UNIT 4 BUILDING MANAGEMENT INFORMATION SYSTEMS

Structure

4.0 INTRODUCTION

This final unit outlines some of the steps involved in building MIS. A detailed study of the techniques would be taken up in a separate Block (System Analysis and Design). This unit is intended to provide a quick overview and to enable the reader to appreciate the role of System Analysis tool in the broader MIS context. Such a coverage will provide an overall perspective to the broad area of MIS, the subject matter of this block.

4.1 OBJECTIVES

After going through this unit, you should be able to:

- focus on the steps that form part of systems analysis that have special relevance to MIS
- appreciate the need for 're-engineering' an organisation rather than merely computerising existing manual procedures
- realise that development of management information systems is fundamentally different from programming
- focus on the tools and techniques used in actually building real world management information systems.

4.2 SYSTEM ANALYSIS

Building computer based information syslem needs a careful analysis and design before implementation. Detailed study to understand the information requirement forms the major component of the analysis. Providing a system that would meet the information requirement in an efficient manner is the subjects of infonnation system design. With the increased availability of new products and services ever-improving price performance ratio, such analysis and design become far more challenging. The newer products and services provides new ways of daring business as well, While the conventional analysis of information system requirement may indicate some possible alternatives, an alert analyst who is abreast of current and emerging developments in information technology may propose entirely new systems. Such exercises known as business "re-engineering" are the by products of good systems analysis and design, It should be clearly understood that good systems analysis is not merely an efficient way of getting computers to do what has been done manually for years. Such mere mechanisation fall short of the real objectives of good systems design, particularly these days when Information Technology is moving leaps and bounds.

The techniques of systems analysis are the ones that got covered in courses an **MIS.** While *the* techniques are definitely important it must be clearly borne in mind that systems analysis is fundamentally different from programming. In the opinion of the author much of the ills of the customers getting obnoxious university results and atrocious ulility bills, reported time and again in the Indian media over the past few years precisely arise out of programming efforts without going through the necessary systems analysis. As an analogy, systems analysis can be viewed as the job of an architect, while the job of programming of comparable to that of construction. Naturally systems analysis is for more conceptual in nature as contrasted to programming that is one Of details. It is no way to belitlle the intellectual process that can be demanded by programming profession. Programming is clearly analytical in nature but system analysis is far deeper conceptually. The input of system analysis naturally is far reaching compared to immediate impact of programming. Yet another way of distinguishing between these two different, yet related disciplines is that programming is skill oriented but systems analysis is more knowledge oriented. Systems analysis calling for very generous skills in limited to a handful of successful individuals; while programming skills being specific are easily acquired by fairly intelligence people. Naturally mass producing programmers is fairly easy compared to training a large number of analysts. In summary systems analysis is holistic in nature while programming is individualistic in nature.

Such detailed comparative sludy of programming and syslems analysis was necessary to reinforce the fact that building information system gives far beyond simple programming and coding. With a substantial emphasis laid an programming (thus street comer shops to train people in BASIC, FORTRAN, COBOL & C) in the Indian scene, this point can never be over emphasized. Teaching programming to develop on intellectual maturity is perfectly fine. This is similar Lo training in basic mathematical skills. But simply providing training in programming and expecting people to develop information systems in just not the right thing to do.

The steps in systcm analysis include the following eight steps generally known as systems life cycle.

1. Requirement Analysis
   (a) Determination
   (b) Specification
2. Feasibility Analysis
3. Basic system specifications
4. Hardware/Software Study
5. System Design
6. System Implementation
7. Evaluation and Acceptance
8. System Maintenance

(for a detailed study of the activities involved in each of the steps the reader is referred to the excellent text [Rajaraman p. 29].)

4.3 TECHNIQUES OF SYSTEMS ANALYSIS

Over the past decades, based on extensive empirical studies of the different steps of systems analysis enumerated in the last section, several techniques have emerged. We will briefly look at some of the fundamental techniques.

4.3.1 Requirement Analysis

Requirement Determination is generally done though extensive study of the system including the understanding of the goals, processes and constraints of the system for which information systems arc designed, Several forms are also designed and illustrated in the texts of system analysis. In the view of the author such techniques must be left to the ingenuity of the analyst; there is no straight forward algorithm to elicit the requirement from the user. It is an iterative process which the analysts use while interviewing several user/users groups. It will continue to remain an art rather than science.

For Requirement Specification both at the preliminary as well as the detailed stage, several diagramming techniques have evolved. In fact such diagrams have become the language of the analysts just as blue prints have become the language of the designer or balance sheet becoming the language of the accountant. We will detail below some of the diagramming techniques.

4.3.2 Diagramming Techniques

Data Flow & Document Flow Diagrams represent perhaps the most widely used diagramming technique of the systems analysis. The document flow diagram graphically represents the various documenls (typically paper developments in most cases) that flow across the system; the information carried by the paper document. must be generated and processed by the proposed information system. Often the documenls themselves may continue in paper form for administrative control, archives etc. as the necessary imaging techniques are still not cost effective, at least in India. Given the constraints of electrical power, paper may continue to be the major source of document, for many more years to come.

Data Flow Diagram (DFD) is a powerful diagram that can be used to document the information flow. It also presents itself to be broken down in lop-down fashion. At the top level, data flows are represented at very abstract aggregale level. Each component of the data flow is further broken down to different levels, so that at each level wc have just a few entities lo concentrate on. DFD have developed a representalion scheme to represent data store (storage & retrieval of data), processes (where some changes are made to the system) and entities (the player in the game) and the actual informalion flows.

The example data flow diagrams explains the concept. Breaking down the DFDs into different levels is generally known as Leveling and the DFDs are correspondingly termed Top level DFD and Leveled DFD.

A diagram similar lo DFD is the **context** diagram that is gcncrally a summary diagram that fully depicts the information flows and the entities involved. It is more useful lo the end user than the analyst. The detailed DFDs carry enough details that the analyst can hand them over to the programmer for detailed coding.

4.3.3 Data Dictionary

Another powerful tool that is extensively used is system analysis is the data dictionary. DDs as they are called provide a detailed reference lo every data item - the different names by which the item is represented, in different program modules, different data structures used to represent the item in different modules, the modules where the data item is generated, where it is stored and destroyed. In essence it provides a quick snapshot of every data item used by the information system. Needless to say it is extremely detailed and very useful for consistency checks, system modification and completeness checking.

A typical data dictionary appears as follow :

1 Page Data dictionary (*** ??? ***)

While these techniques are general in nature and used by the analyst in the different stages of the system life cycle the following are specific to some of the steps of the system life cycle.

4.3.4 Feasibility Report

A typical structure of the feasibility report will be as under:

Apreamble that sets the stage for the project, followed by goals statement that quantifies precisely the goals of the proposed information system. This is followed by a short narrative that describes in unambiguous yet jargon free language the present system. This must be understandable to any intelligent person not necessary a computer professional or even a computers literate, The proposed alternatives are then described once agdn in a reasonably jargon free language, Being a feasibility study the alternatives are unlikely to be detailed to the full extent, Until the full system is developed in its entirety, the full details are unlikely to be known, Yet we cannot go ahead with the final systein without doing a feasibly analysis. In a sense the details of the system to be built will unfold gradually from feasibility study stage to system specification stage to final implementation stage. These steps must be clearly understood by the user as well as the analyst, Based an 'sketchy' design of the proposed alternatives, an order of magnitude cost benefit study is prepared. It will be order of magnitude only as the costs and benefit are also unlikely to be in fully quantifiable form. Based on the detailed human/organisational/technological problems likely to be encountered in the project detailed in the feasibility study the end user decides a particular alternative that is worked out in detail for further implementation. The detailed design phase starts here.

4.3.5 Detailed Design

Roughly the detailed specifications are worked out followed by hardware/software plan. This constitute system design which once again need to be whetted by the user. Once this is done detailed system design starts. Effectively one can say that the analysis phase ends here and the design phase begins. Being a detailed design it may involve substantial effort on the part of technical system analysts, hardware, software, communication specialists etc. A major component of detailed system design is the database design covered in the next section. Actual coding is undertaken after the database design is complete.

4.3.6 Database Design

While DBMS permit efficient storage and manipulation of data files they do not cater to the structuring of the database themselves. After extensive uses of database in real world applications,several early analysts have felt the need for the right abstraction of data into the database so that any update/query operation captures the spirit of the meaning of the data stored in the data bases. One of the basic observation made by the early analysts was the possible "loss of information" by careless update operations on databases. This led to the concepts of normalisation pioncered by Codd []. Intuitively normalisation leads to the decomposition in such a way that no information is lost due to processing of data. Normalization ensures no loss of information and avoids insertion update and other anomalies. A table (relation) is said to be in First Normal Form (1NF) if there is an identifying key and there are no repeating groups of attributes. Intuitively First Normal Form ensures that all the table entries are atomic. A formal definition of a relation will be as follows:

A rclation is in (1NF) iff attributes of non-key fields are dcpcndent on a key.

For examplc, consider a student table with data as follows:

{Student_Namc, {Coursc_No., Course–Name),
(Course–No., Coursc_Namc},.......}

with repeating groups. Obviously such a varying field leads to blank fields (that do not depend on the key). A simple way to transform a rclation that is not in (1NF) is to replace it with a simple relation with as many records as the non-full repeating groups. For example,

Student_Name, Course–No, Course–Name
Student_Name, Course–No, Course_Name

for as many courses as the student has. A further refinement of the First Normal Form (INF) is the Second Normal Form (2NF) where we avoid any dependencies among the relationships. For example in the earlier example course name depends on Course–No which is not a key field. Such a structure may lead to inconstancy where there could be different Course–Name combinations for the same Course–No. To ensure that the relations (tables) are free of such inconsistency we break down the rclation further into

{Student_Name, Course–No.) (Course–No, Course_Title}
(Student_Name, Course–No.}

A still further refinement is Third Normal Form (3NF) whcrc we further specify that there is an identifying key, no repeating of attributes, no attribute exist that does not require the whole of the key and there is no transitive dependency (i.e., an attribute depending an another non-key attribute) i.e., in general attributes are mutually independent.

The benefits of normalisation arc the avoidance of update anomalies explained through the following examplc:

Example (**** ???? ****)

Note that the 1NF, 2NF, & 3NF are upwards inclusive, that is 2NF relation is necessarily in 1NF and a relation that is in 3NF is necessarily in 2NF. They represent progressively increasing refinement of data relationships represent.

Database **theory** details further degrees of normalisation including 4NF and 5NF. While theoretically sound, such further refinements add(?) like, if any, to data modeling real world data. Since our text is primarily on Information System and not on Database theory we will not further elaborate an advanced normalisation. It may be mentioned in passing that a slightly modified 3NF known as Boyce-Codd Normal Form (BCNF) lends itself to automatic decomposition from an un-normalised form using Bernstein algorithm [].

Physical design of database includes the storage devices and their management. This calls for detailed knowledge of the specific hardware and software on which the information system is likely to be implemented. Hardware/Software Selection is indeed a difficult job considering the fact that many competing products are being introduced in the market with improved benefit cost ratio. An alert analyst has to keep himself posted continuously about the trends in technology, obsoleteness of technology, viability of emerging technology, future trends and even the financial and market viability of the hardware/software vendor in this fiercely competitive area of Information Technology. Access to specific experts in areas, use of bench marketing literature and user group and peer opinion are some of the resources used by most analysts in this difficult task.

4.3.7 System Implementation

System implementation includes the detailed design of the process, their validation and thorough checking. While the formal methods of proving program correctness [] are evolving, they are still not useful to test out large commercial software to help information system planning. Many of the analysts use experimental version using what is known as "parallel runs". Here **both** the current system and proposed new system **are run in parallel** for a specified time **period** and the current system is used to validate the **purposed** system. (This is like the practice of the incumbent senior officer spending some time with the outgoing office before actually taking over the full responsibility.)

4.4 SUMMARY

This module provides a quick introduction to the techniques of system analysis and design **that** are to be used in actually building real world management information system. Where the full details will be shown in details in a separate course, this unit is intend to provide a quick overview of all the issues involved in MIS development. Accordingly many of the tools System Analysis are only touched upon.

4.5 SELF-ASSESSMENT EXERCISES

1) Outline the steps of system analysis.
2) Understand the difference between programming and system analysis.
3) What is a Data Flow Diagram (DFD)?
4) Briefly describe the structure of a Feasibility Report?
5) Explain the role of Normalisation in relational database design.