An Introduction to
Structured Systems Analysis and Design Method (SSADM)

Why have Systems Analysis?

It is often difficult to explain what is achieved by Systems Analysis and Design - especially when talking to a user who wants a system tomorrow. The user may have his own home computer or has a friend with one and knows that a system can be developed which will solve a problem in a matter of hours. If systems are that easy to develop, why do systems in the business environment take so long, need these special methods and become very expensive?

To give a simple example consider that you have a plot of land, wish to build a house, and are a keen DIY person. You could go to a builders' merchants, buy all the materials and build It yourself. You may find the walls insufficiently strong to hold the roof, drains are insufficient etc. You FAILED to plan what you were going to do.

Alternatively you could have requested a builder to erect your house. He knows how to build walls with the correct strength, lay drains and install electricity. Do his ideas of size and shape correspond with yours?

To obtain the house of your dreams, you must define your exact requirements in such a way that both you and the builder are happy that you understand what you want is going to be built and that nothing has been overlooked or misinterpreted. You would use an architect to draw plans (pictures/diagrams) of your exact requirements and superimpose on them all the rules and regulations for building houses. The builder would take this plan and produce a house to these specifications.

This simple example illustrates the need to specify the exact requirements of the house before construction begins. This principle applies to computer systems.

If the architect is replaced by the systems analyst and the builder by the systems builder (programmer?) the analogy Is appropriate. The systems analyst discusses with the user his requirements to identify what is needed from the new system and produces a required system specification. After consultation with the user the analyst, together with the systems designer, produces several solutions for the user to choose from. Once the choice has been made the system can be built.

We have identified those major requirements of good system design:-

User Involvement

Quality Assurance

Separation of logical and physical specification.

Why use a Structured Method?

Structured methods share these characteristics:

- they structure a project into small, well-defined activities and specify the sequence and interaction of these activities:

- they use diagrammatic and other modelling techniques to give a more precise (structured) definition that is understandable by both users and developers.

Why can a systems analyst not use his or her experience and just ask all the right questions? Some of the advantages of structured methods are given here:

Structured analysis provides a clear requirements statement that everyone can understand and is a firm foundation for subsequent design and implementation.

Part of the problem with systems analysts just asking 'the right questions' is that it is often difficult for a technical person to describe the system concepts back to the user in terms the user can understand. Structured methods generally include the use of easily understood, non technical diagrammatic techniques. It is important that these diagrams do not contain computer jargon and technical detail that the user won't understand - and does not need to understand.

More effective use of experienced and inexperienced staff.

Another part of the problem with this approach is the availability of staff with enough experience to ask all the right questions. A structured method does not remove the need for experienced staff, but it does provide the option of spreading the experience more evenly. The use of structured techniques means that certain tasks can be delegated to inexperienced staff who can then be guided by the more experienced.

Improved Project Planning and Control.

The use of a structured approach makes for more effective management of projects. Splitting a project down into stages and steps allows better estimation of the time taken to complete the project. Also, by following a detailed plan, it will be possible to detect slippage as it occurs and not just before the system is due to be implemented.

Better Quality systems.

By making the specification very comprehensive, it is possible to ensure that the system built will be of a high quality. The use of structured techniques has been found to lead to a system that is very flexible and amenable to change. Within SSADM, users participate in formal quality assurance reviews and informal "walk throughs" and 'sign off' each stage before the developers progress to the next. This means that the analysts can be confident that the new system will meet the user's requirements before it is built.

SSADM

Background

Structured Systems Analysis and Design Method (SSADM) was selected by the UK government as the mandatory method to be used on government projects. It Is continually being enhanced in line with the needs of the user. (Appendix 1 is an extract from version 4. Although the government funded the initial development for their own use it is open' to all to use without the need for a license or any other form of permission.

Why Choose SSADM?

We discussed earlier the need for a standardised approach to designing systems under the guise of "Systems Analysis and Design". To achieve this we identified the need to talk regularly to the user, document our findings and obtain the approval for the system to be developed to that specification.

This has identified the milestones to be achieved but has not identified any methods of reaching milestones. There are analysis techniques available, which were developed independently, which help to solve some of the problems. SSADM has taken these techniques, put them into a framework which defined sequence and cross checking of the information obtained.

SSADM has been developed to cover six major stages of analysis and design.

SSADM stands for Structured Systems Analysis and Design Methodology and meets all the requirements of a good systems development method. It is an integrated set of standards (Figure 1) with three components:

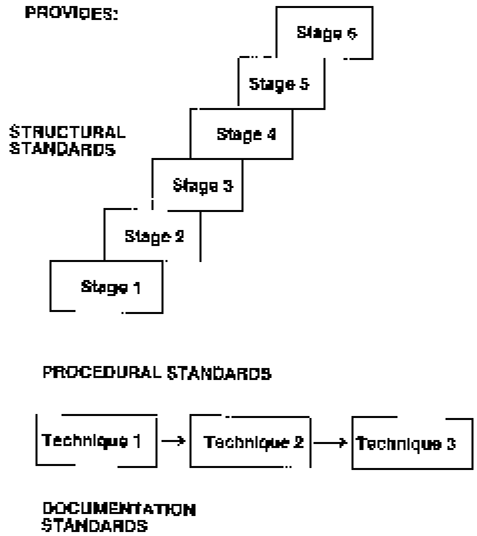


Figure 1-SSADM Standard

1) Structural Standards

   Working on the premise that every development project must have some structure of development stages and requires the same series of tasks to be carried out within those stages, a standard framework has been designed of six stages, each with its explicitly defined tasks. The interfaces between the tasks are defined, as are the appropriate products or deliverables of the tasks.

In support of the main framework there are:

2) Procedural Standards

   Consisting of specific techniques, each with detailed rules and guidelines on when and how to use them,

3) Documentation Standards

   Specific documents are available, enabling the products of the systems development to be recorded at the appropriate stages.

Before we consider the six stages let us consider the stages of a Project Life Cycle.

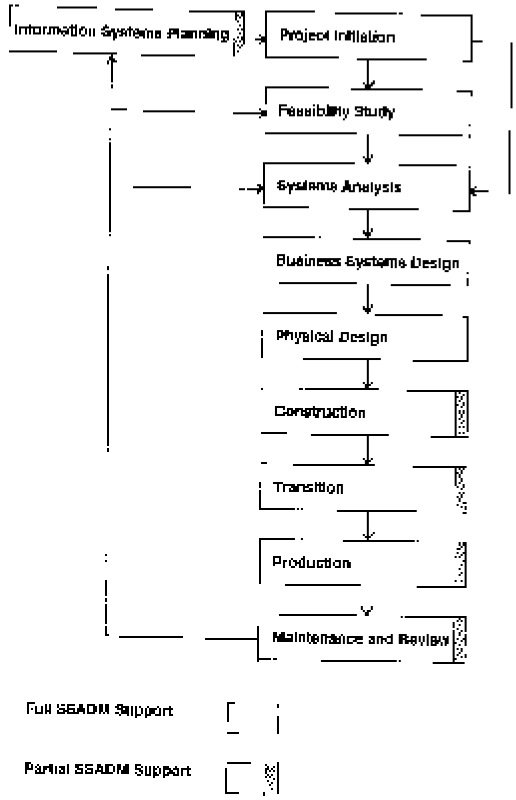


Figure 2-Life Cycle Stages

Information Systems Planning

Many organisations, recognising the contribution of information systems to their success, have invested in strategic planning for the development of future and existing information systems. Recently many methods for information systems planning have been put forward. These take a variety of approaches but generally the result from a planning exercise will be an analysis of the organisation's present position, recommendations as to which systems should be developed or enhanced, a plan showing the order in which these projects should be done, and outline project plans and terms of reference for each project. Two of the techniques used by SSADM - Data Flow Diagrams and Logical Data Structures - are used, in some form, by many of these methods. Many projects which have used SSADM have been initiated by an information systems planning study. To this extent SSADM offers partial support to this activity. There is at present no information systems planning component within SSADM, although it is likely that one will be developed for future versions of the method.

Project Initiation

This is the phase where the project is set up, terms of reference agreed, team members assigned, and plans drawn up. SSADM provides detailed guidelines for this activity.

Feasibility Study

This is the phase where it is decided whether the project is technically possible, whether it can be financially and socially justified, and whether the new system will be accepted by the organisation. SSADM provides detailed guidelines on the conduct of feasibility studies detailing the steps and stages required.

Systems Analysis

Here the current system is analysed in great detail to determine the requirements for a new system. SSADM does not give guidelines on such basic systems analysis skills as interviewing and other data collection methods but provides the means of recording and analysing the results of the investigation. Stage 1 of SSADM deals with the analysis of the current system and Stage 2 specIfies the requirements for the new system.

Business Systems Design

The requirements for the new system will have been broadly specified in the previous phase. In this phase various technical solutions that meet the requirements are evaluated and one selected. A detailed logical design of the new system is developed which shows clearly, in a non-technical way, how the new system will operate within the business. This phase is dealt with by SSADM Stages 3,4 and 5.

Physical Design

The logical design is converted to a design that fits the computer hardware and Software selected. This is known as the physical design and is dealt with by Stage 6 of SSADM. Physical design involves the specification of files (or database definitions), the specification of programs and the detailed operating and manual procedures that support them.

Construction

This concerns the programming, the assembly of programs into a system, and the testing of the system. SSADM does not address this phase. However, the plans for system building and testing are laid in SSADM Stage 6. Many projects are now using fourth-generation environments for systems development and have integrated their use into Stages 5 and 6 of SSADM.

Transition

This phase Involves the transition from operating the old systems to operating the new. It Involves the installation of equipment, the conversion of old system data to formats required by the new system, and the training of users. Some systems' life cycles join the construction and transition phases together to form an Implementation phase. SSADM does not fully address the transition phase although the plans for it are developed in Stage 6.

Production

This phase begins when the system has been completely handed over to the users. The term production conveys that the system is operating and producing the information that was required of it. This activity is not supported by SSADM.

Maintenance and Review

Throughout the production phase the system will require maintenance in various ways; correction of errors, adaptation to new software and hardware releases, and minor enhancements. The system will need to be reviewed to show how well it has met the requirements and objectives set for it and whether it continues to meet the users' requirements. These enhancements and reviews may lead into further system studies as shown in the diagram. Guidelines are available as to how SSADM can be employed in a maintenance environment.

3 **SSADM in the Project Life Cycle**

3.1 Major phases

SSADM addresses the pre-prog ramming stages of jhe system development. The two major phases involved are *Systems Analysis*. defining what has to be done, and *System Design* defining how it will be done. Many projects have a preliminary feasibility phase for which SSADM can also be used.

3.2 Input to SSADM

Every organisation is likely to have its own view of where computer systems development fits into its structure and operations. SSADM, as a generally applicable development method, must be able to cope with widely differing project circumstances. There is, therefore, no detailed specification for the "Start-up Document" which is the input to SSADM. It can be as simple (and as vague) as a few lines broadly outlining the area for investigation, or as detailed as a preliminary study which may have involved some man-years of effort to produce. Generally the more detailed the "Start-up Document", the less effort is required In the Analysis stage.

3.3 Output from SSADM

The final products from an SSADM project are:

- Program specifications and file/database designs for the target environment;
- Manual procedures;
- Operating instructions;

- Implementation plan.

3.4 The Analysis stages

The objective of Analysis is to create a detailed specification of user requirements, and to agree with users the levels of service and performance required. The stages are:

- Analysis of systems operations and current problems;
- Specification of Requirements, in terms of what the system must provide rather than how It will work;
- Selection of Technical Options, in which the user selects a system for implementation from a menu of possibilities, each having differing service levels, costs/benefits, development implications.

3.5 The Design Stages

The objective of Systems Design is to define the system to provide the required services, and formulate a plan of how it is to be implemented. The stages are:

- Data design, in which the detailed logical data structure and content Is defined;

- Physical design, In which the conversion of the logical design to one for implementation on a particular hardware/software environment is specified and refined to meet performance requirements.

3.6 SSADM and the overall project plan

It should, at this point, be emphasised that whilst the stages and steps appear to form a strict sequence, some tasks may well begin earlier In the method that their position in the sequence would suggest. This is particularly true of Stage 6 where It is recommended that work should begin on such tasks as the creation of Systems Test Plans, Implementation Plans and the definition of manual procedures as early as possible. The earliest point when these can begin is typically at the completion of Stage 3. This work can therefore run in parallel with Stage 4 & 5 to eventual completion in Stage 6. It should also be emphasised that SSADM does not encompass every aspect of the project and that other activities must proceed alongside those covered by SSADM, amongst which examples would be: procurement of hardware and software; site preparation; user education etc. SSADM works within the framework of the Project Plan. It is not a substitute for it.

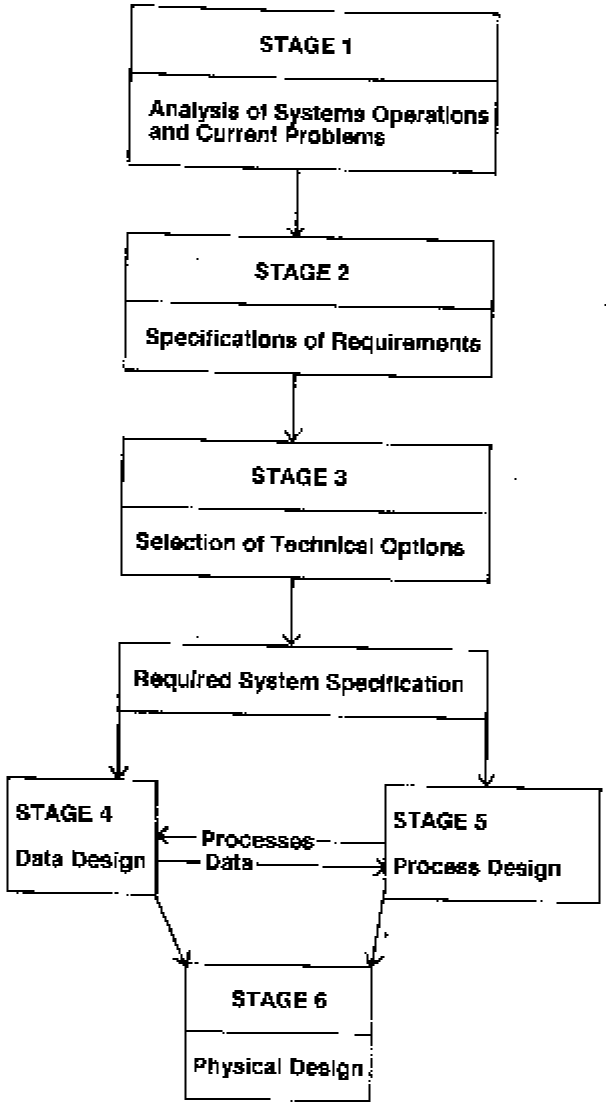


Figure 3- SSADM Stages

KEY TECHNIQUES IN SSADM

Three views of the system are taken and these are used for checking against each other.

Data Flow Diagrams (DFD)
Logical Data Structure (LDS)

Entity Life History (ELH)

We will briefly consider each of these methods.

1) **Data Flow Diagrams**

Pictorial representations of systems have long been acknowledged as being more effective than narrative.

Conventional flowcharts have been used for many years, with considerable advantages over narrative. Here are some examples:

- System flowchart (for high-level system description)
- Clerical procedure flowchart (for low-level description)
- Computer run chart
- Program logic flowchart
- Interactive system flowchart

While of undoubted value, these types of flowchart have a number of disadvantages for analysis documentation:

- They confuse the logical aspects of the system (what is done) with the specific physical implementation (how it is done). They do not provide a mechanism for separating these two views.

- They do not allow any easy transition from high-level "broad brush description" to detailed low-level description.

- They are "procedure orientated", placing the greatest emphasis on procedures in the system, rather than on data which appears only occasionally on the flow chart.

The use of Data Flow Diagrams (DFDs) removes or lessens these disadvantages. There are still conventions to be observed, as there are with flowcharts, but everything in a DFD has an exact equivalent in the activity being described.

It is therefore much easier for the user, the systems analyst and the designer to relate the diagram to the activity being carried out, or desired to be carried out. The DFD is also more suited to high-level descriptions than the flowchart.

Systems design has borrowed from the traditional approaches of engineering to develop techniques known as structure design.

Structured analysis (and the DFD is an important technique within this) carries the concepts of structure design into the analysis phase.

Consider figure DFD.1. This is the Initial Data Flow Diagram which identifies the major areas and data flows which are within the Terms of Reference. This diagram identifies the functions which are to be reviewed together within the sources of input and output for those functions.

Each of the functions is further sub-divided as shown In figure DFD.2. In this diagram files of information are identified. The process identified here can be further sub-divided as shown in figure DFD.3. These processes may be sub-divided again if necessary.

Ultimately a brief description of the process can be written using any of the possible documentation techniques, such as Decision Tables or Structured English.

These diagrams will be drawn on the basis of the fact finding exercise and when complete will form the definition of the "Current Physical System".

Eventually the diagram will be re-drawn, removing the physical aspects to show what is done, not where or by whom.

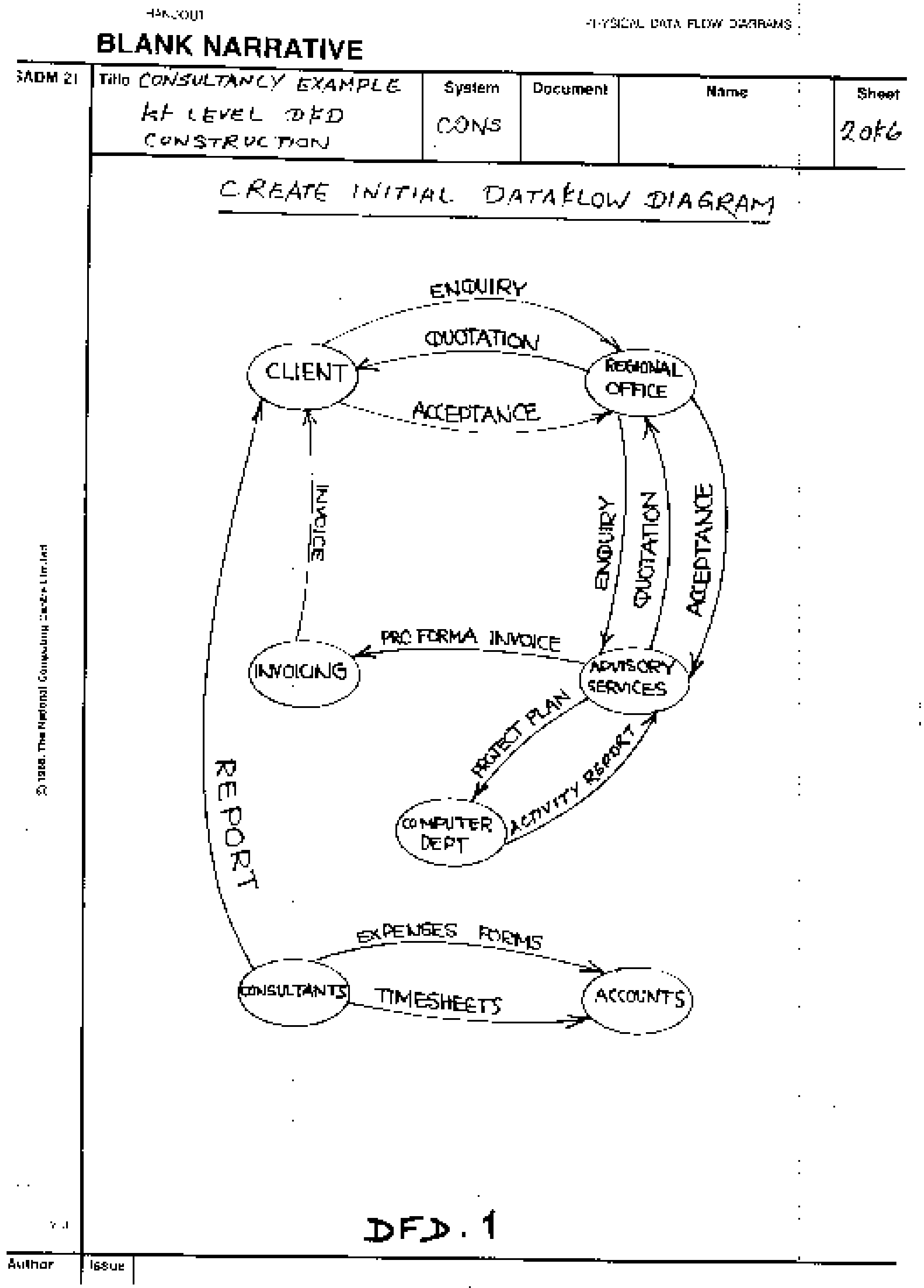


Figure 4 - context Diagram

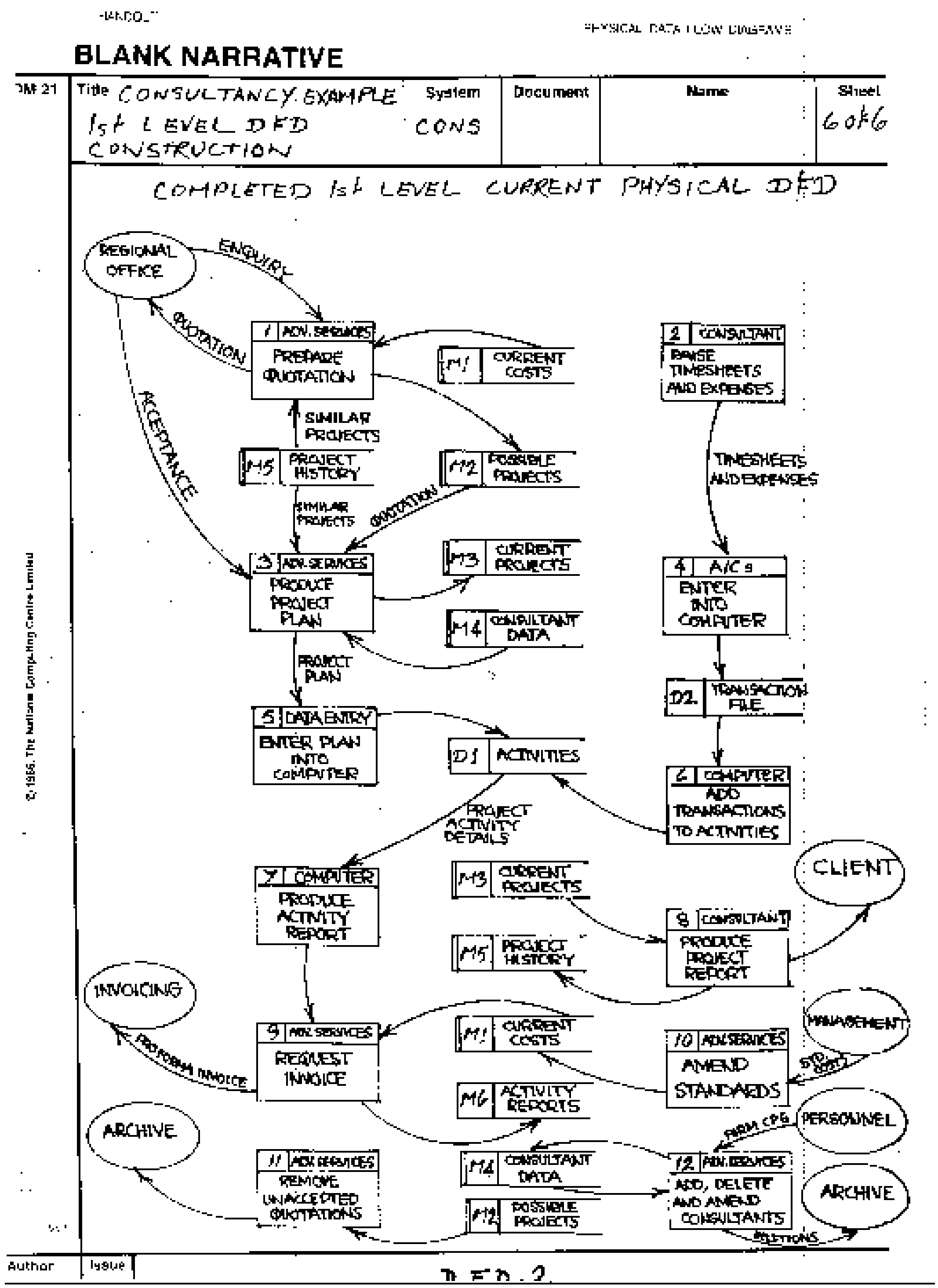


Figure 5 - Level 01 DFD

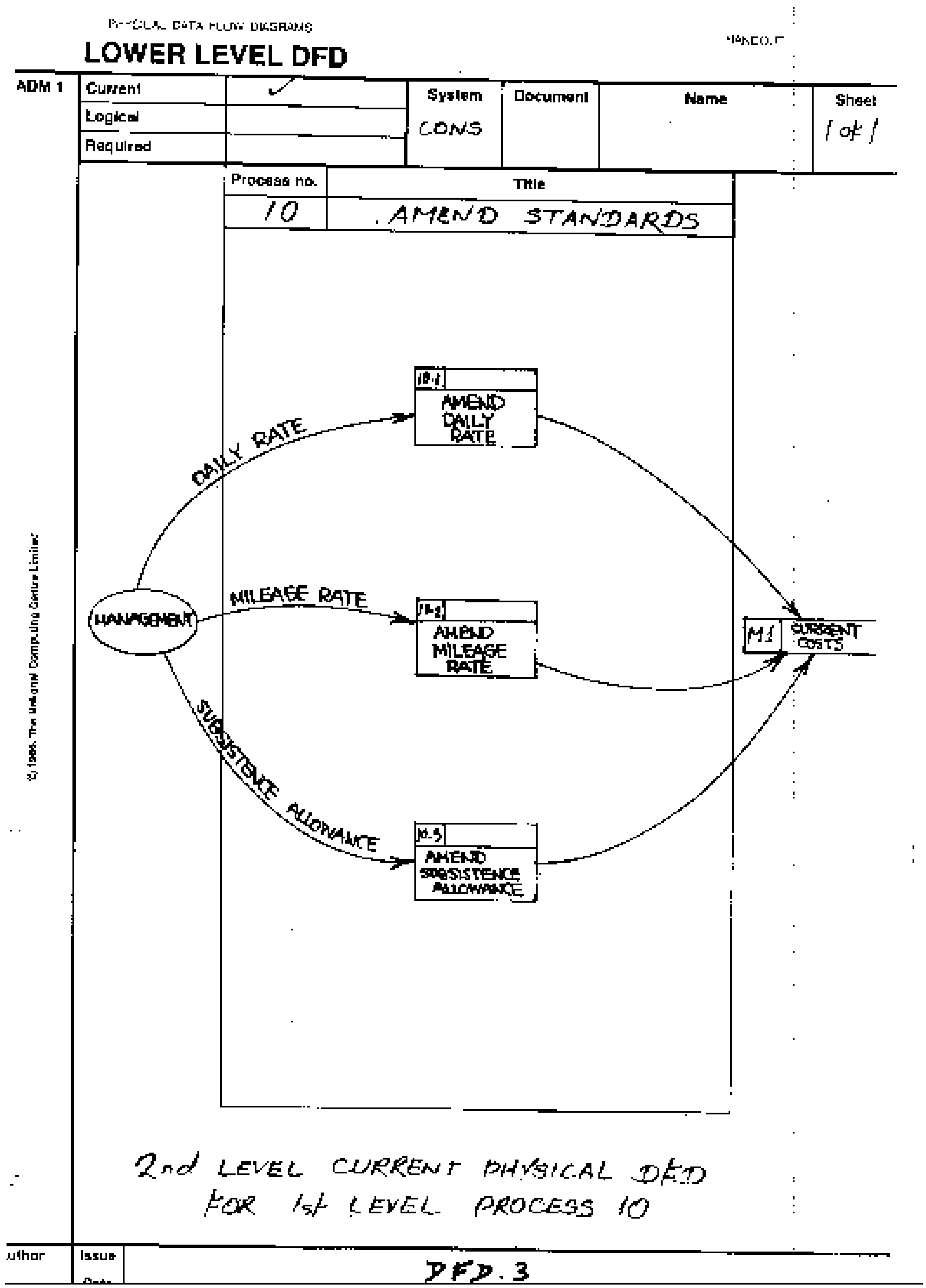


Figure 6 - Level 02 DFD

2 Logical Data Structure

2.1 Definitions

The pictorial representation of entities and the relationships between them is called an Entity model. A current system Logical Data Structure (LDS) is a form of Entity model which represents entities and their relationships in the current system environment.

2.2 Entities

For those unfamiliar with the term "entity" its nearest equivalent in more familiar terminology is "file". The difference between the two is best understood in the context of producing an LDS, so a formal definition of an entity will be delayed until the method of producing an LDS has been described.

A working definition, suitable at this stage, is "something about which an organisation keeps data". An entity may be physical, eg. person, building,. machine etc, or conceptual, eg. cost centre, project, etc.

Other terms which are used in the construction of an LDS are "entity occurrence" and "attribute" or "data item". The latter two are synonyms. Terms used in LDS with their physical data storage equivalents are:

Entity - file
Entity occurrence - record
Attribute or data item -filed

2.3 Purpose

Files in many existing computer systems cause problems because they do not reflect the structure of the data within the system. As a result it is often necessary to produce subsidiary files to enable the processing to be carried out, and additional processing demands cause the creation of yet more subsidiary files.

For this reason it is essential to determine the structure of the required system's data before the design of physical files. The production of a current system LDS is the first stage in this process.

A second reason for producing a current systems LDS is because it provides a view of the system in terms of the way its data is organised, therefore supplementing the functional view of the system provided by physical system DFDs. The two give a more complete definition of the system than either can separately.

The "Current Logical Data Structure" is drawn from the same information as that used for the DFD (see Figure 07). The contents of the symbols represent those entities which are significant to the areas of the company under consideration. To supplement the diagram is a list of attributes for each entity identified (see table 01). The lists at this stage are not exhaustive.

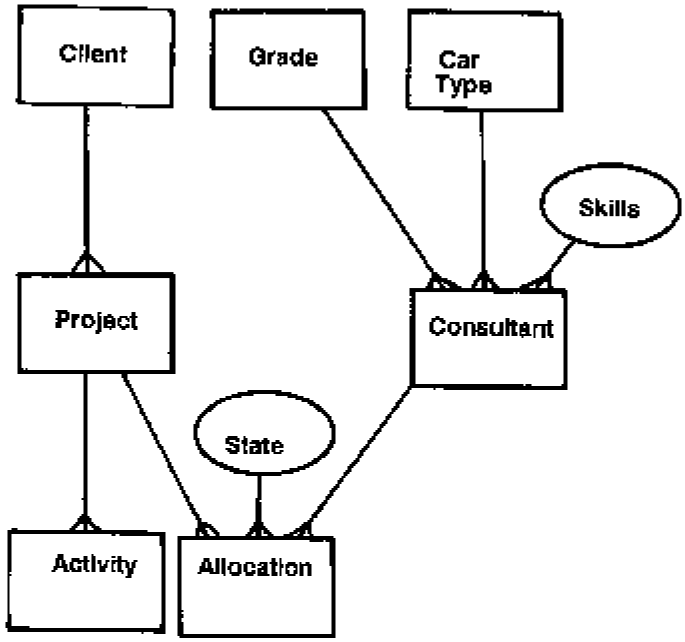


Figure 7 - Logical Data Structure

ENTITY DESCRIPTION / OPTIMISED TNF RELATIONS

Title	LDS	T	System	Document	Name	Sheet
Consultant	TNF		CONS			1 of 1
	CLDD					

Description	Volumetrics	
	Average	Maximum
Information relating to each internal and external consultant	50	60

Key	Data items	Format	Length	Comments
T	Consultant No	999	3	1 - 50 currently
	Name	X(30)	30	
	Address	X(30)	30	Occurs 4 (4 line of address
	Grade	X	1	
	Salary scale	X9	2	
	Car Type	99	2	
	Skill Description	X(20)	20	Occurs 10 max
	Qualifications	X(10)	10	Occurs 10 max
		Total Length	458	
Author			Issue	Date

Table 01 - List of Attributes

3) Entity Life History

For an analyst to understand a system thoroughly it is necessary to examine it from a number of different viewpoints. Two such viewpoints are represented by Data Flow Diagrams (DFDs) and Logical Data Structures (LDSs).

DFDs look at the system from the standpoint of its processes, indicating the relationship between processes in terms of the data passing between them and to and from data stores and external entities.

The LDS, on the other hand, looks at a system from the point of view of Its data, indicating the way in which data items are grouped together into entities and specifying the relationships between entities.

However, these two views of a system are static. DFDs do not necessarily show the sequence in which processes are performed, and an entity model does not indicate how the model changes with the passing of time. For example, the content of entities will change as entity occurrences are added to them or deleted from them, the values of attributes within entity occurrences will change and the relationships between entities will be created or deleted. DFDs and LDSs, even together, provide only a partial representation of a system.

Entity Life Histories (ELHs) look at a system from a third viewpoint, providing a means of representing how entities change within a system with the passage of time. ELHs start with the creation of an entity, record the sequence of changes which take place during its life within the system and end with its removal from the system.

3.1 ELHs and processes

Before describing Entity Life Histories, there are a number of concepts which need to be understood. ELHs are used to help define the logic associated with update processes, so an examination of processes provides a starting point to describe ELH concepts and terminology.

3.2 Update processes

In SSADM, processes are divided into those which result in changes to the system's data, called update processes, and those which do not result in changes, called enquiry or retrieval processes. ELHs are concerned with update processes only.

There are three major steps to the life of an entity and these are below

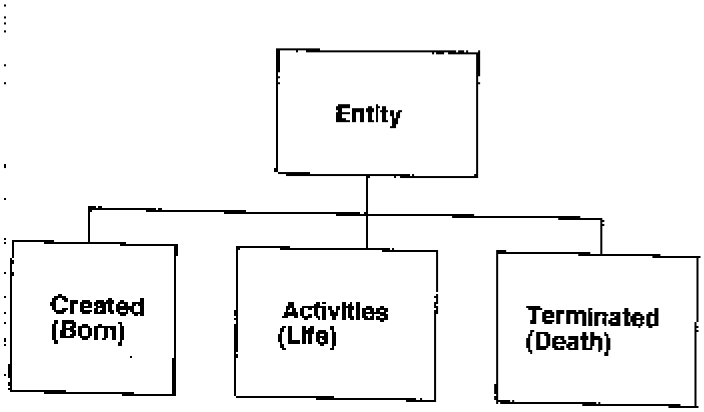


Figure 8 - ELH

Fig ELH

These activities may be sub-divided. The example ELH.2 shows the life history of the entity "Consultant" and in particular the two sorts of detail changes - grades and addresses. The diagram indicates the sequence of changes (if applicable) so that invalid combinations may be eliminated as errors in subsequent processing.

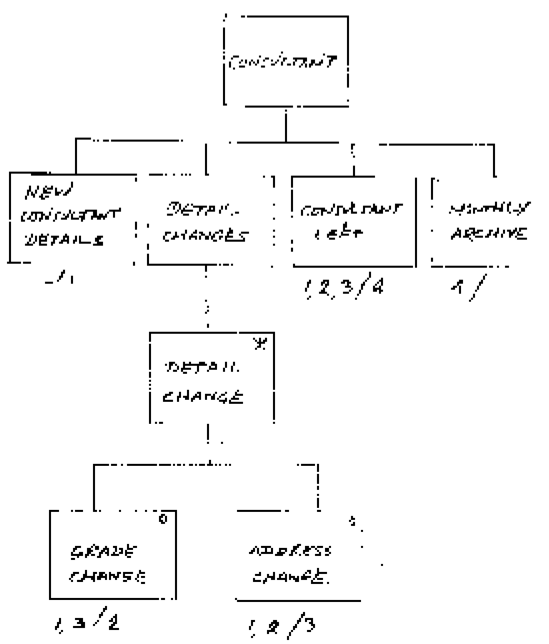


Figure 9 - Entity Life History

A SUMMARY OF THE SIX STAGES OF SSADM

STAGE 1 - CURRENT SYSTEM

The first stage of SSADM is concerned with the analysis of current systems operations and current problems.

At the end of this stage we should have a complete understanding of:

- Problems and requirements
- Current data flows and processes
- Current data

and be able to satisfy users of the completeness and accuracy of our findings

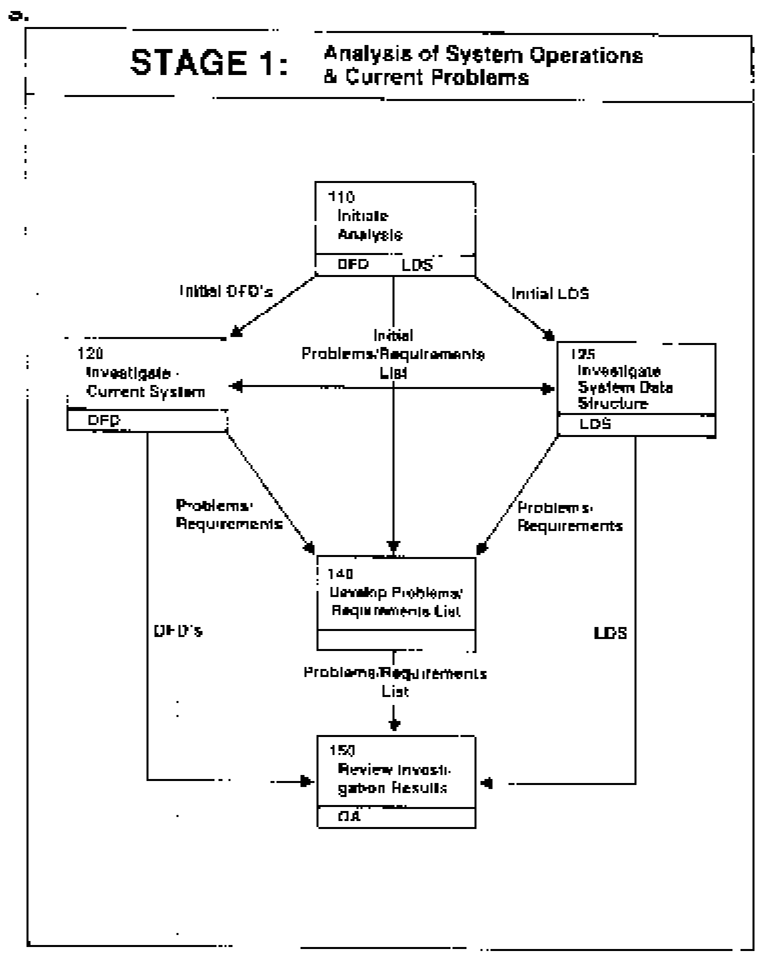


Figure 10 - Stage 1

In practice steps 120,125 and 140 are undertaken in parallel. Step 120 will be the step which requires most efforts in the Analysis phase, particularly where little of the operation of the system is known to the development team.

Step 110 will provide the management control reports for the project ie.

Project Task list and time estimates
Elapsed time schedule
Interview Plan
Project Control instructions
Quality Assurance mechanisms

STAGE 2- SPECIFICATION OF REQUIREMENTS

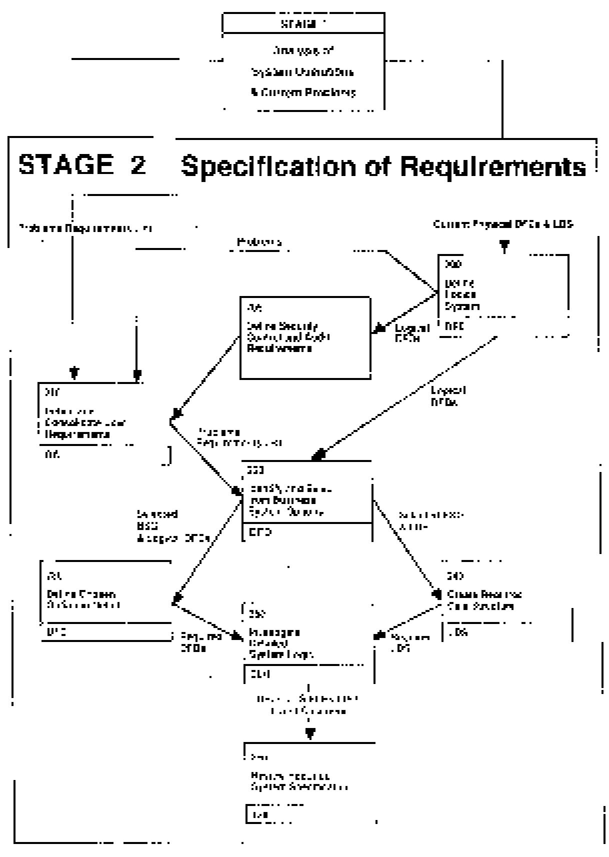


Figure 11 - Stage 2

The "Current Physical" system definition is converted to a “Current Logical" system definition and to this is added logical solutions to the problems and requirements which have been identified to give the user several possible solutions. Once the user has made his choice further detailed work may be carried out. A logical DFD of the agreed system is to be drawn down to the lowest level, the LDS of the data created and the ELH for each identified entity drawn.

We now have a detailed logical description of the required system.

STAGE 3-SELECTION OF TECHNICAL OPTIONS

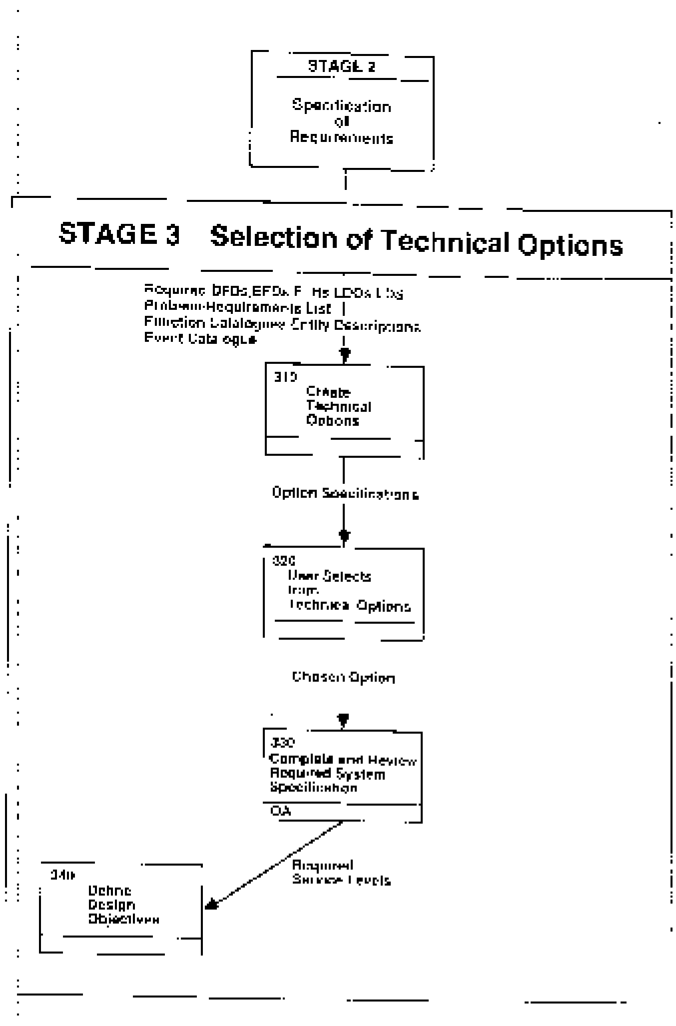


Figure 12 - Stage 3

We have given the user the choice of features for his new system to produce the "Logical" definition of the new system, we now provide alternative physical solutions to meet the requirements. To assist the user in his choice a cost/benefit summary is produced for each physical solution.

Once the user has made his choice the Required System Specification is completed.

This signifies the end of the analysis phase of the project and theoretically all the documentation produced could be passed to a new team to design and implement the system.

STAGE 4- Data DESIGN

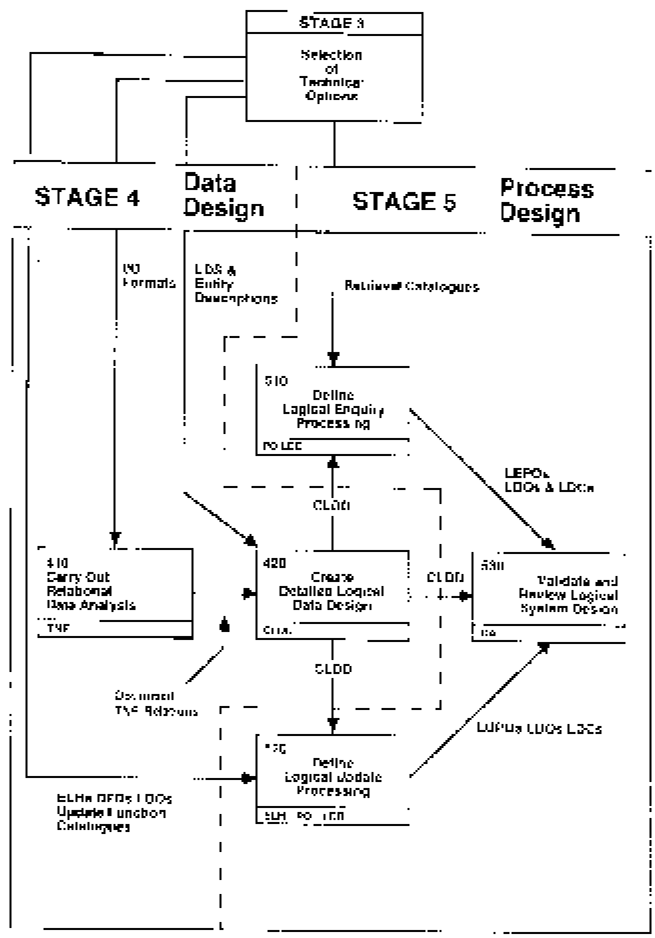


Figure 13 - Stage 4

This stage analyses the data requirements of both input and output which is then compared/merged with the Logical Data Structure to form the Composite Logical Data Design (CLDD). This stage is done in parallel with Stage 5 (process design).

STAGE 5- PROCESS DESIGN

Completion of Stage 5 marks the end of the LOGICAL DESIGN. The completed logical design will be "signed off" by the user and formal authorisation to proceed to Stage 6 - Physical design - obtained. Figure 14 summarises the steps in Stage 5.

This stage expands upon the detail entered in the Required System Specification. Stage 5 parallel Stage 4, Data and Processes are inter-related.

NB. Although Relational Data Analysis removes duplication of data, processing requirements may force some duplication in order to meet performance targets.

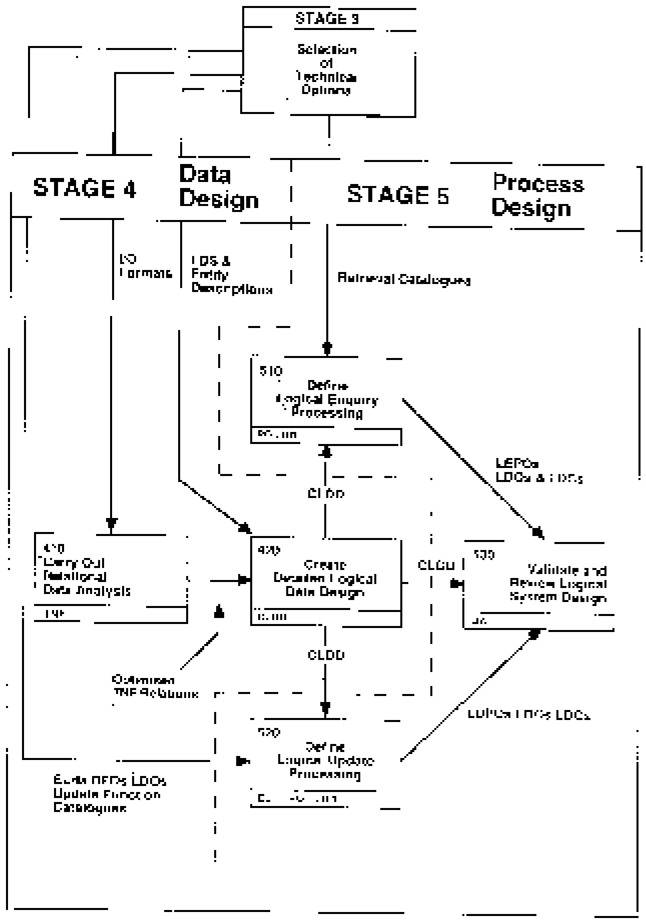
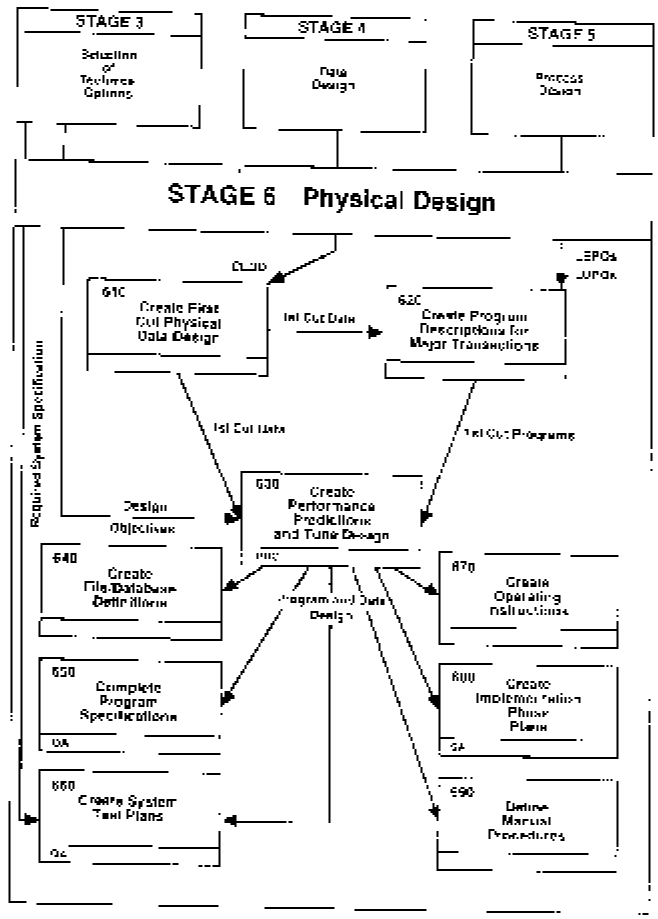


Figure 14 - Stage 5

STAGE 6- PHYSICAL DESIGN

This is the last of the design stages It requires as input the output from Stages 3,4 and 5. This provides all the documentation to allow programming and file creation to begin. It also lays down broad guidelines for the testing and implementation of the system.

Note: There are recommended steps for converting the logical data structures to suit some of the more common database packages.



Summary

The six stages of SSADM provide guidelines to the analyst to assure that all steps are taken. They do not make an experienced analyst out of a trainee. They will allow a trainee to take a more active part within the project guided by the experienced analyst.

The advantages of this approach are:-

- The user is involved at all stages
- The new design is not affected by the existing methods. The separation of the physical and logical view of the operation ensures this.
- Three views of the system are taken which lead to consistent details by cross checking means of cross-checking.

- Every stage is terminated by a Quality Review. This is staffed by people of mixed experience and not all are directly connected with the projects.