been carried out the new system is 'invented' i.e. the analyst must use his/her creative skills to develop the logical model of the new system.

The starting point is the context diagram already identified. Partition from the top down, think logically, not physically.

Add new processes to cover new features.

Analyse New Data Structures

The requirements for the new system will almost certainly involve data structure changes - data elements will either be added to or deleted from data flows and stores, and completely new input and/or output data flows will be required.

Retrace the normalisation process on these new flows and add the data elements to the appropriate structure or new structures created. The result is a complete set of normalised structures for the new system.

Validate the logical model:

* ensure that all necessary activities are being performed;
* all outputs are being produced;
* all inputs are being processed;
* all new information requirements are being met.

The output from this stage is:-

* a set of DFD's for the new system;
* supporting mini-specs for lowest level processes;
* updated DD containing definitions of any new data flows required in the new system.