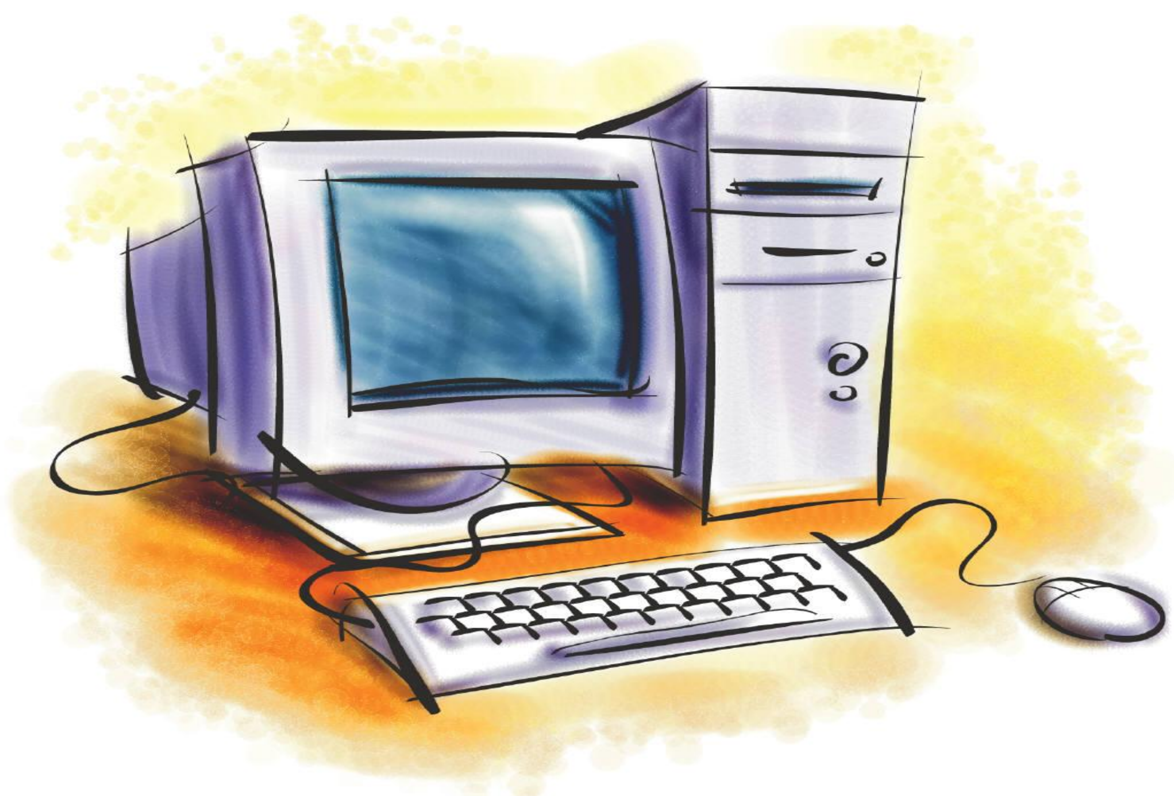
UNESCO-NIGERIA TECHNICAL & VOCATIONAL EDUCATION REVITALISATION PROJECT-PHASE II

NATIONAL DIPLOMA IN COMPUTER TECHNOLOGY



Introduction to System Analysis & Design

COURSE CODE: COM 125

TABLE OF CONTENTS

WEEK 1

OBJECTIVES:

The student is expected to learn the following:

- **Definitions of system,**
- **common types of system and**
- **General system principles**

DEFINITION OF SYSTEM

There are many common types of systems that we come into contact with every day. It is important to be familiar with different kinds of systems for at least two reasons:

- First, even though your work as a systems analyst will probably focus on one kind of system - an automated, computerized information system - it will generally be a part of a larger system. For .example, you may be working on a payroll system, which is part of a larger "human resources" system, which is, in turn, part of an overall business organization (which is itself, a system), which is, in turn, part of al larger economic system, and so on. Thus, to make your system successful, you must understand the other systems with which it will interact. Many of the computer systems that we build are replacements, or new implementations of, non-computerized systems that arc already in existence. Also, most computer systems interact with, or interface with, a variety of existing systems (some of which may be computerized and some which may not). If our new computer system is to be successful, we must understand, in reasonable detail, how the current system behaves.
- Second, even though many types of systems appear to be quite different, they turn out to have many similarities. There are common principles and philosophies and theories that apply remarkably well to virtually all kinds of systems. Thus, we can often apply to systems that we build in the computer field, what we have learned about other systems, based on our own day-to-day experience, as well as the experience of scientists and engineers in a variety of fields.

Thus, if we understand something of general systems theory, it can help us better understand computerized (automated) information systems. Today, this is more and more important, because we want to build stable, reliable systems that will function well in our complex society, and of course, there are many examples of non-computer systems that have survived f$_u$r thousands of years.

And now, we can consider a definition of the basic term "system". It provides several definitions:

1. A regularly interacting or interdependent group of items forming a unified whole.
2. An organized set of doctrines, ideas, or principles, usually intended to explain the arrangements or working of a systematic whole.
3. An organized or established procedure.
4. Harmonious arrangement or pattern: order.
5. An organized society or social situation regarded as stultifying establishment.

Common types of systems

There are many different types of systems, but indeed, virtually everything that we come into contact with during our day-to-day life is either a system or a component of a system (both).

It is useful to organize the many different kinds of systems into useful categories. Because our ultimate focus is on computer systems, we will divide all systems into two categories: natural systems and man-made systems.

Natural systems

There are a lot of systems that are not made by people: they exist in nature and, by and large, serve their own purpose. It is convenient to divide natural systems into two basic subcategories: physical systems and living systems.

Physical systems include such diverse example as:

- Stellar systems: galaxies, solar systems, and so on.
- Geological systems: rivers, mountain ranges, and so on.
- Molecular systems: complex organizations of atoms.

Physical systems are interesting to study because we sometimes want to modify them. We also develop a variety of man-made systems, including computer systems, which must interact harmoniously with physical systems; so it is often important to be able to model those systems to ensure that we understand them as fully as possible.

Living systems encompass all of the myriad animals and plants around us, as well as our own human race. The properties and characteristics of familiar living systems can be used to help illustrate and better understand man-made systems.

The living systems, whether at the level of the cell, the organ, the organism, the group, the organization, the society, or the supranational system, all contain the 19 subsystems:

1. The reproducer, which is capable of giving rise to other systems similar to the one it is in.
2. The boundary, which holds together the components that make up the system, protects them from environmental stresses, and excludes or permits entry to various sorts of matter-energy and information.
3. The ingestor, which brings matter-energy across the system boundary from its environment.
4. The distributor, which carries inputs from outside the system or outputs from its subsystems around the system to each component.
5. The converter, which changes certain inputs to the system into forms more useful for the special processes of that particular system.
6. The producer, which forms stable associations that endure for significant periods among matter-energy inputs to the system or outputs from its converter, the materials synthesized being or growth, damage repair, or replacement of components of the system, or for providing energy for moving or constituting the system's outputs of products or information markets to its suprasystem.
7. The matter-energy storage subsystem, which retains in the system, for different periods of time, deposits of various sorts of matter-energy.

8. The extruder, which transmits matter-energy out of the system in the form of products or wastes.
9. The motor, which moves the system or parts of it in relation to part or all of its environment or moves components of its environment in relation to each other.
10. The supporter, which maintains the proper spatial relationships among components of the systems, so that they can interact without weighing each other down or crowding each other.
11. The input transducer, which brings markers bearing information into system, changing them to other matter-energy forms suitable for transmission within it.
12. The internal transducer, which receives, from other subsystems or components within the system, markers bearing information about significant alterations in those subsystems or components, changing them to other matter-energy form of a sort that can be transmitted within it.
13. The channel and net, which are composed of a single route in physical space, or multiple interconnected routes, by which markers bearing information are transmitted to all parts of the system.
14. The decoder, who alters the code of information input to it through the input transducer or internal transducer into a private code that can be used internally by the system.
15. The associator, which carries out the first stage of the learning process, forming enduring associations among items of information in the system.
16. The memory, which carries out the second stage of the learning process, storing various sorts of information in the system for different periods of time.
17. The decider, which receives information inputs from all other subsystems and transmits to them information outputs that control the entire system.
18. The encoder, who alters the code of information input to it from other information processing subsystems, from a private code used internally by

the system into a public code that can be interpreted by other systems in its environment.

19. The output transducer, which puts out markers bearing information from the system, changing markers within the system into other matter-energy forms that can be transmitted over channels in the system's environment.

Keep in mind that many man-made systems (and automated systems) interact with living systems. In some cases, automated systems are being designed to replace living systems. And in other cases, researchers are considering living systems as components of automated systems.

Man-made systems

Man-made systems include such things as:

1. Social systems: organizations of laws, doctrines, customs, and so on.
2 An organized, disciplined collection of ideas.
3. Transportation systems: networks of highways, canals, airlines and so on.
4. Communication systems: telephone, telex, and so on.
5. Manufacturing systems: factories, assembly lines, and so on.
6. Financial systems: accounting, inventory, general ledger and so on.

Most of these systems include computers today. As a systems analyst, you will naturally assume that every system that you come in contact with should be computerized. And the customer or user, with whom you interact will generally assume that you have such a bias. A systems analyst will analyze, or study, the system to determine its essence: and understand the system's required behaviour, independent of the technology used to implement the system. In most case, we will be in a position to determine whether it makes sense to use a computer to carry out the functions of the system only after modeling its essential behaviour.

Some information processing systems may not be automated because of these common reasons: Cost; Convenience; Security; Maintainability; Politics.

Automated systems

Automated systems are the man-made systems that interact with or are controlled by one or more computers. We can distinguish many different kinds of automated systems, but they all tend to have common components:

1. Computer hardware (CPUs, disks, terminals, and so on).
2. Computer software: system programs such as operating systems, database systems, and so on.
3. People: those who operate the system, those who provide its inputs and consume its outputs, and those who provide manual processing activities in a system.
4. Data: the information that the system remembers over a period of time.
5. Procedures: formal policies and instructions for operating the system.

One way of categorizing automated systems is by application. However, this turns out not to be terribly useful, for the techniques that we will discuss for analyzing, modeling, designing, and implementing automated systems are generally the same regardless of the application. A more useful categorization of automated systems is as follows:

1. Batch system: A batch system is one which in it, the information is usually retrieved on a sequential basis, which means that the computer system read through all the records in its database, processing and updating those records for which there is some activity.
2. On-line systems: An on-line systems is one which accepts input directly from the area where it is created. It is also a system in which the outputs, or results of computation, are returned directly to where they are required.
3. Real-time systems: A real-time system may be defined as one which controls an environment by receiving data, processing them, and returning the results sufficiently quickly to affect the environment at that time.
4. Decision-support systems: These computer systems do not make decisions on their own, but instead help managers and other professional "knowledge workers" in an organization make intelligent, informed decisions about various aspects of the operation. Typically, the decision-support systems are

passive in the sense that they do not operate on a regular basis: instead, they are used on an ad hoc basis, whenever needed.

5. Knowledge-based systems: The goal of computer scientists working in the field of artificial intelligence is to produce programs that imitate human performance in a wide variety of "intelligent" tasks. For some expert systems, that goal is close to being attained. For others, although we do not yet know how to construct programs that perform well on their own, we can begin to build programs that significantly assist people in their performance of a task.

General systems principles

There are a few general principles that are of particular interest to people building automated information systems. They include the following:

1. The more specialized a system is, the less able it is to adapt to different circumstances.
2. The more general-purpose a system is, the less optimized it is for any particular situation. But the more the system is optimized for a particular situation, the less adaptable it will be to new circumstances.
3. The larger a system is, the more of its resources that must be devoted to its everyday maintenance.
4. Systems are always part of larger systems, and they can always be partitioned into smaller systems.
5. Systems grow. This principle could not be true for all systems, but many of the systems with which we are familiar do grow, because we often fail to take it into account when we begin developing the system.

WEEK 2

OBJECTIVES:

The students are expected to learn the following:

- **The stages of system analysis and design,**
- **System approach,**
- **System requirement,**
- **User's requirement,**
- **Technical requirement and**
- **System survey.**

System development Life-cycle

The six phases in the system development life cycle can be identified by different names. Also, there are no definite rules regarding what must be included in each of the six phases.

Initial investigation

This phase introduces the objectives of the initial investigation, the steps required to initiate an investigation; the tasks involved in the initial investigation, and the data gathering and interviewing techniques. It also includes information and exhibits that should be in the initial investigation report, with regard to "How the standards manual might be used?" and "why to do this after reading this section."

Feasibility Study

This phase determines the objectives of the feasibility study and who this task belongs to – analysts or the project team? It lists out the steps required to complete a feasibility study, identifies the scope of the current system, problems and unexploited opportunities in the current system, which may be either manual or automated. It then discusses the major objectives for the new system, and the various methods to gather data and determine how to use the methods. It also helps to estimate the costs of each possible solution, and develops estimates of the benefits and shortcomings of each solution. It presents users and the management views on the above issues and their decision of whether to commit to the analysis part of the project. This phase may be included some related subphases:

- Current physical model: The description of the system as it is now, including the mechanisms used to accomplish tasks (e.g., people, devices).
- Current logical model: The system description in term of functions, processes, and data with the mechanisms removed.
- New Logical Model: The Current Logical Mode with new features added.
- New Physical Mode: The Current Logical Model with the various processes allocated to automation, manual procedures, other mechanisms.

General Design

Show the users the view of the application. In this phase, as well as the previous two, users must be directly involved. In this phase, the analysts' imagination, creativity... and initiative are used to their fullest. It also issues the System Flowcharts to develop. In addition, broad specifications that describe how the data is to be processed by the computer will be developed.

Detailed Design

The Analysts have to transform from general design specifications to detailed requirements that can be used in implementing the many tasks that make up the system. The final report should include the Procedural Flowchart, record and report layout, and a realistic plan for implementing the design. No major changes should be made until the design is implemented and the system is operational, Then the programs for both transaction processing and batch jobs are executed. If there is the problems, the professional data processing staff is responsible for determining the cause and implementing a solution.

Implementation

Detailed logic plans must be developed for all programs before they can be written, tested, and documented. After each procedure and program are tested, the system is tested. The conversion to the new system is made according to a plan developed in the detailed design phase. Many companies encourage their customers documenting their own systems.

System Audit

When the implementation report is submitted, an evaluation should be made to determine whether the system meets the objectives stated in the general design report. In this phase, users may be able to suggest the easy-to-implement improvements.

As in the six phase development life cycle, the project can be dropped at any point prior to implementation. A project may be dropped if the benefits derived from the proposed system do not justify commitment of the needed resources. Or if the costs is higher, than expected.

System approach

Approaches to systems development, in professional organizations, usually follow one of two basic models: the waterfall model or the spiral model.

The Waterfall model is the basis of most of the structured development methods that came into use from the 1970s onwards. It provides a framework for planning top - down systems development. The development flows down a number of successive stages are typically:

- Systems analysis;
- Systems design;
- Systems build and test;
- Systems introduction and transition;
- Maintenance of production status systems.

The Spiral model of systems development proposed by Barry Boehm, has become popular is basis for iterative systems development. The spiral model follows the same breakdown into stages, but takes an incremental approach to systems development. Typically, a system is divided into smaller sub-sets for development and delivery. This provides functionality to end users at regular intervals, rather than at the end of a waterfall development. It is also common in iterative development for highly-skilled developers to model systems with stakeholders, the design and implement them, rather than assign the stages to different departments or groups. Effective use of the spiral approach to system development can deliver systems quickly and ensure user involvement, especially when prototypes are part of the development approach.

System requirements

The system is developed base on the requirements of the system itself (to help manage an organization) and technical requirements.

There are different view points of what information system automatization is like, however, we can classify them into 3 main groups: view point of the system that will be developed, information expert's view point and user's one. These points of view often conflict with one another, at the same time we are required to build up a successful system in which the system, information experts and end users share the same view point. Information system is a system that collects information, manages them and creates information products for its users. The following part will discuss requirements of system, information experts and users.

The system's requirements

- Suitable with the general strategies: Small changes in the organization's development may result in bigger impacts on the information system's requirements. Therefore, during the system development process, these requirements should be regularly checked for its suitability with the general strategies.
- Supporting decision maker: Together with hands-on experience, knowledge and anticipating ability, information system plays an important role in supporting decision making process;
- Competition edge: The more competitive the environment, the more demand for better systems;

- Return on Investment: A new information system needs to show financial benefits it can bring, because decisions on investment, development costs and operation costs are based on financial analysis; More advanced evaluation techniques can be applied. These techniques take into consideration issues such as customer support, organisational effectiveness, risk, etc.
- Overhead control: human resource change will influence staff number, staff skills and workload. In many cases, while human resource structure is unchanged, workload and requirement for staff skills are yet much higher;
- Supporting operational management: This is clear in preparing detail information, making reports quickly, which contribute to a more flexible and efficient way of management;
- Improving information communication: Optimizing the flow of information, preparing necessary updated information and providing users with the information;
- Impact of information products: Information products are final outcomes of the IT system. We need to pay special attention to requirements for information products so as to thorough analysis. These requirements shall be frequently in comparison with general strategies while developing the system;
- Ability to implement more quickly and better.

Users requirements

Users are those who use the information system to manage their organizations rather than simply those who work with computers. They are the ones who master the current information system (from information sources, management requirements to the system's shortcomings) and are future owners of the system. Thus their requirements should be respected while developing any information system. Attention should be paid in the following issues:

- Easy access: The system must be able to timely access data and manipulation supports.

- The system: The system must be solid and stable, being able to meet staffs requirements and provide accurate information, easy to maintain and restructure, quick in identifying and correcting mistakes;
- Interface: Suitable with working style of users, stable, easy to control data, independent and flexible, enabling users to approach in different ways.

Technical requirements

Technological requirements should be taken into account when designing information systems. The important points are as follows:

- Information volume: Information technology equipment must be suitable with the volume of information that is to be processed;
- Period: Everyday information which arises regularly is repetitive information that requires special care;
- Accuracy: Specially high accuracy is required now and then. Accuracy is important but difficult to meet;
- Complexity: Issues in information treatment can be processed in principle. However, due to its complexity, the current system fails to resolved the issues that need to be resolved by the new system.

Besides the issues of the three view point groups, we would also like to remind you of the following popular issues.

- Incompatibility: Applications developed in different environments are often incompatible. Computers of different kinds are difficult to be connected together, making offices isolated from information processing system;
- Shortcomings: Lack of typical information, unfriendly interface with users, bad storage of information.
- Low reliability: Data is conflicted, inadequate and unamendable, information is not updated regularly;
- Poor resources: Inadequate ability to search for information, lack of exploitation tools for users, low information quality;

- Bad support: Users are not aware of what they have handy, there are no clear development strategies, development pace is slow, support is inadequate, mutual understanding is low.

It is clear that the IT experts and the users view the system from different perspectives thus having different requirements. Analyzer's capability is shown in his/her ability to collect ideas and evaluate them from a wider perspective, because system developers are only knowledgeable of their own areas.

System survey

The survey process is often divided into 2 main phases:

- Preliminary survey: define the project's feasibility;
- Detailed survey: define what needs to be done and accomplishments that should be achieved

Preliminary survey: in this phase, important questions to ask are:

- Do we need to carry it out?
- What do we need to continue to do?
- How long do we need to do it?
- What is the estimated price?
- What are the benefits and difficulties?

To get the answers for those questions, follow the steps listed below:

- Define what needs to be done to get the same requirements from the organization, users and the information system;
- Define the scale of the problem that needs to be solved in considering particular issues of the organization;
- Define the users whose work will be changed following the development of the system. Users are often divided into 4 categories: users at

manipulating level, at supervision level, at management level and professional level;

- Make up a preliminary survey report base on discoveries from initial observation to get an overview from different perspective, making up a solid basis for the next phase.

WEEK 3

OBJECTIVES:

The students are expected to learn the following:

- **Feasibility studies,**
- **Cost of the existing system,**
- **Cost of operation of the proposed system,**
- **Cost of development of the proposed system,**
- **Benefits of proposed system and**
- **Presenting the assessment of feasibility study.**

FEASIBILITY STUDY

The main objective of a feasibility study is to test the technical, social and economic feasibility of developing a computer system. This is done by investigating the existing system in the area under investigation and generating ideas about a new system. The proposed system(s) must be evaluated from a technical viewpoint first, and if technically feasible their impact on the organisation and staff must be assessed. If compatible social and technical systems can be devised, then they must be tested for economic feasibility.

Assessing technical feasibility

The assessment of technical feasibility must be based on an outline design of system requirements in terms of inputs, outputs, files, programs, procedures and staff. This can then be quantified in terms of volumes of data, trends, frequency of updating, cycles of activity, etc, in order to give an indication of the scale of the technical system. Methods used for investigation, analysis and design of this system are described in the chapters which follow. Having identified an outline system, the investigator must go on to suggest the type of equipment required, methods of developing the system, and methods of running the system once it has been designed.

With regard to the processing facilities, the feasibility study will need to consider the possibility of using a bureau or, if in-house equipment is available, the nature of the hardware to be used for data collection, storage, output and processing. As the technology develops, the range of choice is widening and the complications of butch processing, on line processing, distributed processing, microprocessors, etc, all have to be taken into account.

On the system development side, the feasibility study must consider the various ways pf acquiring the system. These include me purchase of a package, the use of a consultancy organisation or software house to design the system and write the programs, a co-operative development with another similar organisation, or in-house development of the system (or a combination of these). Most of these technical possibilities will not be

feasible in a given environment and so the eventual proposal will reflect the constraints of the data processing situation in the organization. It should also reflect the social aspects of the system.

Assessing social feasibility

The assessment of social feasibility will be done alongside technical feasibility. Each of the alternative technical solutions which emerge must be evaluated for its social implications. The needs of various people affected by the proposed system (both directly and indirectly) must be taken into account. Impact on organisation structure, authority, salary levels, group relationships, and jobs should be considered-not just in negative terms (i.e. bad effects) but positively (i.e. what can be improved? what contribution can be made?).

The various social costs must also be evaluated; the-e will include the costs of education and training, communication, consultation, salary changes, job improvements, redundancy payments, and hidden costs like those caused by hostility, ignorance and fear. But primarily the social evaluation should rank the possible technical solutions in terms of the extent to which they improve the jobs and the working environment of those affected.

Assessing economic feasibility

Justification for any capital outlay is that it will increase profit, reduce expenditure or improve the quality of a service or goods which in turn may he expected to provide increased profits. Proposed or developing systems must be justified by cost and benefit criteria to entire that effort is concentrated on projects which will give the best return at the earliest opportunity.

The determination of development and subsequent operational costs, and of the savings compared with existing systems, can be difficult, especially at the initial study stage, but the quantification of benefits, which management are coming increasingly to expect is often much more ddifficult for example, improved customer relationships arising from the provision of a more informative sales and product analysis for the sales manager, quicker response to customers' enquiries and orders, or faster despatches, invoices and statements.

The technique of cost benefit analysis is often used as a basis for assessing economic feasibility. This type of analysis is not new; it is carried out as a matter of course for many other capital development projects. The factors for evaluation are:

- cost of operation of the existing and proposed system;
- cost of development of the proposed system ;
- value of the benefits of the proposed system.

Cost of operation of the existing system

These costs will normally be calculated from cost records. Items to be identified and investigated include:

- manpower;
- materials;
- equipment;
- overhead expenses;
- intangible costs.

Some of these factors are more easily obtainable than others, but they can all be determined within reasonable limits either by interviews with management or by scrutiny of records. Departmental budgets and accounting records are the most fruitful source to reveal the operating costs of the existing system. Costs ascertained by interviews should be confirmed by analysis of the appropriate records.

Manpower costs can be extracted from budgets or payrolls, and it is usual to add an amount relating to the additional cost to the company of each employee such as insurance and pensions.

Materials costs include consumables such as stationery, but stocks and work in progress may need to be considered.

The operating cost of equipment may be expressed as a unit rate if it is established as a cost centre; in this case, it will need to be extended to a period of time, such as annual cost ; otherwise depreciation costs, initial or replacement costs of the equipment will need to be estimated.

Overhead expenses are indirect expenses incurred by the company on behalf of all department, such as rent, rate, power, lighting and may be extracted from records of cost centres or departmental centres to which they are allocated by the company's accountant.

Intangible costs of the existing system include such things as lost sales, as a result of inappropriate stock levels, or loss of interest because of poor credit control. Such costs must be estimated and included in present system costs. It must be borne in mind also, that the future costs of the existing system may well change. Estimates of growth must be obtained from user departments, and these may affect the viability of the project. For example, the operating costs of the existing system may be increasing rapidly at the lime of the evaluation, and if projected to the point in the future at which the proposed system would operate, these costs would be considerably higher.

Costs of operation of the proposed system

The proposed system is likely to include costs in all the areas mentioned above. There are, however, certain other specific cost areas which must be considered in a computer-based system:

- data preparation operators, consumable materials, equipment and maintenance;

- computer running costs, operators, consumable materials, equipment, and maintenance;
- data control staff, equipment and storage;
- system maintenance staff.

Each computer installation will have to have its own method of apportioning these costs. These may all be allocated, including overheads, to the computer as a cost centre and charged out as a rate per hour to company overheads or direct to the user departments concerned. Computer run limes for a system, including program, maintenance and modification costs, need careful consideration; they have been known to escalate considerably after implementation. Systems analysts must be guided by past experience at their installations, preferably by recorded performance.

Cost of development of the proposed system

The costs of development will be based on the time estimates for the overall project plan. The accuracy and detail of such estimates will depend on the point in development at which justification of the system is required. The estimates at the initial study stage will of necessity be less detailed and accurate than at a later stage of the project. During the later stages, some costs w ill have become factual, and estimates for the remaining parts of the project should be that much more accurate, being nearer in time and based on later information.

The stages of development of a system to be costed will be those described in chapter 2. Because experience has shown hat despite careful planning, deviations occur and unforeseen circumstances arise, some installations include a contingency allowance in the costs to cover this short-coming. This practice is not encouraged by all d managers because its misuse can negate detailed planning; it should therefore be used carefully and preferably be based on recorded past experience. Particular attention should be devoted to the costs of user education and participation, and project management. Project development cost is a once-only cost, though in some installations it is extended to include any subsequent maintenance and modifications during the operational life of the system.

Benefits of proposed system

The benefits of any proposed system can be considered as falling into two categories:

- tangible benefits;
- intangible benefits (which may be difficult to quantify).

Tangible benefits are those which are readily evaluated in money; they arise out of a direct comparison between the costs of operating the existing and proposed system as net cost reductions or savings. These cost reductions are of the type discussed under 'the costs of operating the existing system' (i.e. ma

npower, materials, equipment and overheads). However, direct comparison may show only minimal savings or even increases in costs. A reduction in personnel cost alone is not usually sufficient to justify capital expenditure, unless the reduction is very substantial.

At any one time, a company will have a list of many innovations or projects, against which a proposed system will have to compete. It will be further constrained by the limitations of the company. These include:

- financial limits of cash and credit;
- capacity of management to undertake the heavy load of innovations put forward;
- customers and employees who may be affected by the innovations.

Management are now looking more selectively at the extent to which projects contribute to achieving the corporate objectives of the company within the limitations set out above.

A significant benefit of a computer-based system may be that it can enable the company's financial resources to be used more effectively; for example, in an inventory control system, the computer enables techniques to be used which can provide very large savings in capital, previously locked up in stock holdings, by reducing the quantities previously held, and at the same time offering an improved service.

Another system may lead to improved profit margins by identifying less profitable products; management may then either modernise these production lines or concentrate available resources on the more profitable products to reduce the unit cost.

Intangible benefit are more difficult to estimate a id justify, usually requiring the skill of the particular management concerned; in the example used above, given adequate information, production management and the accountant would be able to estimate the effect on profit margins of reducing the unit cost of production. This, in turn, could improve the ability of the company to compete in its markets, but the estimated value of this benefit would require the judgement of the marketing and sales manager. Generally speaking, the computer can be used for lots of activities whose benefits are intangible. For example, greater accuracy, resulting from elimination of boring tasks and more extensive checking, is difficult to evaluate. The computer can make time available to managers for more useful activities instead of record keeping or control, but it is difficult to assess the financial benefit or the likely utilisation of this extra time.

The determination of the value of improved information, or of information in general, is the most difficult task of all. At an academic level it may be said that information is only of value if the possession of information causes the person in possession to carry out some activity which he would not otherwise carry out. The value then is the difference in benefit between the action without the information and the action with it. It is true to say, however, that the manager concerned is in the best position to evaluate this type of benefit.

Presenting the assessment of feasibility

The proposed system must be presented to management with some indication of anticipated performance. The factors which should be included are:

- cost: operating, maintenance, unit, once-of;
- time: response, access, elapsed, resource, cycle, pro jess, management;
- accuracy: frequency, significance and correction of errors;

- reliability: stability, durability;
- security: legal, safety, secrecy, confidentiality;
- flexibility: variability and sensitivity;
- capacity: average, low and peak loads;
- efficiency: performance ratios;
- acceptance: customer, employee, management, shareholders.

Not all of these factors will be important in every system, nor every measurement factor specified with equal precision. For convenience, these may be shown graphically on a measurement scale indicating comparisons between:

- existing;
- acceptable;
- desirable;
- proposed.

On this scale, the systems analyst will be able to record the performance of these factors, as say, average and best. This highlights those factors which need reviewing or justifying in the first place. In establishing what is acceptable and desirable, the systems analyst will try to produce that performance which will gain management acceptance. For this, the systems analyst must know what are the business needs, the views and opinions of individual managers and the capabilities of the data processing facilities available or specified.

WEEK 4

OBJECTIVES:

The students are expected to learn the following:

- **Feasibility study report,**
- **Concept of data flow diagram,**
- **Component of data flow diagram and**
- **Analysis specification**

FEASIBILITY STUDY (REPORT)

Theoretically, feasibility report* should be written in user's language (not necessarily non-technical language). Technical parts for designing should be put into an appendix.

All reports must have a cover page with name of project, its author, address, contact numbers. Cover page is followed by content pages with the following main items:

- Objectives of the system;
- Inter-connection between related departments;
- Details of the current system;
- Future system and sketchy estimate of costs and benefits;
- Advice;
- Time frame and plan for system development;
- General description (non-technical);
- Original document;
- Evaluation of the current system in terms of: organizational structure, technology, information system, users' IT skills, policy renovation...

The following points should be included in the conclusion part of the report:

- Is the document of the current system adequate?
- Have users reviewed and agreed with your point of view?
- Have users been consulted and has analyzer addressed their concern properly?
- Has the whole report been researched thoroughly?

- What functional requirements need further research?
- Have all requirements been reviewed?
- What are the substitute designing solutions?
- What are the project's possible changes?

DATA FLOW DIAGRAM: DESCRIBE THE INFORMATION FLOW IN THE SYSTEM

The next step of system analysis is to consider in detail the information necessary for the implementation of functions discussed above and the one necessary for the improvement of the functions. Modeling tool frequently used for this purpose is data flow diagrams. Data flow diagram will support 4 main activities:

- Analysis: DFD is used to determine requirements of users
- Design: DFD is used to map out a plan and illustrate solutions to analysts and users while designing a new system.
- Communication: One of the strength of DFD is its simplicity and ease to understand to analysts and users;
- Documents: DFD is used to provide special description of requirements and system design. DFD provide an components. We have to use other tools like database dictionary, process specification to get an idea of which information will be exchange and how.

THE MAIN COMPONENTS OF DATA FLOW DIAGRAM ARE:

- The process: The process shows a part of the system that transforms inputs into outputs; that is, it shows how one or more inputs are changed into outputs. Generally, the process is represented graphically as a circle or rectangle with rounded edges. The process name will describe what the process does.
- The flow: The flow is used to describe the movement of information from one part of the system to another. Thus, the flow represents data in motion, whereas the stores represent data at rest. A flow is represented graphically by an arrow into or out of a process.

- The store: the store is used to model a collection of data packets at rest. A store is represented graphically by two parallel lines. The name of a store identified the store is the plural of the name of the packets that are carried by flows into and out of the store
- External factors: External factors can be a person, a group of persons or an organization that are not under the studying field of the system (they can stay in or out of the organization), but has certain contact with the system. The presence of these factors on the diagram shows the limit of the system and identifies the system relationship to the outside world. External factors are important components crucial to the survival of every system, because they are sources of information for the systems and are where system products are transferred to. An external factor tends to be represented by an rectangle, one shorter edge of which isj omitted while the other is drawn by a duplicated line.
- Internal factors: While the external factors' names are always nouns showing a department or organization, internal factors' names are expressed by verbs or modifiers. Internal factors are syster functions or process. To distinguish itself from external factors, an internal factor is represented by/ rectangle, one shorter edge of which is omitted while the other is drawn by a single line.

ANALYSIS SPECIFICATION

You can construct DFD model of system with the following guidelines:

- Choose meaningful names for processes, flows, stores, and terminators
- Number of processes
- Re-draw the DFD many times
- Avoid overly complex DFD
- Make sure the DFD is consistent internally and with any associated DFD

WEEK 5

OBJECTIVES:

The students are expected to learn the following:

- **Fact finding techniques, and**
- **Requirements of a system**

INVESTIGATION METHODS (Fact Finding)

Most of the difficulties one can meet in system analysis result from the survey process. Some people perceive incorrectly that the survey process is finished after the questions on the current system and the future system have been answered and the answers have been analyzed. In fact, all information reflecting the current situation have to be collected, and it requires great effort so as to decide what information to collect and how to collect it. In this section, we'll discuss several popular survey methods.

SURVEY METHODS

The contexts in which you make interviews are often different and unpredictable. However, interviews are the main source of information about the future system and the current system. There are 2 main reasons for interview failures:

(i) Interviewer fails to understand what users say,

(ii) Bad communications between interviewer and interviewee. The followings are tips for interviewer

The interview: Before the interview, you should contact the interviewee directly (or through his/her secretary) to set up an appointment and agree with him/her on time, venue and interview objectives. During the interview, if you behave professionally, you will receive the same attitude from your interviewee. Try to attentively listen to the people you're talking to and take notes of all the necessary information you are provided during the interview. The interview paper and minute are always useful for you and your successor because it helps you master the origin of the information you have. It's important to open and close the interview carefully because this may impact the way your questions are answered. When opening the interview, try to do it in a trustful, respective way with your good will. When closing the interview, you should recap the main points of the interview, make arrangement for the following cooperation and leave the issues open for discussion between both sides. Don't make the conversation too lengthy nor prepare too many questions to ask. The best way is you prepare a short list of the main topics you want ask about.

Questions: You should carefully choose the kind of question you will ask. As each kind of question serve a particular context, you will suffer if you choose it wrongly. Open questions always create more opportunities for the interviewee to answer, but don't expect too much, not all the answers are good enough. You should take those questions which show your prediction such as "I feel that...", "I sense that..." and use the words that can help emphasize your idea. Learn to know how to be quiet and listen to your interviewers. Don't present a question while (s)he thinks or manage to answer a question, however sometimes it's useful to give him/her a hint. Don't try to ask in a way so as to lead the interviewee to your own direction, you wouldn't get much information that way.

Communications: The language you use and the way you talk about technical stuff should in some way create a close contact between you and the one you're interviewing.

OBSERVATION METHODS

Official observation: It's not a good method to observe every single elements while collecting information to develop the system. The future system you're building up may be deemed to change the current way of working. Moreover, those you're looking at, may feel uncomfortable and may behave unusually, which will affect your survey's quality.

Unofficial observation: In order to get an overview of an organization, take a look at its pile of paper and document, interruption of work, unreasonable timing and positive reflection of a good working environment... It's also important to know the quantity and quality of data that need to be processed and predict how they change over the time. Researching through document is the final good method to get important information.

QUESTIONNAIRE METHOD

This method requires your clear instructions to the user. A questionnaire can be designed base on the following points:

- Title: describe the objectives and main contents;
- Data classification: categories of data that will be used;
- Data: contents of the data in each category.

In general, this method is complicated and ineffective for inexperienced analyzers.

It's clear that each method has its own strong and weak points and is suitable for a particular context. However, regardless what method you use, the general principle is: The more information you get about the operation environment of an organization, the more you understand its issues and be able to make realistic questions about the matters you're interested in. Information can be divided into 3 groups: General information of the organization's vertical structure, information about the organization and information about the units that directly relate to the current issues.

In this material, an example of library management system of Institute of Information Technology (IOIT) will illustrate through the survey, analysis and design process.

The main purpose of library management is to serve readers. A library management system is required to receive books and magazines number or code them, store and manage them, produce index of document.

Main contents of survey of the Library management system are:

- Relations between the current library management system and other library management systems;
- Main jobs of the library management system;
- The jobs that need to be improved by the new computer system.

In this survey, we concentrate on interviewing method.
The interview questions should consist of the followings:

- What are the main jobs of the library management system?
- What are the main jobs of the library management system?
- What is the function of each job?
- How are the jobs currently done?
- Who is responsible for them?
- What are the restraints and difficulties in each job?
- Who does the library serve? And so on

Base on the answer of the librarians, we can continue asking them to get a clear vision of the library management system, and all the requirements for the new library management system on computer.

REQUIREMENTS OF A SYSTEM

Theoretically, feasibility report* should be written in user's language (not necessarily non-technical language). Technical parts for designing should be put into an appendix.

All reports must have a cover page with name of project, its author, address, contact numbers. Cover page is followed by content pages with the following main items:

- Objectives of the system;
- Inter-connection between related departments;
- Details of the current system;
- Future system and sketchy estimate of costs and benefits;

- Advice;
- Time frame and plan for system development;
- General description (non-technical);
- Original document;
- Evaluation of the current system in terms of: organizational structure, technology, information system, users' IT skills, policy renovation...

The following points should be included in the conclusion part of the report:

- Is the document of the current system adequate?
- Have users reviewed and agreed with your point of view?
- Have users been consulted and has analyzer addressed their concern properly?
- Has the whole report been researched thoroughly?
- What functional requirements need further research?
- Have all requirements been reviewed?
- What are the substitute designing solutions?
- What are the project's possible changes?

SPECIAL PURPOSE RECORDS

Sometimes the existing records do not supply the information required, and the only way of obtaining reliable information may be to install, for a limited period, special-purpose records. The missing information is likely to be quantitative, concerned with volumes, frequencies, trends or ratios, " or it may relate to management information requirements. The kind of information that one might wish to gather in this way would include, for example, the time at which documents arrive in a department, the number of times files have to be-consulted, the volume and frequency of telephone conversations, the number of queries to management, the type of enquiries from management, etc.

Account needs to be taken of the fact that keeping the special records will be additional to the normal work done, and therefore must affect, however slightly, the volume of work being processed. .To minimise this effect and cause as little ill-will as possible, the record should be simple: it may consist of doing nothing more than making a mark on a line or in a column whenever a particular activity takes place, or making an extra carbon copy of each document produced. To avoid the suspicion that it is a permanent addition to the work -load, a specific duration should be set, i.e. whatever minimum time will give a representative sample of what is required to be known.

SAMPLING

Where there is a high volume of documents or a number of recurring activities, it may not be necessary or practical to measure the total number. The economical way may be to take a sample. The safest way is to use random numbers, but it can be acceptable to count every 'n'th document or take a reading every ‘n’ minutes, so long as 'n' does not coincide with some cycle in the occurrence of the types of document or activity.

The sampling technique can be used to reveal volume, frequencies, trends or ratios. It is also commonly used as a method of work measurement.

WEEK 6

OBJECTIVES:

The students are expected to learn the following:

- **System design,**
- **System specification,**
- **Program specification and**
- **Documentation**

LOGICAL SYSTEM DESIGN

INTRODUCTION

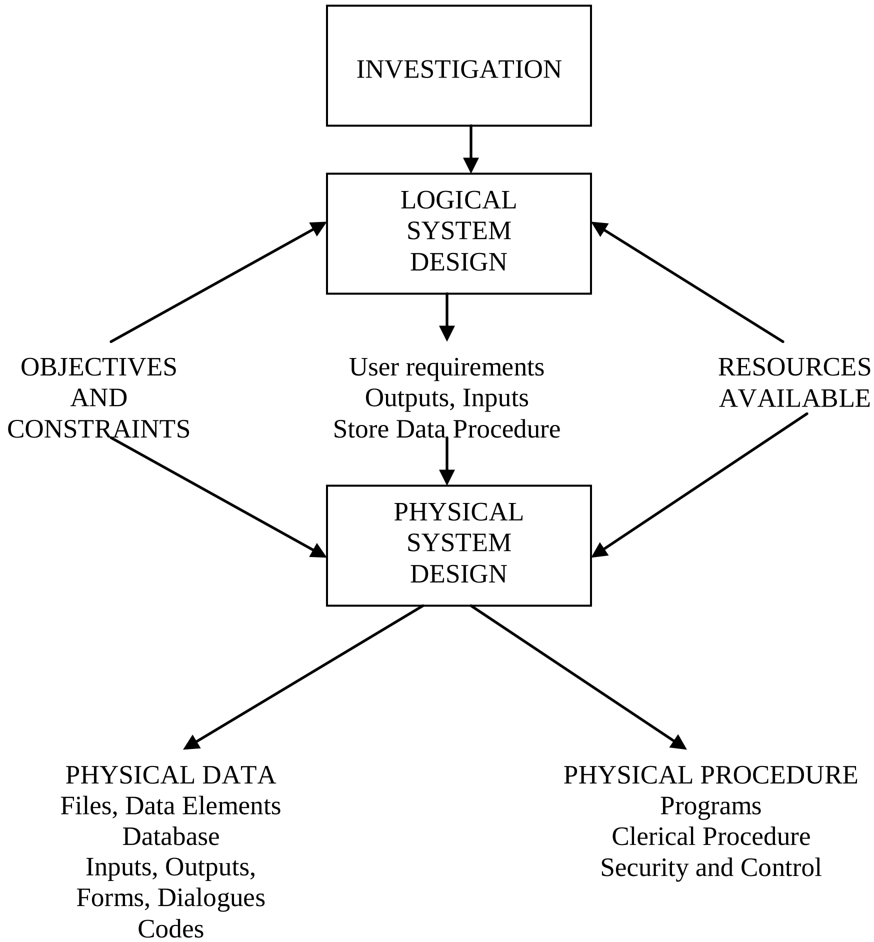
As a result of analyzing user requirements, the systems analyst has some conception of the new system. The next stage is to define the system in terms of a User System Specification. This process is called logical system definition because the system is still being conceived in logical terms (i.e. what the user requires); it has not yet been converted into physical terms (i.e. how the requirements are to be achieved in terms of hardware, software, equipment, people, procedures, etc).

The systems analyst's technical knowledge will inevitably affect this process. At the same time, the aim is to avoid letting technology dominate or dictate either the identification of requirements or design of data processing systems to meet these requirement

In logical system definition, a number of factors have to be considered:

- The analyst must clarify the specific objectives of the design process;
- The results of analysis must be converted into an outline of outputs from and inputs to the system;
- It is necessary to structure the data which will need to be stored in order to produce the outputs;
- It is necessary to partition the outline system into computer and clerical subsystems;

- It is necessary to consider the nature of the processing needed to meet the user requirements (e.g. batch processing is on line processing).



DESIGN AS A CREATIVE ACTIVITY

While analysis is essentially logical, design is essentially creative. The elegant design achieves its objectives with minimum use of resources. Resources are always limited, and the acceptable design often proves to b a compromise between a number of theoretical ideals and resources available. To say that design is a creative activity does not mean that it consists simple of a series of bright ideas. Design requires a full understanding of the problem: there is a need for analysis of requirement and resources.

The essential feature of the creative process is the association previously unconnected concepts. A technique which may produce result for the individual working in isolation is the one known as 'morphological' analysis. This consists of listing all the elements of the problem and associating each item of the list with each other item. The grid chart provides a useful way of laying out the two lists.

If there are several people involved in the problem area, then the technique of 'brainstorming' may produce useful results. The essential features are:

- no idea is subjected to criticism;
- all ideas are acceptable;
- the wilder the idea the better;
- combinations of ideas and improvements of ideas are sought.

Following a lapse of several days, the ideas are evaluated. Ultimately an idea has to be evaluated against other ideas and checked against the user requirements and the available resources. Finally, it has to be checked against the alternative uses of the same resources. This may seem to reduce the creative element to a small position in the total process; it is still an essential element.

DESIGN OBJECTIVES

At an early stage in defining a new system, the systems analyst must have clear understanding of the objectives which the design is aiming to fulfil. These objectives must be established by management and included in the terms of 'reference for the project. There is usually more than one way of achieving a desired set of results. The acceptable design is likely to be *a* compromise between A number of factors; particularly, cost, reliability, accuracy, security, control, integration, expansibility, availability, and acceptability.

Cost

Cost is associated with the two activities of *development* and *operation.* Development comprises all the stages from initial design to successful implementation. Operation includes data preparation, processing, and-handling of output and consumables (particularly paper). Operation also includes maintenance of the system: it is very rare for any computer-based system to run for years without substantial changes, to take account of changing business requirements or changing hardware and software. In fact, it is the experience of many long-established computer installations that the major part of their analysis and programming effort is devoted to maintaining existing systems. Clearly the first objective is lo minimise costs in these areas.

Reliability

This includes the robustness of the design, availability of alternative computing facilities in the event of breakdown, and the provision of sufficient equipment and staff to handle peak loads (whether seasonal or cyclical).

Accuracy

The level of accuracy needs to be appropriate to the purpose. For instance, the accounts of an individual customer will probably be kept to the nearest penny, whereas the monthly sales for a region may only be required to the nearest £1000. For each defined level a balance needs to be achieved between avoidance of error and the cost of avoiding the errors.

Security

There are many aspects to security, but the ones which particularly concern the systems analyst are confidentiality, privacy, and security of data.

Confidentiality

Some information, vital to the success of a firm, could cause severe damage if it reached the hands of competitors. The system has to ensure that only authorised staff have access to such information.

Privacy

This concerns information about the individual employee or members of the employee's family. If the personnel file or the payroll file is held on the computer, the design of the system must guard against any unauthorised access to the information.

Data security

If the data held on computer files is incorrect, then the system objectives cannot be achieved. Measures are needed to guard against alteration or destruction of data, whether accidental or intentional.

Control

The system should give management the facility to exercise effective control over the activities of the organization. One way is the provision of relevant and timely information, particularly by extracting the important information from the mass of available but less important information: this is lho principle of 'exception reporting', able to produce *ad hoc* reports. Another essential approach is routine control of goods and monies handled.

INTEGRATION

Before the advent of a computer, it is usual for each department to keep its own records. This allows the individual departments immediate access, but often causes disagreements. This can result from errors in copying, from source documents being mislaid, or simply from the time-lag between an action occurring (for instance, an item being withdrawn from stock) and the occurrence being entered on the various sets of records.

The ideal, when a computer-based system is being designed, is for all files relating to a given item of information to be automatically updated as a result of a single input. Where this is achieved, it becomes even more important than before to avoid any source document being mislaid before the updating takes place. This means that the systems analyst must be concerned, not simply with the computer procedures, but also with the clerical procedures in user departments and the computer operations department (ie where effective control over the source documents needs to be exercised).

The different systems which make use of any item of information may be designed at different points in time: there is a need for consistent documentation, regardless of who is doing the design.

EXPANSIBILITY

Any system needs to be able to cope with seasonal or cyclical variations in volume. Estimates need to be made of volume trends, and there has to be provision for handling whatever load the trends predict for the expected life of the system.

AVAILABILITY

It is the responsibility of the systems analyst to ensure that all the resources required to make the system work are available at the planned implementation time. These can include buildings, hardware, software, stationery, computer operations staff and procedure manuals, as well as the fully tested and working system.

ACCEPTABILITY

A system which the analyst believes to be perfect is certain to fail unless il has the support of the user departments and management, as well as the support of programming and computer operations.

A system which affects people (and there are very few systems which do not) needs to be agreed by staff representatives, possibly the relevant trade unions, as well as by the individuals whose jobs will be changed. A system which concerns use of assets needs to be acceptable to the auditors.

DESIGN TOOLS

Various tools are used by systems analysts to specify computer procedures, e.g. narrative, flowcharts and decision tables. These have already been described (Chapter 7) and will be discussed here only from the viewpoint of their specific use in computer procedure design.

FLOWCHARTS

Flowcharts of computer procedures can be drawn at various levels. In theory there is no limit to the number of levels to which a system can be divided the convention in this book is for the overall system to be defined by a System Outline (fig. 7.7vol. 1) for an overview of the total system, and by a System Flowchart (fig. 7.4 vol. 1) for a more detailed examination of both computer and clerical procedures. System Outlines and System Flowcharts can be drawn for the complete system or for subsystems: for example, there may be a System Flowchart for the complete production control system, and a series of System Flowcharts, one for each subsystem, Such as Order Entry, Stock Control, Production Scheduling, at a lower level of detail.

Once the overall system or subsystem flowchart has been produced, the boxes which represent computer procedures have to be defined in detail and a number of types of flowchart are used for this.

COMPUTER RUN CHART

Computer Run Chart is a way of showing the inputs, processes and outputs in the computer part of the system in a logical structure. It can be regarded as a master plan of the computer subsystem, but cannot be used to show details except by cross reference. Figure 1 3. 1 shows a Computer Run Chart for a computer despatch procedure. It shows the inputs, master files, presses, transfer files and outputs, in that order, from left to right across the sheet with the flow of data shown by the direction of lines and arrows. Inlets outputs, and master files symbols bear the reference of their appropriate data specifications, and the process symbols of their procedure specifications at a lower level. The symbols used are part of the set shown in figure 7.3 (vol. 1).

Consequence of processes, execution of the first is followed by execution of the subsequent ones in the order indicated. Such a processing sequence excuted at predetermined periods, e.g. daily, weekly, monthly, implies ba… processing; if executed as an immediate response to a single transaction it implies real-time processing. In batch processing, the same procedures (e.g. input validation) may be included in a daily run chart, as well as a weekly (input - validate - update - print) run chart.

Computer Run Chart is use to group logically-related sequences into specific procedures; it is not necessarily expressed in terms of programs; and a process symbol on a Computer Run Chart does not necessarily refer to *i* program. This run chart will subsequently be replaced by a pr. crammer-produced run chart which will specify programs. These may coincide with the procedures as specified by the systems analyst, or may be subdivided or aggregated, into programs. Any coincidence may well depend on the relevant programming knowledge of the systems analyst.

Where the boundary lies between the work of the systems analyst and the programmer will depend on several factors, e.g. their working relationship, particularly in project teams, and any standards laid down "by the project leader or data processing management. Where the systems analyst designs program structures, this assumes an appropriate experience of the programming language, software and operating system used in the installation.

COMPUTER PROCEDURE FLOWCHART

A Computer Procedure Flowchart depicts in more detail the process symbols shown on a computer run chart. An example is given in figure 13.2.

The processes are analysed to define their internal logical steps and then organised into procedures which best satisfy these operational requirements.

The detailed steps within a procedure level may be as small as individual machine instructions or as large as groups of instructions (forming a macro instruction, subroutine or segment of a process or run). To define the levels of design of Procedure Flowcharts, each symbol is examined for relevance and detail required; if further detail is required, an operation symbol can be cross-referenced to another Procedure Flowchart and expanded into still further lower levels of detail.

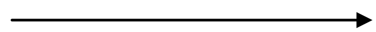
For each operation symbol on a computer run chart, there will be one or more levels of Procedure Flowchart to show in detail how that computer operation is executed.

NETWORK CHARTS

Network charts (sometimes called Phillips diagrams) are used to depict the interactions between components of a complex system or program, especially those used in real-time mode. They show, at a general level, how events *can* happen, not necessarily how they *will* happen. They provide in effect, a map of the system program which shows all possible routes, and are used when the actual sequence of events cannot be readily specified or is unpredictable.

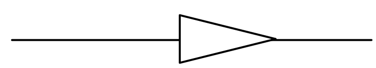
In Network Charts the data movement triangle is used to qualify, the transfer lines, as indicated below.

Direct Control - General



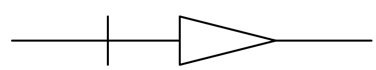
- Transfer of control from one process to another;

Direct Control Temporary

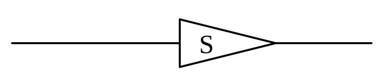


- Transfer of control from one process to another, with direct return to the calling process when the called process is completed;

Direct Control Permanent



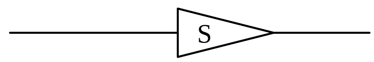
- Transfer of control from one process to another with no guarantee of return to the calling process;



Indirect Control

- Activation of one process by another via an operating system or scheduler. An alphanumeric symbol within the triangle indicates the type of activation, e.g. 'S' for Scheduler';

Interrupt



- Activation of a process by means of an interrupt generated externally, e.g. by a peripheral device.

It can be seen (fig. 13.3) that arrows are used more frequently in network charts than in flowcharts, and that two lines are used to indicated two-way flow. Also is should be mentioned that these charts must always be supported by lower level documentation which specifies data and processes involved.

INTERACTIVE SYSTEM FLOWCHART

An Interactive System Flowchart is used to depict the sequence of events in an on – line system where the user is in control of the program at execution time and can enter data between execution steps or determine the sequence of operations. The Chart (fig. 13.4) is divided into three main areas showing, from left to right, terminal user procedures, terminal formats and computer procedures. Each area may be subdivided into columns to show different levels or locations of operation. Procedure boxes are cross – referenced to their detailed procedure specification; terminal formats are cross-referenced to display charts (fig. 12.4).

SYSTEM SPECIFICATION

Once management has authorized the start of design and development work for a new or amended system it is necessary to provide user management and staff with information about the main features of the system and how it will affect them. This information is contained in the User System Specification, and should form the basis for any final agreement with those who are expected to use the new system. Agreement should be reached before detailed design work is completed.

The specification should not require t be updated once the contents are agreed. Subsequently, however, it may be possible to produce the User Manual from the User System Specification. The specification also plays an important part in user education and may be used for staff familiarisation before the User Manual becomes available. It should therefore be presented as a training aid, and not as a reference manual.

The contents checklist provided below contains cross-references to working documents. However, working documents should only be used when it is known that these will be acceptable and understandable to the user. It may often be preferable to present data or procedures in plain-language terms.

Introduction:

brief introduction to the system;
reasons for introducing changes;
statement of problems;
objectives of the new system and expected benefits.

Procedure summary:

one page summary showing major changes and explaining principles of the new system;
system outline;
system flowchart;

Procedure specifications:

description of the clerical and interactive procedures within the system;
brief description, using non-technical terminology, of the objectives of the new/revised computer procedures.

Data:

samples of mock-ups of input and output documents and displays; specification of new clerical files; ,

summary description of computer-based data files, showing which/ files they replace.

Supporting information:

Organisation chart showing lines of responsibility in terms of the new system; document/department grid.

Changeover:

plans for change to a new system;

timescales, critical activities, and work loads.

Operations:

anticipated schedules;

deadlines and critical points.

It should not be forgotten that users will ultimately have full responsibility for running the system. Computer processing, carried out in the computer operations department, is done on behalf of users. The user departments will pay for development and operation of the system, either directly or indirectly through the management accounting system.

Such a specification is often the users' last opportunity to request changes where the design fails to meet their requirements. Once they have accepted the specification, it is often declared to be 'frozen', ie, no amendments will be accepted during the remainder of the project.

PROGRAM SUITE SPECIFICATION

Once the final system proposal has been accepted by management the systems analyst has the task of specifying the computer functions of the system to the programmers. The Program Suite Specification must provide sufficient information to the program development team about the computer functions of the system to enable programs to be developed and tested to a defined level. The content will, therefore, vary between different installations and between different teams. Much will depend upon the point at which program development is distinguishable from system development. Other factors are organisational structure, experience of the people involved and design novelty.

At one extreme the information that has to be communicated may be little more than a statement of the objectives to be met by programming activity, together with a broad description of input, output and maintained files. At the other extreme the information may be so detailed and complete that coding can commence following receipt and examination of the specification. Where programming is a function of a multi-discipline "project team', and there is no readily distinguished transfer of responsibility, there may be no need for a formal Program Suite Specification.

The specification will be produced to form the basis of sub-contracting responsibility for the programming work. The handover of the specification represents an identifiable control point in a development project and usually involves locally-defined procedures such as for authorisation, specification appraisal and acceptance and handling of queries. Local standards must-also provide for specification maintenance.

The content of the Program Suite Specification will be a subset of the documents maintained in the New System File, the subset being determined in the light of local requirements. For ease of subsequent referencing and maintenance the specification should be organized in the same sequence as the System File from which it is derived.

WEEK 7

OBJECTIVES:

The students are expected to learn the following:

- **Database design**
- **The schema**
- **Logical and physical database**
- **Physical data organization**
- **Terminologies across references**
- **Designs of the structures of a database file**

DATABASE DESIGN

The organisation of data in a database aims to achieve three major objectives: data integration, data integrity and data independence.

Data integration

In a database, information from several files is co-ordinated, accessed and operated upon as though it is in a single file. Logically, the information is centralised ; physically, the data may be located on different devices and in widely scattered geographical locations, connected through data communications facilities. In order to achieve the objectives of data centralisation, links between data must be maintained; direct-access techniques are used to permit efficient and flexible linking, although sequential organisation can be used in a database.

It must be possible to access data records using a wide variety of search keys. This can reduce the costs of implementing new applications, where multiple references are made to the same data, eg order processing of home, overseas and government orders at various factories. Data integration is achieved by using the techniques described later, chaining, data inversion and indexing.

Data integrity

Very often, within the same computer system, reports or analyses referencing the same logical information are inconsistent owing to differences in duplicated physical data. This could, for example, occur when changes are made to data in one file but not to a copy of the same data in another file.

One way to solve this problem is to ensure that when a field is updated, all other copies of that field are updated at the same time. This becomes difficult when copies of the field are held in separate files which are used by separate programs. Another way to solve this problem is to store all data in one place only and allow each application to access it; this is the database approach. This approach to data integrity results in more consistent information; one update being sufficient to achieve a new record status for all applications which use it. This leads to less data redundancy: data items need not be duplicated. There is also a reduction in the direct-access storage requirement. It will probably not be possible to achieve complete non-redundancy of data, due to performance, security and back-up requirements, but this approach makes it possible to control and maintain redundancy at minimum levels.

Data independence

Applications evolve as more information is required, and its. usage improves. Changing requirements will influence the need to use stored data differently. Conventionally, this results in a certain amount of modification to the physical data to meet these changes, with perhaps an increase in data duplication and redundancy.

Conversely, when control and optimisation of the system are improved by the reorganisation of physical stored data and the installation of new hardware and software, existing applications must recognise the new physical data organisation; and so modifications to application programs are necessary.

Data independence is the insulation of application programs from the changing aspects of physical data organisation. This objective seeks to allow changes in the content and organisation of physical data without re-programming of applications; and to allow modifications to application programs without reorganising the physical data. The DBMS will match new and different programs to the data by relating their subschema to the existing schema; new physical data structures will be accommodated by cross-referencing the new organisation to the existing schema. The concepts of schema and subschema are described below.

LOGICAL AND PHYSICAL DATA

The concepts of logical and physical data can be illustrated by a set of filing cabinets, a filing clerk, and some departmental managers who use the files. Physical storage is represented by the riling cabinets, the units of physical storage being the cabinets themselves, their drawers, the suspension folders, and the pieces of paper they hold. Physical data is that which is written on those pieces of paper. Let us assume that the filing system supports a motor insurance activity, and the data is concerned with policy holders, claims and other such information. One manager who is concerned with handling new claims instructs the filing clerk to retrieve certain data, such as customer details and previous similar claims from the files.

The filing clerk, who understands the structure of the filing system, retrieves the required data, and presents it to the manager in accordance with the instructions; perhaps, in a certain sequence in summary form after having made calculations based on some of the retrieved data. This is the logical data for that department manager. Other managers may require the same data, but with different instructions and results. In this example, the filing cabinets are equivalent to direct-access storage devices, the managers are programs, and the filing clerk is the DBMS.

If a new filing system is introduced, the only retraining necessary is that for the filing clerk; the managers may view the data in the same way as before. When managers wish to change their method of working, or perhaps a new manager is appointed, the filing clerk is informed of the new logical data requirements and uses the same physical data as before to fulfil them.

A manager never retrieves data directly from the filing cabinets. Because the filing clerk is the only person who knows how the data is organised and how the filing system works; the filing clerk is the only means of access to the physical data.

In a computer-based data processing system, separation of physical and logical data provides the same advantages as in our filing cabinet example. Application programs may be restructured or replaced without damage to the data. The data can be re-organised on the same storage device or new device types without change to the programs. This is because the DBMS is able to interface logical and physical data, and make the necessary adjustments when one or the other change. Because the DBMS is the only way of accessing the data, privacy, recovery and integrity controls are easier to implement.

Database terminology

In order to promote the portability of software among computer systems, standard terms are often employed, particularly in standard compilers such as COBOL or FORTRAN. This in itself does not guarantee portability but at least allows a common basis for design and conversion. Beyond the realm of compilers, very little has been achieved in the field of standard terminology, and this is true of database systems; different systems are often unrelated and use entirely different terminologies.

However, the Data Base Task Group (DBTG) of the Conference on Data Systems Languages (CODASYL) has proposed specifications for non-manufacturer-oriented terminology for database systems. (Details of CODASYL recommendations are given in Appendix B.) The specifications have not been used in all DBMS implementations, even those which claim to follow the CODASYL recommendations. A wide variety of terms to describe the nature of logical and physical data are used but this section will begin by explaining the CODASYL terms.

Logical data is described in terms of DATA ITEMS, DATA AGGREGATES, RECORDS and SETS. Physical data is described in terms of AREAS and PAGES.

Logical database units

In the motor insurance application example, the customer details could be a RECORD consisting of a customer's name, address, vehicle details, details of previous claims, account number, policy numbers. When the details of a particular customer are required by the manager, the manager gives the filing clerk the customer's name, and the filing clerk takes the record from the files.

The customer name in this case is a KEY; if the files are organised in alphabetic sequence of names, then this is a PRIMARY KEY. If the filing clerk has first to match the name, in a card index, with a reference number, by which the files are organised, then the customer name is a SECONDARY KEY, and the reference number the primary key.

The contents of the record are generally grouped together either on one sheet of paper, or on several sheets of paper clipped together or in a folder; if a particular customer's record contains many details, it may be necessary to use a second folder. When a customer is a large organisation with numerous policies and vehicles, then only recent data may be stored in the main files, other less frequently used data being removed to separate filing cabinets, perhaps in a separate location such as an archival store. The record may thus be retrieved in part or in whole.

In this case, the manager should specify the parts of the record required; for example, a manager requiring details of customers' accounts may not also need details of vehicles or policies. The filing system can accommodate this, and make life much easier for the filing clerk, by abandoning the record format which contains all customer details, and storing separate records for each client's name and .address (e.g. header information), policies, accounts, and claims. Under these conditions, if the manager who handles claims requests the details of a certain customer arid past claims, this may be done by first including the claim numbers on each client's header record. The filing clerk then matches and retrieves the client's header record, notes the claim numbers, then matches these numbers against the separate claim records file and retrieves the detailed claim records.

Alternatively, the relevant claim records may follow the client header records in the files. In this case, the manager is referring to an association between different record types, this association is called a SET. The filing clerk is fetching an OCCURRENCE of the set. Another set could be a client header record, accounting records, and policy records; here, there are three record types (there is no particular limit). A set must have a unique identifier or primary key, this means that one of the record types in the set has a primary key \\hich is also used to identify a particular occurrence of the set. This record t\pe is the OWNER record type of the set and must therefore have an occurrence of only one per set. Others known as MEMBER record types can have any number of occurrences. From time to time, the manager will refer to the various contents of each record.

Details such as account number, policy number, premium amount and vehicle type will probably be DATA ITEMS; other details such as dates, addresses and claim numbers may be DATA AGGREGATES. The difference is that a data item cannot be further subdivided, while a data aggregate consists of a number of data items, but can be referred to as a whole. For example, date has three data items, namely, DAY, MONTH & YEAR, and address has STREET, TOWN, COUNTRY, etc. The distinction between these two units must be made for two situations. Firstly, where a larger unit of data is always used in a program, but only a part of it may be updated, eg, in date, all three data items may be moved in a program, but only one changed; secondly, where a repeating group is involved, e.g., a premium received may consist of data items policy number and amount, but may exist several times for one policy.

Each occurrence of a data item will have a specific value (for example the value of premium amount' may be '£91 25'' or the value of vehicle type' may be 'saloon car'); these values will be represented in a suitable code.

These logical data units can be transferred fairly easily from the clerical example to one where programs, DBMS, and direct access storage replace the manager, filing clerk, and filing cabinets. Programs may request an occurrence of a SET by providing either a primary or a secondary key (the primary key of a member record). A number of

RECORDS will then be retrieved for the program, which will manipulate the DATA ITEMS and DATA AGGREGATES stored therein.

Physical database unit

Input to and output from the central processor in a computer system is accomplished by a command which causes a secondary storage device to be read from one indicator to another, and the data thus found transferred to main memory. In the case of a magnetic tape, the indicators will be inter-block gaps; in the case of magnetic disks or drums, the indicators will be address markers.

All the data recorded between these indicators form a physical record sometimes known as a block or sector. A PAGE may be one or more of these (contiguous) physical records. In practice, it is unlikely that a page will be longer than a track of a magnetic disk, and it should be possible to perform a multiple physical record read in a short time to retrieve a page. Where it is possible to vary the size of physical records in a computer system, positive performance advantages will be gained by making the page size exactly equal to the physical record size. A page can contain a number of records, and a record can span more than one page. There may be different record types to a page because there is no restriction on record/page relationships.

An AREA is simply a collection of pages, normally contiguous. There is no need for an area to be restricted to disk pack, cylinder or track boundaries, as an area will consist of a multiple of pages. The specification of areas provides a means of physically segmenting secondary storage; the page is generally that part of an area which is read into main storage at every input or output command.

It is the job of the DBMS to match requests from programs for set or record occurrences to area and page identifiers. (This is the CODASYL approach, other approaches are different in detail.) The database approach enables the introduction of the concepts of VIRTUAL data and TRANSPARENT data: VIRTUAL data is the term given to data item types whose values are calculated from other stored data items and presented to a program only when they are requested by that program. TRANSPARENT data is the term given to data items whose values are for the benefit of the DBMS, and are never seen by the programs.

Terminology cross-references

While an explanation of the CODASYL terms is useful in conveying concepts, other types of DBMS use different terms, and occasionally different concepts. Here is a list of other terms used and their CODASYL equivalence:

ELEMENT
FIELD
DATA ELEMENT } DATA ITEM
DATA FIELD
ELEMENTARY ITEM

GROUP ITEM
REPEATING GROUP
VECTOR } DATA AGGREGATE
SEGMENT
ENTRY } RECORD

FILE.................appro. ALL OCCURRENCES OF A RECORD TYPE; or
DATA SET......approx. ALL OCCURRENCES OF A SET TYPE.

These are all logical data units; physical data units are so widely variable according to manufacturer that such equivalence is almost impossible. All terms which refer to large areas of secondary storage, such as, EXTENT, GRANULE, and in some cases POSITION and CELL, may have some equivalence to AREA. Terms which refer to smaller elements, such as BUCKET, SECTOR, BLOCK and CELL (again) may be equivalent to PAGE. Because of wide differences in physical data notation, equivalence between terms in this area will be useful only in comprehending a move from one manufacturer to another, and should be of concern only to those who will be responsible for physical data storage.

The Schema

Once a database system has been designed, it will be possible to identify each type of data item, data aggregate, record and set, by a name or code. It will be possible to state which data item types go together to make data aggregate types and record types, and to identify which record types are members and owners of set types. A coded set of tables describing this information and stored in the computer system on direct access devices is called a SCHEMA. It is a description of the data structure which is separate from the data itself.

In the CODASYL recommendations, the Schema also describes the areas, their identifiers and page sizes, and indicates how these are related to the records and sets. In other systems, a different set of tables is used for this.

The Schema, therefore, is the view of the data, the overall logical data structure, which is held by the DBMS. Each time a program requires data, the DBMS will look up in the Schema for the details of the structure of the data requested; for example if the program requires an occurrence of a set, the DBMS will look up in the Schema which record types are required, how to find the relevant records given a certain key by the program, and perhaps also which areas the pages containing the relevant data are stored in.

The term schema is used by CODASYL, but any collection of data that describes logical data structures may be seen as a type of schema. The view of the data as seen by the program may be different from the schema which is used only by the DBMS. The program will only know of the existence of certain records and data items, and these may be in a sequence different to the sequence of storage or DBMS reference. There may be other differences between the data used by the program and the data as seen by the DBMS.

The program will need its own description of the logical data structures as well, which can be achieved by a preamble to the program itself, such as the DATA DIVISION in a COBOL program.

In a database system, it is not always possible to allow programmers to write the data division of their choice for reasons of security or control. It is more usual to provide the programmer with a standard description of the logical data to be used in a particular application. All references to data within the program will be to this description, which is called a SUBSCHEMA by CODASYL and is similar to the SCHEMA in structure.

The DBMS has the job of matching data requests on a subschema and data requests based on the schema (fig. Bl appendix B). In addition to this, the DBMS must match schema-based requests to physical data requests, based on the physical storage structure.

PHYSICAL DATA ORGANISATION

Database systems employ techniques for organising data which are designed to provide different logical record sequences from a single physical sequence.

Ran existing physical sequence:

- a partitioned file;
- a chain file;
- an inverted file.

Each technique has its own variations, and each method of representation is applicable in specific environments.

DESIGNING A STRUCTURES OF A DATABASE FILE

Databases are normally implemented by using a package called a Data Base Management System (DBMS). Each particular DBMS has somewhat unique characteristics, and as such, general techniques for the design of databases are limited. DBMS fall into two broad categories: 'pointer-driven' systems and 'table-driven' systems. The latter are inverted-file systems which allow the user to set up and maintain a database which may be searched using a wide range of different keys, via either a set of commands supplied with the package, or 'calls' from a host language such as COBOL or PL/1. These systems are generally straightforward to implement; the user specifies the records and fields, indicates which of the fields will be keys, and supplies these parameters to the DBMS, which will then set up the database. Pointer-driven DBMS use techniques such as partitioning and chaining. These are normally host-language systems which use high-level language verbs coded within application programs. The design of the database using such systems will have a crucial effect on the performance and flexibility of the end result. Complex data structures may be used, and the effort needed at the design stage is much greater than with table-driven systems.

Good examples of pointer-driven DBMS are the packages produced to the CODASYL recommendations. Techniques for designing the logical data structures for CODASYL Data Base Definition have been developed, and this section examines one in detail.

A most important step in the process of CODASYL database implementation is the design and implementation of the SCHEMA. The design features of the schema will have-a significant effect on the performance and flexibility of the systems using it.

Key Item	Data Item	Data Item = Key?	One-to-one One-to-many Many-to-one Many-to-many	Length of Data Item	F or V	Occurrence of Data Items	R or S
empl. no.	Name	Yes	many-to-one	32	f	n	S
"	Address	No	one-to-one	0-200	v	1 per empl.	R
"	nhi no.	Yes	one-to-one	10	f	"	R
"	date of birth	No	many-to-one	6	f	"	R
"	dept. no.	Yes	many-to-one	8	f	n	S
"	job no.	Yes	many-to-one	8	f	n	S
"	Grade	No	many-to-one	8	f	n	R
"	Salary	No	many-to-one	8	f	n	R
"	bank no.	No	many-to-one	6	f	n	R
"	bank acc. no.	No	many-to-one	8	f	1 per empl.	R
"	skill detail	No	one-to-one	0-200	v	n	S
"	Experience	No	one-to-many	0-200	v	n	S
"	Education	No	one-to-many	100	f	< 20 per empl.	R
skill type	skill detail	No	one-to-many	0-200	v	n	S

The next stage is to select the Key Items which are PRIMARY KEYS. A primary key should be unique to a record (in exceptional cases where certain page addressing techniques are used, the primary key may not be unique). A primary key should be suitable for use as a parameter for storing the record, fixed length, reasonable size. The primary keys will also be values which are available to a program when it is run, these may be values from input documents, or input or master data. Once a primary key has been established, all other data items which are related and have R in the final columns, become data item names in the record for which the selected key item is the primary key. This may be repeated for all primary keys until a list of record types, their primary keys and other contents have been established. The remaining entries in the matrix will be represented as sets.

Using the example above as an illustration of this technique, the first primary key is employee no. and relates to the following data items having R in the final columns:

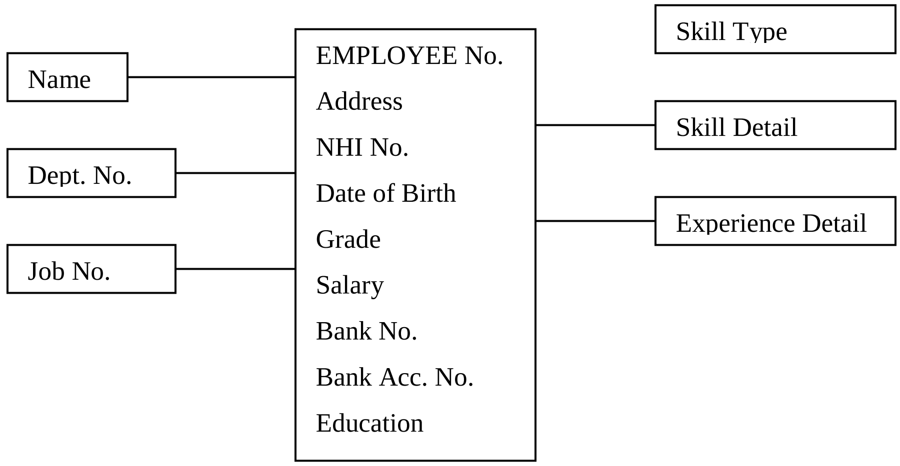
- address;
- NHIno.;
- date of birth;
- grade;
- salary;
- bank no.;
- bank ace. no.;
- education.

These can be grouped as a record with the primary key.

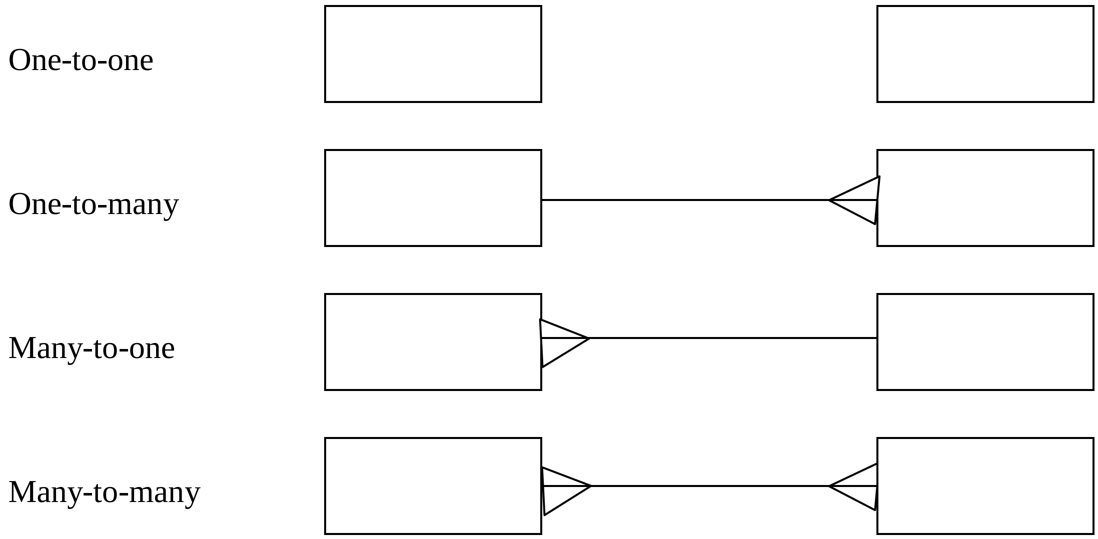
All the other relationships involving Employee No. will be represented as sets, namely:

- name;
- dept. no.;
- job no.;
- skill detail;
- experience.

The other primary key will be skill type, which is related to Skill Detail by a set. Thus, the following record types and set types are shown in the schema:



This diagram will form the basis of a schema. The complete documentation of a schema will include a number of indications showing primary keys and set characteristics. To complete this preliminary diagram, information regarding the basic characteristics of the sets must be shown; the data structure diagram conventions in chapter 9 can be used:-



Using these conventions the schema can be redrawn showing the basic characteristics of the sets, as facing page.

WEEK 8

OBJECTIVES:

The students are expected to learn the following:

- **Input capture/stages,**
- **Input types**
- **Error avoidance**
- **Input specification**

INPUT AND OUTPUT DESIGN

Input design is a part of overall system design which requires very careful attention. Often the collection of input data is the most expensive part of the system, in terms of both the equipment used and the number of people involved; it is the point of most contact for the users with the computer system: and it is prone to error. If data going into the system is incorrect, then the processing and output will magnify these errors. Thus the designer has a number of clear objectives in input design:

- to produce a cost effective method of input;
- to achieve the highest possible level of accuracy;
- to ensure that the input is acceptable to and understood by the user staff.

INPUT CAPTURE AND STAGES

Several activities have to be carried out as part of the overall input process. They include some or all of the following:

- data recording (i.e. collection of data at its source);
- data transcription (i.e. transfer of data to an input form);
- data conversion (i.e. conversion of the input data to a computer acceptable medium);
- data verification (i.e. checking the conversion);
- data controel (i.e. checking the accuracy and controlling the flow of the data to the computer);
- data transmission (i.e. transmitting, or transporting, the data to the computer);

- data validation (i.e. checking the input data by program when it enters the computer system);
- data correction (i.e. correcting the errors that are found at any of the earlier stages).

Not all of these stages need to be present; for example, with on-line data entry, data conversion and verification are not usually necessary. Indeed, one of the aims of the systems analyst must be to select data capture methods and devices which reduce the number of stages so as to reduce both the chances of errors and the costs. Nor need the stages be carried out in the sequence listed; data control and data correction will be involved at several points, and data transportation may occur before data conversion.

INPUT TYPES

One of the early activities of input design is to determine the nature of the input data. This will have been done partially in logical system design but it now needs to be made more explicit. Inputs can be categorised as:

- external, which are the prime inputs for the system, eg sales orders, purchase invoices;
- internal, which are user communications with the system, eg file amendments, adjustments;
- operational, which are the computer department's communication with the system, e.g. job control parameters, file names;
- computerised, which are inputs in computer media coming from other internal systems or external systems, eg bank records passed over on magnetic tape;
- interactive, which are inputs entered during a dialogue with the computer.

In addition to identifying the input types the analyst needs to consider their impact on the system as a whole and on other systems. Often inputs to one system can be used as inputs to another; and inputs can be pre-processed or completely raw when entered into the system (this means that clerical procedures must be designed alongside inputs).

Input media

Once the input types and their contents have been examined the analyst can start to think about input devices, of which there is a very wide range. The first classification of input devices might be source-document conversion devices, like card punches, paper tape punches, key-to-tape, key-to-disk, key-to-cassette, and key-to-diskette. These are used for conversion of data from source documents into computer acceptable media.

There are also by-product data-capture devices such as billing machines, accounting machines, and cash registers; these are used to capture data in a computer-acceptable form as a by-product of some other essential operation. And there are direct data capture devices, such as Optical Mark Readers, Optical Character Readers, Magnetic 'Ink Character Readers, and Kimball and Datatag Tag readers; these devices are linked to the computer and receive the source document directly without any conversion or verification processes.

Another classification is on-line data entry devices, such as teletypewriters, visual display units, data collection devices, audio response terminals, light pens and optical wands, which collect data directly from the source document into the computer one transaction at a time.

Much careful thought has to be given to the choice of input media and devices. Consideration can be given to:

- type of input;
- flexibility of format;
- speed;
- accuracy;
- verification methods;
- rejection rates;
- ease of correction;
- off-line facilities;
- need for specialised documentation;

- Storage and handling requirements;
- automatic features;
- hard copy requirements;
- security;
- ease of use;
- environment of data capture;
- portability;
- compatibility with other systems;
- cost
- etc

These areas are explained in depth in books on data capture.

ERROR AVOIDANCE

Every effort must be made to ensure that input data remains accurate from the stage at which it is recorded and documented to the stage at which It is accepted by the computer. This can only be achieved by careful control each time the data is handled.

The conditions under which the tasks involved are carried out can affect the legibility and accuracy of the data. For example, dirty and damp conditions in office or factory premises can affect the fitness of both people and machines and consequently the effectiveness of the results; poor form design can lead to a misunderstanding of the instructions or insufficient space on which to write clearly; lack of control can lead to documents being lost or mislaid without the loss being realised. Any measures which are taken to improve these conditions are likely to reduce the amount of errors and loss of data.

The effectiveness of checking data by verification or sight-checking can only be assessed by keeping individual records of the preparations of input data and 'tracing' errors which are subsequently found by the computer, or even later in the system, back to their source.
Error detection

While every effort is made to avoid errors during the preparation of input dau. past performance shows that a proportion of errors is always likely to be present. Experience also teaches that the further into the system that errors are found, the more complex may be their effect and the more difficult to correct.

CONTROLS

Control checks can be carried out in various way?, usually at three level: batch, record and item.

A batch can be any collection of transactions, usually of a regular size suitable for checking and control convenience, or perhaps a total quantity or value processed per period (e.g. day, week, month). The transactions in each batch are totalled by the computer as part of the input, and then compared with controls previously prepared and fed into the computer with the user input data. Any discrepancy is then shown as an error which must be checked, corrected and re-input.

Errors may be in the control batch total itself, individual records items incorrectly totalled, keyed or overlooked somewhere along the line of input procedures; even a whole batch of transactions may have been mislaid. The control procedure must therefore be designed to detect errors at every level.

The systems analyst must, as part of the input design activity, design the system of controlling the processing of batches of data. This involves determining the nature of the batches (on the basis usually of location, timing or volumes); calculating the data volumes in relation to time and deciding on the make up of a batch; designing batch control slips and batch progress records; setting up procedures for batching and totalling and setting up procedures for error location and correction. The flow of a batch is illustrated

The batches will be fed into the computer accompanied by a control record, which may contain the following data:

- date of preparation of run or other data relevant to the process and unique to the run (e.g., today's date);
- run identification code;
- batch identification;
- number of records or groups of records in the batch;
- control totals e.g. totals of all values or quantities in the batch or hash totals of product codes or customer codes.

The position of the batch control record, whether in front or behind the relevant batch will also need to be specified.

Variations to this procedure are designed appropriate to the methods of data collection and input processing, e.g. on-line interactive data entry from visual display, audio response terminals, turn-round pre-recorded documents.

Data validation

Computer input procedures must also be designed to detect errors in the data at a lower level of detail which is beyond the capability of the control procedures. These are combined with the design of the input process itself, reading the data on the input medium from the appropriate peripheral unit, editing and transferring data to a magnetic medium for subsequent main processing, and sometimes including a partial sorting process.

The validation procedure must also be designed to detect each record, data item or field against certain criteria specified by the systems analyst for the programmer. Each type of record has codes to be checked for acceptability. As the record type indicates that a certain process or series of processes are to be performed, an incorrect or non-existent code must cause the whole of such a record to be rejected showing the reason by means of narrative or a set of error codes designed for this purpose.

Other input design considerations

Some of the other considerations which the analyst must take into account are:

- the nature of input processing, eg complex, involving all inputs, or separate for each input;
- flexibility and thoroughness of validation rules;
- handling of priorities within the input procedures;
- use of composite input documents to reduce the number of different ones;
- relationship with other systems, e.g. can the validation procedures carry out checks that might be required by another system which makes use of the input data?
- scheduling of input runs in case of large rejection rates at validation;
- forms design, to ensure accuracy and efficiency of input;
- relationship with files, e.g. types of data which can be stored or which need to be input at each run.

INPUT SPECIFICATION

Input files can exist in document form before being input to the computer, in which case their specification is described on a Clerical Document Specification.

When input data is converted into computer files, (card, paper tape or magnetic) they are described by a Computer File Specification and their detailed contents by a Record Specification . These data specification forms apply to all computer data, not only to input data. As an example of the record specification being used to specify an input file, a card input file is shown

WEEK 9

OBJECTIVES:

The students are expected to learn the following:

- **Structured fact and agreement,**
- **Flowchart,**
- **Flowcharting symbols,**
- **Principles of flow charting,**
- **Level of flowcharting and**
- **Relationships.**

STRUCTURED FACTS AND AGREEMENTS

During the investigation a lot of information is gathered as a result of interviews and discussions. Notes should be taken during an interview, but usually it is necessary to distill the essential facts afterwards. Some of the facts are best recorded in narrative form; others lend themselves to other methods, such as flowcharting and document specification. The notes should be turned into an appropriate record for future reference by the systems analyst or other members of the project team. The record can also be used for verification of facts by the person who provided the information. Both of these communication processes will be made easier if the record is in a standard format.

For the unstructured facts and agreements which need to be expressed in narrative form a Discussion Record form is recommended. This includes space for mention of the people involved, the objective or agenda of the meeting, its date, location and duration, and cross-referencing to flowcharts or document specifications which have been produced alongside the discussion record.

PROCEDURES

It is usually easier and more intelligible to record identified procedures in diagrammatic rather than in narrative form. The two main methods of doing this are flowcharts and decision tables. Narrative is occasionally necessary (for example to expand on the flowchart or decision table), but it should always be directly linked (ideally on the same sheet) to the flowchart or decision table.

FLOWCHARTS

A flowchart, using words to indicate a sequence of events, differs from normal narrative in that it does not use sentences. The words are enclosed in symbols (linked by flow lines) with conceptual meanings indicated by their shape. Flowcharting is the most common method of describing procedures in a computer-based system, whether for the benefit of a line manager, a clerk or a programmer. The advantages of flowcharts over narrative are that:

- they show logical interrelationships clearly;
- they are easy to follow;
- they allow tracing of actions which depend on conditions;
  they can be produced in a standard way and so allow several people to work on them simultaneously;
- they are useful to the systems analyst for experimenting with different approaches to a particular problem.

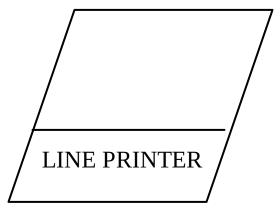
These advantages begin to decline as the flowchart increases in size and complexity and it becomes difficult to amend the logic. The two major disadvantages of flowcharts are: the difficulty of tracing back from actions to conditions; and the requirement to maintain a consistent level of detail in order to avoid confusion.

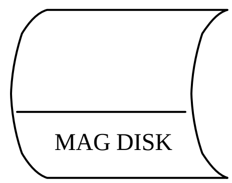
FLOWCHARTING SYMBOLS

There are several different sets of standard symbols for flowcharting, the main ones being those developed by ASME (American Society of Mechanical Engineers) and ECMA (European Computer Manufacturers Association). ASME symbols are used mainly by O & M and work study officers, ECMA by data processing staff. Neither of these is suitable for both clerical and computer procedures (ASME. for example, includes no decision symbol, and ECMA no movement symbol).

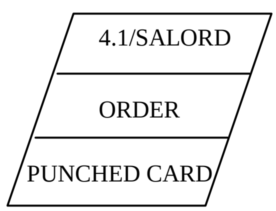
A set of symbols has been developed by NCC to meet the needs of both clerical and computer procedure flowcharting. The set of NCC symbols is small in number, each symbol representing a concept rather than a device. (If devices were used more than 30 symbols would be required and this would always be changing as new devices are developed.)

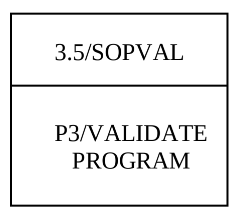
To indicate the device being used for input/output or file, a bottom stripe is used and the name of the device is written in; for example:

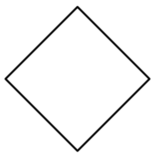
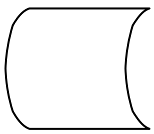
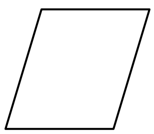
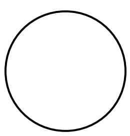
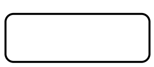
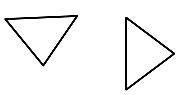




Cross-referencing of processes (to other flowcharts or decision tables) and files, inputs or outputs (to their specifications) is used by writing the cross-reference in a top-stripe of the symbol; for example:





Type of Chart / Symbol	System Flowchart Interactive System Flowchart and Clerical Procedure Flowchart	Computer Run Chart	Computer Procedure Flowchart
	All operations or procedure		
	All decisions		
	Storage media, permanent or temporary	Computer Backing Storage	Not used
	Documents, cards, paper tape, displays etc	Data passing between computer and non-computer parts of the system	Not used
	Connector, showing continuously between symbols where it is not possible to joint them by a flowline		
	Terminator, showing entry to or ……..		
	Data moving from one location to another	Not used	

Tape disk

Principles of flowcharting

It is essential lo keep the flowchart clear and easy-to-follow by:

- marking the start and end points;
- using standard symbols;
- a voiding crossed flowlines;
- using simple decisions (yes or no);
- working in a consistent direction.

The logic should be thoroughly checked to ensure that no actions are missed or wrongly repealed. The flowchart should show a consistent level of detail (by bearing in mind the use of the flowchart). Finally it is important to verify the completed flowchart by passing data through it and checking that the result is as expected (known as dry-running).

Levels of flowcharting

Flowcharts can be produced at different levels of detail. A System Flowchart which outlines a complete system is shown . The symbols are arranged in columns to indicate the function or department within which the activity or entry represented by the symbol occurs.

It is a general convention of flowcharting that flowlines go from left to right and from top to bottom; any exceptions to this convention are indicated by arrowheads on the line, such as the one entering immediately he-lore symbol 5. This is lo be distinguished from (he closed arrowhead, represented by symbols 13 and J.*>, which indicates physical movement of the medium containing the information. Note that this physical movement symbol could be used between symbols 5 and 6, but it would be redundant, since the movement is clearly signified by the flowline moving from one column to another.

Where a document containing a lower level of detail is available, relating lo one of the symbols on this systems flowchart, cross-reference is provided 10 that document in the top stripe of the symbol. The Clerical Procedure Flowchart is a lower-level representation of symbol 2 in the System Flowchart.This particular flowchart uses the columns to differentiate between the different documents and files used in the procedure. The same procedure could be drawn in a different way lo emphasise depart men I activities, or as a single main flowline to emphasise processes. The choice as to how the columns are lo be used is one to be made by the analyst, depending on which format brings out most clearly the essential factors.

DATA

The main source of information about data is the file of clerical documents. Each document needs to be identified and analysed by the systems analyst with the assistance of the user department. The analyst obviously needs to obtain copies of each document, and also requires other information about its usage. The Clerical Document Specification, provides a checklist of factors which the systems analyst should consider.

The specification should always be completed and filed with the actual document which it specifies. The main points which need to be recorded are the contents, organisation, size and usage patterns: each of these is covered in the Clerical Document Specification.

Occasionally the systems analyst will be investigating an existing computer-based system and in this case the data involved will be more complex: there will be a requirement to ensure that documentation of input, files, records and outputs is accurate. Suitable standard forms for recording this information are described later.

RELATIONSHIPS

In addition to procedures and data, the systems analyst needs to record information about relationships between various entities. The main types of documentation to support this activity are:

- organisation charts (which show formal relationships between people);
- physical layout charts (which show the flow of data between people within an office);
- grid charts (which can be used to show any type of relationship).

WEEK 10

OBJECTIVES:

The students are expected to learn the following:

- **Output types,**
- **Output definition,**
- **Data items,**
- **Principles and guidelines,**
- **Output media,**
- **Output specification,**
- **Choice of output medium**

OUTPUT DESIGN

Outputs from computer systems are required primarily to communicate the results of processing to users (or sometimes to other systems, including machine – based systems). They are also used to provide a permanent ('hard') copy of these results for later consultation. There are various types of output required by most systems; the main ones are:

- external outputs, whose destination is outside the organisation and which require special attention because they project the image of the organisation;
- internal outputs, whose destination is within the organisation and which require careful design because they are the user's main interface with the computer;
- operational outputs, whose use is purely within the computer department, eg program listings, usage statistics etc;
- interactive outputs, which involve the user in communicating directly with the computer (often called dialogues);
- turnround outputs, ie, re-entrant documents, to which data will be added before they are returned to the computer for further processing.

OUTPUT DEFINITION

The outputs may have been defined during the logical design stage. If not, they should be defined, at the beginning of the output design, in terms of:

- type of output;
- content (headings? numeric? alphanumeric? totals? etc);
- format (hard copy? screen? microfilm? etc);
- location (local? remote? transmitted? transported? etc);
- frequency (daily? weekly? hourly? etc);
- response (immediate? within a period? etc);
- volume (number of documents? growth? etc);
- sequence (account number? within sales area? etc);
- action required busting? Error correction? Etc)
- The content of the outputs must now be defined in detail.

Data items

The name given to each data item should be recorded and its characteristics described clearly in a standard form:

- whether alphabetic or numeric;
- legitimate and specific range of characters, eg minimum, maximum, fixed values or ranges;
- number of characters;
- position of decimal point, arithmetical sign or other indicator.

At those installations which maintain a register of the name and description of each data item used in the installation, many of these items may already have been registered. Any data item not yet defined must be identified and recorded before output design can proceed. The objective is to prevent the same data item being referred to by various names, or the same name being used to describe different items.

PRINCIPLES AND GUIDELINES

There is often a need at output to provide totals at various levels. These usually result from computer processing, and must therefore be contained on computer files. Their source must be identified, and they must be defined and registered as data items.

The systems analyst must know:

- what are the various total levels required, e.g. column totals, subtotals, grand totals;
- whether they should appear in a vertical column, offset or at the end of a horizontal line;
- what will cause them to occur, e.g. change of key, special markers or other conditions;
- how will they be identified as totals, e.g. implied or by special description;
- if there are brought forward or carried forward totals;
- what is their value and how should this be shown, e.g. numeric, money, weight, volume;
- what is their accuracy requirement, e.g. rounded to units of one, hundreds, thousands.

DATA EDITING

It is not always desirable to print or display data as it is held on a computer. The systems analyst must ensure whether the form in which it is stored in the computer (files) is suitable for the output, for example:

- will decimal points need to be inserted?
- will leading zeros need to be suppressed or are they required to be shown?
- where are money symbols to be shown, to the left of the first digit, in a fixed position or replacing each leading zero?

OUTPUT MEDIA

The next stage for the systems analyst is to determine the most appropriate medium for the outputs. This will involve consideration of a wide range of devices, including line printer, graph plotter, typewriter, visual display unit, magnetic media and microfilm. The choice of output medium will be affected by all kinds of considerations but the main ones will be:

- the suitability of the device to the particular application;
- the need for hard copy (and number of copies required);
- the response time required;

- the location of the users;
- the software/hardware available;
- the cost.

Once the medium has been chosen the output can be specified in detail appropriate to the device to be used.

OUTPUT SPECIFICATION

As the details are assimilated and the systems analyst reaches an understanding of the users' requirements, a mental picture of various alternative layouts will gradually emerge. Even if an outline layout has been provided at an earlier stage, this may be improved upon. Some output requirements offer less scope, e.g., pre-printed stationery, statutory documents, but their design must still be specified and documented; data items still need to be accurately defined and arranged for clarity and easy comprehension. The systems analyst has two specific objectives at this stage:

- to interpret and communicate the results of the computer part of a system to users in a form which they can understand and which meets their requirements;
- To communicate the output design specification to programmers in a way which is unambiguous, comprehensive and capable of being translated into programming language.

Each record in the overflow area is chained to the next highest key by a link field, giving the track and record number: the overflow from a given prime track always has its own chain within the overflow area.

Tagging

Chaining involves detecting a record by following the chain of addresses. Tagging follows a different principle, by leaving a remainder in the form of a tag for every record which has been displaced. The tag, which is written by a housekeeping routine, contains the record key and the address to which it has been written.

Tagging has the advantage that every record can be directly traced without the need to follow a chain. It has, however the following disadvantages:

- a data record may need to be displaced to make room for the tag, thus requiring a tag for itself.
- second-level overflow can be reached when:
- the 'home' page is filled with tags;
- overflow pages or all pages in a cylinder have been filled.

A further area at the end of the file has to be reserved to deal with second-level overflow. This consists of extension -pages specifically related to 'home* pages and chained by a pointer in the 'home' page. Thus the concept of chaining enters the tagging environment.

Choice of medium

Choice of the correct device for storing a file is most important with master files and most of this section is concerned with them, but careful consideration should also be given to media for other files. These are some of the considerations.

Purpose of the file

Direct access devices, such as disks, can be used to advantage as a means of holding transaction data or intermediate files; serial reading and processing from disk may be quicker than reading from magnetic tape. Although transaction and intermediate files on disk can be sorted quickly and conveniently, it is also possible to load transactions to specific areas of the disk, using, for example, data type as the criterion. Specific areas may then be accessed as required, thus removing the necessity for a sort.

Disks as well as tapes can serve as a convenient method of holding output data. This is particularly true, if a number of differently sequenced reports are required from different parts of the file. Random or selective sequential processing might be used for their production.

Program library files may be held on tape or disk. Frequently used programs are best held on disk, as they can be loaded without wasting time in searching through a tape.

Availability of hardware

Where there is no possibility of additional hardware for a proposed system, the choice of file media is restricted, particularly in small installations. It may not be possible to put the file on the best possible medium; the systems analyst will have less difficulty deciding which medium to use, but more difficulty in designing procedures which meet the needs of the user.

On the other hand, if major expenditure on hardware is envisaged, the systems analyst's choice may be more difficult, and new techniques may have to be developed. Not only must the optimum requirements of the system be considered, but also the requirements of other systems which may need to use the hardware.

Method of access

One of the first questions to be asked here is 'is direct access an essential requirement of this system, or is any future system likely to require this file?' If the answer is 'Yes', then choice will be restricted to the various direct access devices available. Consideration must be given here to present and future requirements for interrogation of the file and multi-access systems. If the proposed systems require real-time operation, or are on-line in other ways such as for time-sharing or enquiry, then direct access will be essential.

If the answer is 'No', the choice between serial then direct access will t/c made on a consideration of timing and cost. Even if direct access is not a prerequisite, the use of direct access media may be desirable if this will reduce sorting time, enable the file to be split into smaller files, or significantly reduce the processing time.

File activity

Other basic questions to be asked are:

- how often is the file required for reading or updating?
- will it be used by more than one program?
- If so, will the sequence of operation of these programs have any significance?
- what percentage of the file will be required during the run?
- how is the activity distributed through the file?

The frequently-used file is less likely to tie up hardware. Additionally, it will require less operator time if it is held on a direct-access device. If serial processing from magnetic tape seems to be indicated by the characteristics of the main run, the implications for subsidiary runs must also be considered.

File Size

Small files are usually best kept on disk, because they can be allocated sufficient space without affecting the rest of the disk. The same file on magnetic tape might use only the first 30 feet of the tape, so that the remainder cannot be used or can only be used for several other small files.

If a process requires many small files, then on magnetic tape:

- a large number of tape decks may be tied up;
- time can be wasted in searching for subfiles;
- time can be wasted in changing tape reels;
- files may need to be processed in sequence.

If the files could be stored on direct-access devices, they could possibly all be kept on one or two cylinders of a disk without incurring any hardware or time penalties.

Another possibility in the case of small files, is the use of slow media, such as cards or paper tape. This can be particularly useful when the file is only required at infrequent intervals, as cards and paper tape are cheaper as storage media, than disk or magnetic tape.

Large files are usually held on magnetic tape as it is a cheaper storage medium than disk. A large magnetic tape reel can hold more data than a 10-surface disk pack. Tape reels can be changed more quickly than disk packs when dealing with a multi-volume file. Furthermore, it is easier to allow a file to expand on magnetic tape, than on disk, particularly when the expansion involves an extra reel or cartridge. Other criteria may outweigh these considerations, however. It may be worthwhile to hold a large file with a low hit rate on disks.

A medium-sized file is considered to be one with the range of 4 to 40 million characters. This presents a rather more difficult choice: the particular hardware available may be the overriding factor.

INPUT REQUIREMENTS

Here it is important to choose carefully, not only the medium for the master file, but also the medium for input. For an input transaction file, requirements to be considered are:

- validation;
- control;
- sorting.

It may be necessary to separate different transaction types, to retain valid transactions whilst invalid transactions are corrected, to sort transactions to the same sequence as the master file. Thus the choice of medium for both transaction and master file may be interrelated.

There is also the case where the same input data, either in part or in total is required for two or more processes. If possible, the process of reading and validation of data from a slow peripheral should only be carried out once. The need for sorting this data several times must then be balanced against the possibilities of direct access.

Furthermore, one must consider whether to have a single transaction file or a number of files holding different types of transaction. In the second case, disks would be more attractive for holding the large number of small files.

WEEK 11

OBJECTIVES

The students are expected to learn the following:

- **Output design issues,**
- **Various types of output,**
- **Design various types of report,**
- **Output control,**
- **Report design principles**

Introduction

Output and user interface design is the first task in the systems design phase of the SDLC
v **Output design focuses on user needs for screen and printed forms of output, while user interface design stresses user interaction with the computer, including input design and procedures**

Output Design

Before designing output, ask yourself several questions:
– What is the purpose of the output?
– Who wants the information, why it is it needed, and how will it be used?
– What specific information will be included?
– Will the output be printed, viewed on -screen, or both? What type of device will the output go to?

Output Design

v **Types of Output**
– In the systems design phase, you will create the actual forms, reports, documents, and other types of output
– Internet -based information delivery
– E -mail
– Instant Messaging
– Wireless Devices

Output Design

Types of Output

– Digital audio, images, and video
– Podcasts
– Automated facsimile systems
• Faxback systems
– Computer output microfilm (COM)
– Computer output to digital media

Output Design

Types of Output

– Specialized Forms of Output

• An incredibly diverse marketplace requires a variety of specialized output

– Output from one system often becomes input into another system

– Although digital technology has opened new horizons in business communications, printed output still is the most common type of output, and specific considerations apply to it

Printed and Screen Output

Reports

– Detail reports

EMPLOYEE HOURS
WEEK ENDING DATE: 6/22/07 PAGE 1

STORE NUMBER	EMPLOYEE NAME	POSITION	REGULAR HOURS	OVERTIME HOURS	TOTAL HOURS
8	Andres, Marguerite	Clerk	20.0	0.0	20.0
8	Bogema, Michelle	Clerk	12.5	0.0	12.5
8	Davenport, Kim	Asst Mgr	40.0	5.0	45.0
8	Lemka, Susan	Clerk	32.7	0.0	32.7
8	Linquist, Linda	Clerk	16.0	0.0	16.0
8	Ramirez, Rudy	Manager	40.0	8.5	48.5
8	Ullery, Ruth	Clerk	20.0	0.0	20.0
		STORE 8 TOTALS:	181.2	13.5	194.7
11	Byrum, Cher	Clerk	15.0	0.0	15.0
11	Byrum, Mary	Clerk	15.0	0.0	15.0
11	Deal, JoAnn	Clerk	4.8	0.0	4.8
11	Galzinski, Barbara	Manager	40.0	10.0	50.0
11	Huynh, Loc	Clerk	20.0	0.0	20.0
11	Schuller, Monica	Clerk	10.0	0.0	10.0
11	Stiles, Carol	Clerk	40.0	12.0	52.0
11	Thompson, Mary Kay	Asst Mgr	40.0	1.5	41.5
		STORE 11 TOTALS:	184.8	23.5	208.3
17	De Martini, Jennifer	Clerk	40.0	8.4	48.4
17	Haff, Lisa	Manager	40.0	0.0	40.0
17	Rittenbery, Sandra	Clerk	40.0	11.0	51.0
17	Wyer, Elizabeth	Clerk	20.0	0.0	20.0
17	Zeigler, Cecille	Clerk	32.0	0.0	32.0
		STORE 17 TOTALS:	172.0	19.4	191.4
		GRAND TOTALS:	538.0	56.4	594.4

Printed and Screen Output

Reports

– Exception reports

• Are useful when the user wants information only on records that might require action

– Summary reports

• Reports used by individuals at higher levels in the organization include less detail than reports used by lower -level employees

Printed and Screen Output

User Involvement in Report Design

– Printed reports are an important way of delivering information to users, so recipients should approve all report designs in advance

– To avoid problems submit each design for approval as you complete it, rather than waiting until you finish all report designs

– Mock -up

Printed and Screen Output

Report Design Principles

– Printed reports must be attractive, professional, and easy to read

– Report headers and footers

– Page headers and footers

– Column heading alignment

– Column spacing

Printed and Screen Output

Report Design Principles

– Field order

• Fields should be displayed and grouped in a logical order

– Grouping detail lines

• It is meaningful to arrange detail lines in groups

• Group header

• Group footer

Printed and Screen Output

Report Design Issues

– Good design standards produce reports that are uniform and consistent

– When a system produces multiple reports, each report should share common design elements

– After a report design is approved, you should document the design in a report analysis form

CALCULATING THE LENGTH OF THE REPORT

	66	total lines available per page of stock paper
-	6	lines reserved for top and bottom margins
-	6	lines per page for 2 page heading lines, 1 blank line, 2 column heading lines, and 1 blank line
	54	available detail lines per page
	2,100,000	detail lines
÷	54	detail lines per page
	38,889	**printed pages**

CALCULATING THE TIME TO PRINT THE REPORT

LINE PRINTER AT 2,000 LINES PER MINUTE

	4	printed heading lines per page
×	38,889	pages
	155,556	printed heading lines
+	2,100,000	printed detail lines
	2,255,556	printed lines
÷	2,000	lines printed per minute

1,128 minutes = 18.8 hours of printer usage

LASER PRINTER AT 16 PPM

	38,889	pages
÷	16	pages per minute

2,431 minutes = 40.5 hours of printer usage

Printed and Screen Output

v **Output Control and Security**

– Output must be accurate, complete, current, and secure

– The IT department is responsible for output control and security measures

– Many companies have installed diskless Workstations.

OUTPUT DESIGN CONSIDERATIONS

The choice of the most suitable data organisation for a particular group of applications is important, perhaps even crucial, to the eventual successful performance of systems which use the data. Ideally, the range for choice of storage medium and data access method should be unlimited, but this is rarely the case. The file designer is constrained by various factors, including the requirement to interface with existing applications, their files' existing data storage media, and the range of options supported by the incumbent manufacturer's hardware and software. It is, however, possible to draw up a series of alternatives in designing files, and to evaluate those alternatives against given criteria. The first consideration is the choice of data storage device to be used.

OUTPUT REQUIREMENTS

If any one output report requires information from every record on the file, then this should be produced by serial processing. But if a report requires information only from certain parts of the file, it may well be better to hold the file on a direct-access device.

Where multiple reports are required from the same file, they may be produced either by direct access for one report at a time or simultaneously by serial processing techniques. Often some further processing may be required between extracting the information from the file and printing it; for example, sorting or conversion of data from one format to another, the need to transmit it over a data link.

WEEK 12

OBJECTIVES:

The students are expected to learn the following:

- **Tasks of implementation,**
- **Planning and control,**
- **Implementation coordinating committee,**
- **Education and training,**
- **System testing and**
- **System test data.**

SYSTEM IMPLEMENTATION

implementation is the stage of the project when the theoretical design is turned into a working system. At this stage the main workload, the greatest upheaval and the major impact on existing practices shifts to the user department. If the implementation stage is not carefully planned and controlled, it can cause chaos. Thus it can be considered to be the most crucial stage in achieving a successful new system and in giving the users confidence that the new system will work and be effective.

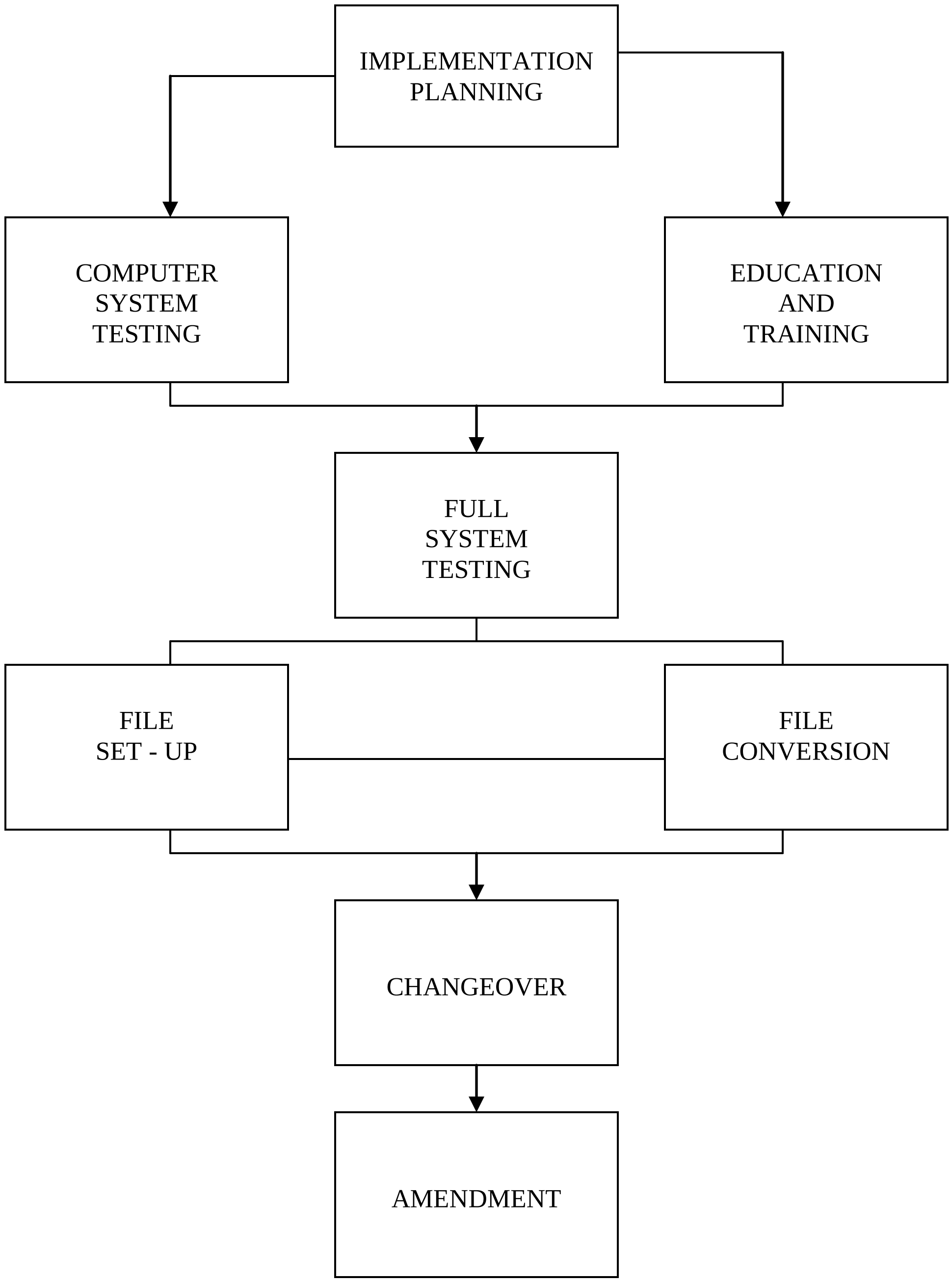
The implementation stage is a systems project in its own right. It involves ireful planning, investigation of the current system and its constraints on implementation, design of methods to achieve the changeover (including .Tiling programs to convert files), training of staff in the changeover procedures (as well as in the new system procedures), and evaluation of changeover methods.

The more complex the system being implemented, the more involved will be the systems analysis and design effort required just for implementation, Indeed in large organisations systems analysts may specialise in implementation system activities.

TASKS OF IMPLEMENTATION

The tasks involved in the normal implementation process. The first task is implementation planning, i.e. deciding on the methods and timescale to be adopted.

Once the planning has been completed, the major effort in the computer department is to ensure that the programs in the system are working properly. At the same time the user department must concentrate on training user staff. When the staff have been trained a full system test can be carried out, involving both computer and clerical procedures. Planning, training and testing, considered in this chapter, are the preparatory activities of implementation.



Once the preparation is complete, the implementation proper can begin. The first part of this involves the conversion of existing clerical files to computer media and the setting up of these files as they are converted on the computer. Then the actual changeover from the existing system to the new system takes place. These aspects are considered in the next chapter.

Finally, when the changeover has taken place, there will be need for amendments to correct or improve the new system, and this aspect is considered as part of Chapter 21.

PLANNING AND CONTROL

The implementation of a system involves people from different departments, and systems analysts are confronted with the practical problem of controlling the activities of people outside their own data processing department. Prior to this point in the project, systems analysts have interviewed department staff with the permission of their respective managers, but the implementation phase involves the staff of user departments carrying out specific tasks which require supervision and control to critical schedules. Some of these people may also need to be employed on the tasks of implementation in addition to their own present departmental tasks.

Who, then, should be responsible for a successful implementation? Ineffective control of implementation can result in the failure of a system which is otherwise excellent. Systems analysts do not have 'line' authority and even if they had, it would probably arouse resentment and resistance in both departmental managers and their staff at a time of critical importance. The systems analyst provides a service to management and cannot take executive responsibility without prejudicing this advisory role. Many of the reasons for not having a systems analyst in control apply equally to the manager of the data processing department. Vesting control directly in the line managers of user departments raises problems because of their lack of data processing knowledge.

A practical compromise is to have control vested in the line managers through an implementation coordinating committee with an advisory role, leaving systems analysts free to carry out their proper function. Where a project steering committee is already in existence, a subcommittee may be deputed to carry out this function.

THE IMPLEMENTATION CO-ORDINATING COMMITTEE

The use of a committee is a mixed blessing. It allows all aspects of problems to be considered and resolved by the appropriate line managers, particularly where the new system cuts across old or new departmental boundaries. However, it can also make decision-making a lengthy process. As with all committees, its success depends upon experienced chairmanship.

The composition of the committee is important. Usually the best person to chair the meetings is the line manager whose department is most affected by the system. There should be at least one representative of each department affected by the changes, and other members should be co-opted for discussions of specific topics. The data processing manager and the systems analyst or project leader responsible for the development of the system should be members. A line manager may be seconded to work directly with the systems analyst during implementation, and either is a suitable choice for the post of secretary. The frequency of the meetings will be influenced to a large extent by the system analyst's plans and schedules.

The committee itself will not normally exercise any executive authority; this belongs to the individual line managers who will however, be influenced by conclusions reached in committee. The committee will set up a realistic timescale for implementation and monitor progress. In the event of any executive problems which cannot be resolved, the committee would be expected to report, through the chairman, to a higher executive authority, such as the computer development steering committee.

The committee will be a sounding-board for ideas, complaints and problems in the user department, but it should also actively consider:

- the implications of the system environment;
- staff selection and allocation for implementation tasks;
- consultation with unions;
- resources available (staff and equipment);
- standby facilities;

- channels of communication;
- alternative methods of achieving changeover.

The appointment of an implementation coordinating committee depends on policy at individual organisations. In the absence of such a committee, the problems of responsibility discussed here are still relevant, and an awareness of these can assist systems analysts in planning their approach. The active cooperation of the respective managers or their representatives as discussed for the formal committee, preferably working as a team, is still required for successful implementation.

It is not suggested that the implementation process must wait until all other activities have been completed. Throughout all phases of the project, the systems analyst will be planning the activities of implementation.

Some of these activities must begin earlier than others because they take longer to complete. Some depend on the completion of others before they can be started. They also involve the use of resources and in some cases may stretch over a considerable length of time. Their coordination and control require careful planning and the establishment of schedules.

For successful implementation, systems analysts should formalise their requirements by preparing an implementation plan using the techniques described in the planning section of Chapter 23. In some organisations the implementation plan is a mandatory procedure laid down by management. A network diagram is useful to show the relationship of activities in their sequence in the overall plan, while bar charts can be used to schedule the lower level tasks for each activity on an individual basis.

With or without the formal establishment of responsibility for implementation, systems analysts will be responsible for scheduling the tasks to be done and monitoring progress. This will test their qualities of persuasion and tact.

EDUCATION AND TRAINING

To achieve the objectives and benefits expected from computer-based systems it is essential for the people who will be involved to be confident of their role in the new system. This involves them in understanding the overall system and its effect on the organisation, and in being able to carry out effectively their specific tasks. As systems become more complex the need for education and training is more and more important.

Staff selection

The implementation of a new system involves people; successful implementation depends upon the right people being at the right place at the right time. Planning networks and charts can show the number and type of people required at the place and the time, but successful implementation requires staff selection and training for that part of the system for which the staff will be responsible. Trained personnel will be needed as soon as the implementation activities begin, so training must begin before that stage.

Selection for training must take place at an early stage. Broad estimates of the numbers and types of people required should be submitted to management for approval. Then, as development proceeds, these estimates will be progressively reviewed and improved. The managers of the departments concerned, both user and operations, define the type and quality of people required for the implementation and operation of the new system, with advice from the project team. In large organisations, the personnel manager will naturally also become involved.

Like most other aspects of systems analysis and design, decisions on staff selection and training are not all made instantly, but are progressively refined as more information about the system requirements becomes available. Some requirements may be known early in the project, but there comes a time when all these requirements need to be firm.
For implementation to be successful, consideration of people who will be affected should be a high priority of the coordinating committee and the systems analyst. They should be told at the earliest possible time why the changes are necessary and how they will be

affected. It is important that this communication takes place through the proper channels, i.e. their respective departmental managers, and not systems analysts or a committee.

It is usually advisable, and in some cases essential, for the personnel manager and trade union officials to be consulted about redeployment and especially about any redundancies. This needs to be done through the proper channels and as early as possible. It is a management responsibility, although the systems analyst may be involved as an adviser.

Systems analysts should ensure that staff selection and training schedules take account of any constraints. These may relate to availability of appropriate staff, lecture rooms, and equipment.

Training

Training requirements are easy to determine. They arise directly from the changes which the systems analyst is bringing about. User managers must be informed of how the whole system works, its objectives, new documentation, files and procedures. New jobs may be created, existing jobs changed or eliminated. User staff must be instructed in how to perform their new tasks. These requirements will be set out in principle in the User System Specification and in detail in the User Manual described in Chapter 22. Thus, the new system can be explained to user management and new tasks specified in job specifications, with the systems analyst on hand to answer any queries which may arise. But this can be arid material and to rely on manuals alone for training is to court disaster.

Training sessions must aim to give user staff the specific skills required in their new jobs. Thus they should contain large elements of practical activity (people learn far more by doing than by watching or listening). Ideally they should consist of short and regular sessions rather than a long, once-off period, and they should be conducted with small groups rather than the full number of staff; this facilitates learning from peers.

The training will be most successful if conducted by the supervisor with the systems analyst in attendance to sort out any queries; new methods will gain acceptance more quickly in this way. This also applies to user manuals; if the users write their own manuals they will be more intelligible and more acceptable.

Plenty of job-aids should be provided to assist in the learning process: these will include visual presentation of procedures (e.g. in flowchart form), notices on machines, wall-charts (e.g. of timescales), and use of different colours (e.g. on forms or switches). Above all, the training sessions should be tailored to the learning process, allowing assimilation over a period before more new information is fed in; and providing feedback on progress.

Education

Education is complementary to training. It brings life to formal training by explaining the background to the changes and the reasons for them. It helps to overcome the resentment that may be caused by the computer seeming to take away responsibility from individual departments.

Systems are now developed by specialists from outside the department. 'First-time' users in particular need to appreciate the contribution of computers, how they work, and how they can assist clerical functions and decision-making.

Large numbers of clerical staff have been content with their jobs. They understand the work and their responsibilities, and do not welcome changes associated with computers. To them, computers have a reputation for creating unemployment and taking the skill out of jobs. There is little evidence that computers cause unemployment, though they do cause redeployment.

Staff may be reassured when shown how computers can remove the drudgery from tasks and can cause jobs to be upgraded. If a system analyst's design produces less-skilled jobs for clerical staff, they should ask why these cannot be performed by the computer. Education should reassure user staff by placing their problem in perspective, by dispelling myths and mystique, and by indicating the contribution that computers can make to organisation objectives and to job satisfaction.

Education involves creating the right atmosphere and motivating user-staff. Education sessions should encourage participation from all staff, with protection for individuals from group criticism. The sessions should make creative use of worries about the public image of the computer, perceptions of the systems analyst's role, rumours about previous implementations. Etc; and they should be vehicles for the natural evolution of ideas about the system and its justification.

Education should start well before any development work to enable users to maintain, or to regain, the ability to participate in the development, of 'their' system. A spirit of cooperation will be extremely valuable to the systems analyst, and will go a long way toward the successful implementation and operation of the new system. The analyst is continually in an educating role. In each meeting with users, they should be encouraged to build up their understanding of the new system. Nothing can help the used more in learning about the system than participation in its design.

Educational information can also make training more interesting and understandable. Instructions to perform new tasks should not be taught parrot fashion'; they should be explained within the context of the system. or the part appropriate to the task. This wider understanding of their tasks will assist user staff to deal correctly with the unusual events which will inevitably arise. The aim should always be to make individuals fee! that they can still make an important contribution, to explain how they can participate in making system changes, and to show that the computer and computer staff do not operate in isolation, but are part of the same organisation. Computers produce information on which managers base their decisions: all staff" influence the quality of this information by the quality of their own individual contributions within the computer-based systems.

Education can be planned for the whole organisation, separately from training for individual projects. Computer education should be included in the organisation's normal training programme for management and staff. This should start with the involvement of directors and senior managers and be presented by specialists from outside or within the organisation. From within, it is likely to be arranged by data processing management or training specialists who understand the requirements of top management.

Once education is established at the senior level it is likely to be more effective with tower levels of management, supervisors, and clerical and shop-floor staff. Data processing staff also need to be educated and trained, and should participate in such a company-wide programme.

The analyst's responsibilities

The systems analyst has a clear responsibility to press for full and adequate education and training for user staff and to ensure that provision is made for this in project budgets. This does not mean that the analyst should necessarily organise and run the training, but be advisory to user managers, providing appropriate technical expertise when called upon.

If the analyst does organise the training sessions special attention must be paid to the user's perception of the change: courses should reflect the user's view of the world rather than the analyst's view.

Finally 21 effort should be made to measure the success of training objectively and to learn from failures. If a member of staff is unable to cope with the new system, it will usually be the system, or the job design, or the training programme which is at fault, and rarely the individual.

SYSTEM TESTING

System testing is the stage of implementation which is aimed at ensuring that the system works accurately and efficiently before live operation commences. In principle, system-proving is an on-going activity throughout the project. The logical design and the physical design should be thoroughly and continually examined on paper to ensure that they will work when implemented. Thus the system test in implementation should be a confirmation that all is correct and an opportunity to show the users that the system works.

When the programmers have tested each program individually, using test data designed by themselves, and have verified that these programs link together in the way specified in the computer run chart to produce the output specified in the program suite specification, the complete system and its environment must be tested to the satisfaction of the systems analyst and the user.

The systems analyst will provide the test data, specially designed to show that the system will operate successfully in all its aspects and produce expected results under expected conditions. The tests should take place as far as possible in the actual operating environment, and they should test people and equipment as well as programs. Where this is not possible, the system should be tested in a simulated operational environment to prove that the computer and clerical procedures are understood and produce the required results. Sometimes it is convenient to use live data from a previous system cycle, but this presupposes that the new files have been set up and can be used. Preparation of test data and the checking of results should be carried out in conjunction with the appropriate user and operations departments.

SYSTEM TEST DATA

There should be careful planning of how the system will be proved and the test data designed. The systems analyst should be quite clear about the test objectives. System test data can rarely be comprehensive enough to test the system fully: some aspects of the system will have to be tested during live operation.

Usually the test data produced is sufficient to test most of the system, whereas to test it fully would be increasingly time-consuming and difficult. This is sometimes referred to as the 'Pareto' or 20 SO effect, where 20% effort can produce 80% of the results.

Systems analysts should deliberately plan the extent to which the system should be tested. This will depend on the purpose and sensitivity of a system. Many statistical results, e.g. sales analyses, are not so sensitive and do not demand the accuracy needed for. say, a labour force piecework bonus, customer accounting, or product-making formulae. These factors should already have been identified and taken into account in planning the detailed design. To take an extreme case, where the system is a 'life or death[1] type, the extra effort and time required for the production of the most comprehensive system test data would be essential.

The system test data, and the results of processing it. should be maintained as a permanent manual throughout the operational life of the system, for audit purposes or to test any subsequent major amendments.

A suggested checklist of the contents of this manual appears in WEEK 8&9 shows an example of how decision tables can greatly assist in the formulation of system test data.

WEEK 13

OBJECTIVES:

The students are expected to learn the following:

- **File conversion,**
- **File set up,**
- **Error detection and control,**
- **Handover,**
- **Basic of organization,**
- **Data control staff in a DP department and**
- **Data preparation staff**

SYSTEM INSTALLATION

Introduction

Once all the preparatory work of implementation has taken place – the system has been tested and the staff trained - the changeover from the old to the new system can begin. This involves three separate, though closely linked, activities. First of all, the old (existing) system files need to be converted to the format and content required by the new system. Then the converted files need to be set up on the computer. Finally the old procedures have to be replaced by the new ones.

FILE CONVERSION

File conversion is a vital activity which is sometimes underestimated. It involves the conversion system of the old file data into the form required by the new system, and is usually a very expensive stage in the whole project. It is rare that the old files can be converted as they stand because of format changes and the need to collect and assemble data from a variety of sources. But the major problem of file conversion is that it usually has to be achieved within very tight time constraints.

Although it is usually regarded as a part of changeover, in fact file con\ersion is often a complete and separate system task in itself, involving fact-finding, analysis, data capture, the design of clerical methods and computer processes, form design, and the production of special training courses.

Setting up new master files for large systems can involve the transfer of tens or hundreds of thousands of records, which may be beyond the data handling capacity of an organisation and must be subcontracted elsewhere. If the task is to be carried out within the organisation, it is essential that adequate advance notice be given. Either of these methods requires detailed planning and monitoring of progress to clearly defined schedules.

The conversion of data on clerical documents invariably demands punching and verifying. It may also be necessary for the data to be first assembled and then written onto specially designed conversion punching documents.

A common problem is that the data contained on the source documents needs to be edited, either because it is irrelevant to the new computer files, or because it needs to be expressed in a different format. This requires mined clerical staff, who often have to be released from the user departments to perform these duties or to supervise temporarily recruited people, if the data is of a special technical nature.

The conversion of data held on punched or magnetic media raises different problems. Again, the data is likely to need editing and restructuring. for which a special editing computer program will be needed. If the new system has been designed to operate on a new computer, there may also be a problem of incompatibility between the coded data on the old and new magnetic media. This should have been anticipated and arrangements made with the computer supplier, but it must be taken into account when planning and scheduling file conversion.

Live Files

Whatever methods of conversion are used, it must be remembered that usually they will involve the conversion of live files (e.g. stock files, ledger files). This poses major organisational and scheduling difficulties, since incoming data (e.g. stock issues, ledger payments) is continually being used to update the files. The difficulties will vary according to:

- the number of records and amount of data to be handled;
- the frequency with which the data changes.

These two factors are interrelated. Large and dynamic files increase the difficulty of capturing changing data. Small files may be released and the clerical updating process halted during the conversion process. Even in this relatively simple case, the timing is critical. Ideally, the actual conversion and setting up new files should be done at the last possible moment, so that the time and the amount of data to be charged between the chart of conversion and the changeover to full computer operation is kept to a minimum.

With large files this may not be possible, the files being available only in a piecemeal fashion. In this case, a method must be devised whereby control can be maintained over a carefully planned release of the data, either in parts or sections.

A record must be kept of data received during conversion and the changeover, but which has not yet been used to update the files. This record is then used to ensure that all these data changes are eventually made to the new system files, and also to the old clerical files according to the changeover method. This calls for critical control, both for scheduling and the control of accuracy.

The conversion of large files may be assisted by first separating and converting the static data part of each record on the files, converting the dynamic contents as late as possible, and then merging the two parts of each record to make up complete records and complete new files.

Added problems occur when files have to be brought together from several locations for a centralised conversion procedure. Not only is the control problem magnified, but often the file records from the different locations are in different formats. In this case, the old files will require further processing before the conversion activity can take place.

FILE SET-UP

‘File set-up’ is the process of creating the new computer file from the converted, computer-acceptable data. Sometimes the file amendment programs, written for the new system can be simply modified to accept the converted data; but this approach assumes that the converted data is in exactly the same form as a new record in the new system. Usually, special programs are required to carry out some 'once only' conversion processes.

The major problems associated with file set-up are the accuracy of the conversion and the error detection and correction procedures. It is essential that, at the end of file set-up, the users are satisfied with the new files. This is a testing time for the new system, because if the new files contain many errors, the users, rightly or wrongly, will claim that the old files were always absolutely accurate. If users lack confidence in the new files, they will soon express a desire to be back on the old system.

Accuracy

It is vital that the data content of master files at changeover is accurate. Otherwise, not only will errors have to be identified and corrected during the operation of the new system, but it may also inconvenience users and cause them to suspect the design of the new system. Incorrect data may arise:

- as errors in the original source documents;
- during clerical transcription;
- during punching;
- from a conversion program.

Other errors can arise if records are lost during their removal for conversion or if some data in certain records are found to be non-existent (e.g. carried in the heads or in the personal records of the clerical staff).

ERROR DETECTION AND CORRECTION

Steps can be taken to detect errors either by computer or by clerical checking of printed files. For large files, a special computer program should be written to check the accuracy of the data. However, there are limits to the extent to which values can be checked in this way, and wherever possible, clerical checks of the printed new files should be made against either the old files or the transcription documents. The control of data, discussed above, will need to be extended to cover the correction of data, using both computer and clerical controls of total value, numeric count, hash totals, and check digits (preferably in identifiable small batches).

This is not an easy operation, but the results make it worthwhile. The production of new files of data which are demonstrably more accurate than the original files would make a good start for the new system, particularly in terms of user confidence.

CHANGEOVER

The changeover from the old to the new system may take place when:

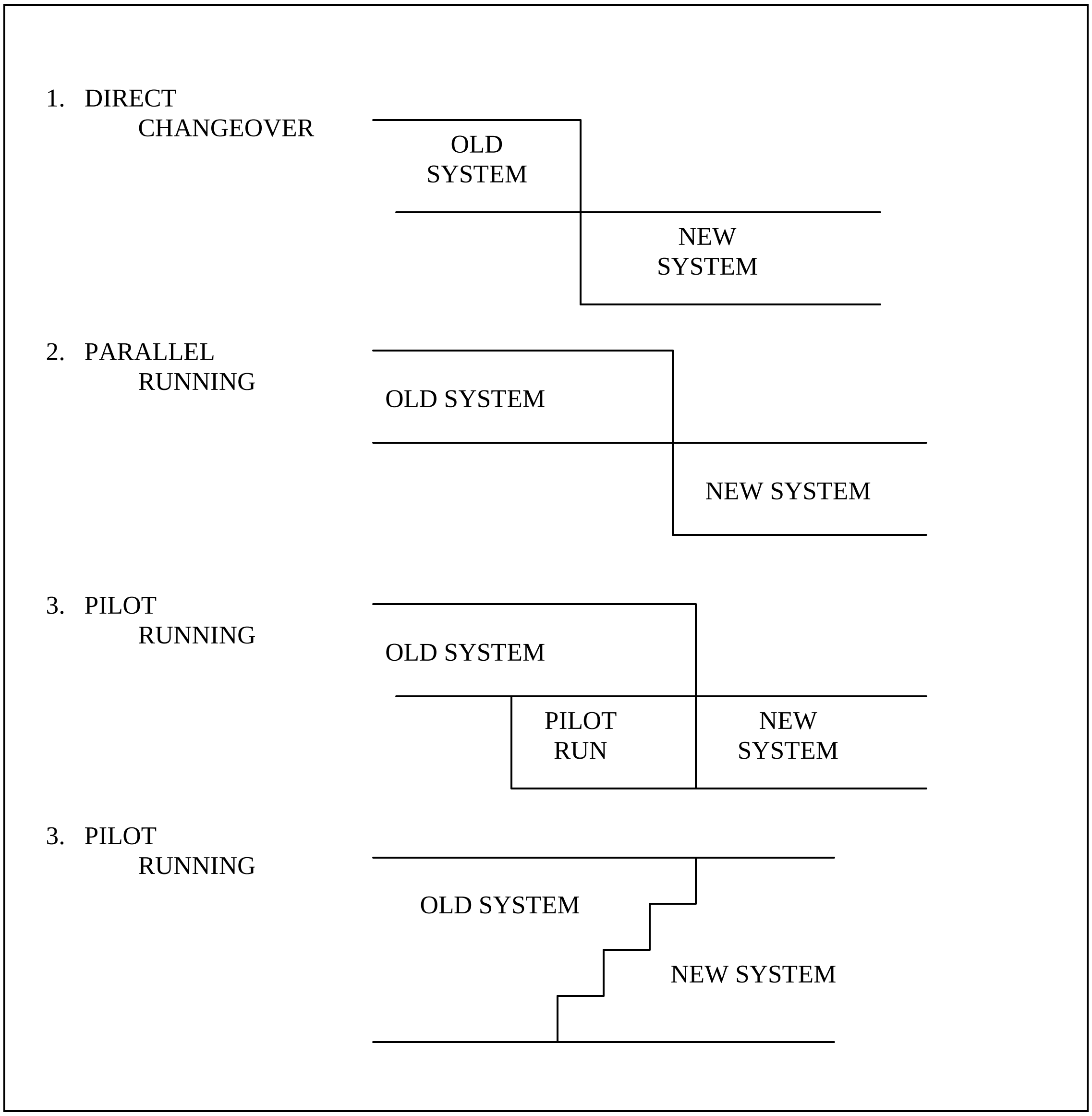
- the system has been proved to the satisfaction of the systems analyst and the other implementation activities have been completed;
- user managers are satisfied with the results of the system tests, staff training and reference manuals;
- the operations manager is satisfied with the performance of equipment, operations staff and the timetable;
- the target date for changeover is due.

The changeover may be achieved in a number of ways. The most common methods are direct changeover, parallel running, pilot running, and staged changeover. They are illustrated diagrammatically (fig. 20.f). Occasionally a combination of these methods will be used.

Direct changeover

As this term implies, this method is the complete replacement of the old system by the new, in one move. It is a bold move, which should be undertaken only when everyone concerned has confidence in the new system. This presupposes a well organised and supervised implementation.

When a direct changeover is planned, system tests and training should be comprehensive, and the changeover itself planned in detail. This method is potentially the least expensive but the most risky. Where possible, a time should be chosen when the work of the organisation is slack. In any event, the busiest time should be avoided.



When planning a direct changeover, it may be decided that the irrevocable disbandment of the old system before the new system has successfully completed its first full cycle is too risky. This would be a reasonable consideration when using new equipment, possibly untried under operational conditions. For security reasons, the old system may be held in abeyance, including people and equipment. In the event of a major failure of the new system the organisation would revert to the old system. Direct changeover is likely to be used:

- for a system which is new to the organisation, as distinct from a replacement system;
- where the new system incorporates such major innovations that a sensible comparison is not possible in parallel operation;
- where the user departments are experienced in computer systems;
- for a relatively small changeover or a short timescale.

It is important that written changeover instructions are prepared by the sv stems analyst to inform both the user and operations departments of the methods by which the changeover is to be effected and of the work to be done. These may take the same form as the User and Operations Manuals including any documents, procedures, programs and schedules specific to the changeover.

Parallel running

Parallel running, or operation, means processing current data by both the old and new systems to cross-check the results.

Its main attraction is that the old system is kept alive and operational until the new system has been proved for at least one system cycle, using full the data in the real operational environment of place, people, equipment and time. It allows the results of the new system to be compared with the old system before acceptance by the user, thereby promoting user confidence.

Where the new system incorporates major changes and enhancements in procedures, the use of new equipment, or in the results, the two systems may not be strictly comparable. Parallel running in this case, would ensure that the user departments could still carry on with the old system in the event of a major failure of the new system.

Its main disadvantage is the extra cost, the difficulty and (sometimes) the impracticability, of user staff having the different clerical operations for two systems (old and new; in me time available for one: and then cross-checking the results, including error-handling, at a time when user staff are fully occupied with the new procedures. The use of additional temporary staff may be neither possible nor desirable. Because of these difficulties, there is a danger of neither system being properly conducted, which defeats the objective of this method of changeover.

It is not suggested that this method does not produce an effective changeover, but that the implications should first be evaluated and the operation carefully planned in detail. As it places a heavy load on the users concerned, this should be fully explained to them beforehand.

The plan should be specific about the extent of cross-checking involved, eg whether on a sample basis, the criteria on which acceptance is to be judged, and the limit on the number of cycles of operating the two systems. Errors found by the comparison are as often due to defects in the old as in the new system, but it may prove difficult to convince users of this.

Where this method of changeover has been agreed, it must not be used as an excuse to curtail proper system testing. Parallel operation does not allow much time for learning or testing activities which should have been done earlier. It can, however, in a well organised implementation, build up users' confidence as they see that the system works and that they can do their job within it.

Pilot running

Pilot running is similar in concept to parallel running. Data from one or more previous periods for the whole or part of the system is run on the new system after results have been obtained from the old system, and the new results are compared with the old. It is not as disruptive as parallel operation, since timing is less critical. However, users still have to cope with the clerical procedures for both the old and new systems. It does not simulate day-to-day timing and scheduling problems, and data capture and error handling are not realistic. This method is more like an extended system test, but it may be considered a more practical form of changeover for organisational reasons.

Staged changeover

A staged changeover involves a series of limited-size direct changeovers, the new system being introduced piece-by-piece. A complete part, or logical section, is committed to the new system while the remaining parts or sections are processed by the old system. Only when the selected part is operating satisfactorily is the remainder transferred.

This method reduces, the risks inherent in a direct changeover of the whole system and enables the analyst and users to learn from mistakes made as the changeover progresses, its disadvantages are that it creates problems of conducting the selected parts of the old and new systems and it tends to prolong the implementation period. It therefore needs good coordination and is most appropriate when the changeover has to take place at a number of different locations, each location being subjected to a direct changeover, or for very large files.

Controls

Whichever method is adopted for the changeover from an old to a new method, a high priority must be given to establishing controls, by value or quantity, in order to maintain the quantitative integrity of the system. Users should keep overall control records incorporating both computer and clerical control figures to prove that the changeover has not corrupted this integrity. This is particularly important for financial systems, the controls for which, including audit trials, should have been planned after prior consultation with the accountants and auditors.

These controls are vital, and the tasks required to establish them can be difficult and time consuming. For large and complex systems, the most difficult task is often proving and establishing accurate controls for the old system before the changeover. This difficulty may be an existing problem, and one of the major reasons for the new system: the terms of reference should specify the extent of the systems analyst's involvement with the controls of the old system. When the accuracy of the old controls have been established, they must then be organised in a way that is consistent with the arrangement of the controls of the new system.

This aspect of controls at changeover cannot be overemphasised since not all existing systems or their control methods are in a good state of order.

HAND-OVER

Once the system has been working for an agreed period of time, the systems analyst will wish to withdraw. Prolonged involvement of the systems analyst with a working system should be avoided. This does not mean that the user department will cease to receive support. The system becomes the responsibility of a maintenance group within the computer department instead of the development staff. This hand-over point should be established as part of the implementation plan.

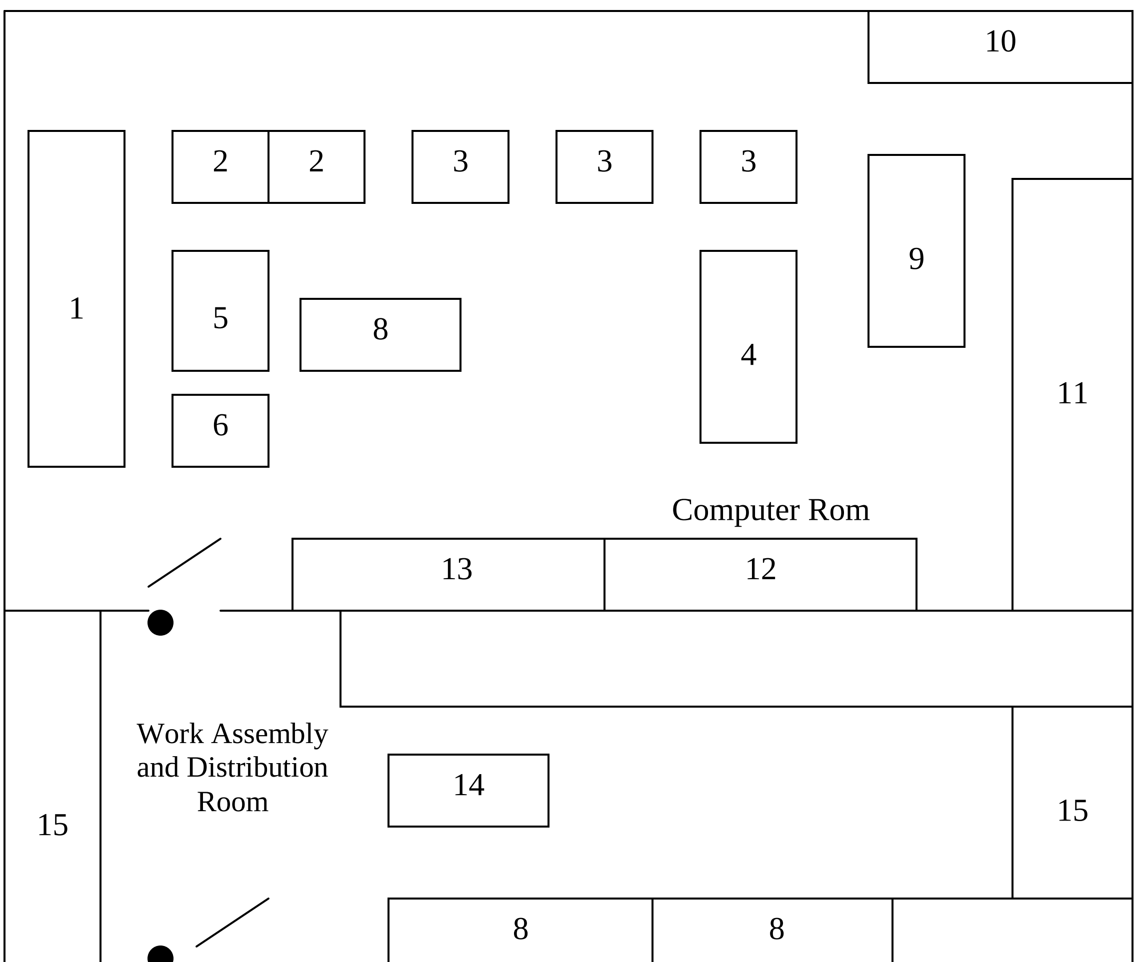
The users must be satisfied that the system works properly and meets all their requirements by the time hand-over takes place. It is essential, therefore, that the hand-

BASIC OF ORGANIZATION

over takes place formally, with a clear understanding on all sides that the systems analyst's involvement has come to an end.

The development and operation of a computer system requires careful organization of the individuals involved and those individuals need to be suitably trained in the correct practices to follow.

This diagram shows the layout of a typical computer room. There is a separate room for the assembly of incoming work which contains a number of storage cabinets for stationery, etc, and a guillotine/decollator for handling output stationery.



KEY

1.	Processor	9.	Librarian's Desk
2.	Magnetic Tape Drives	10.	Magnetic Disk Cabinet
3.	Magnetic Disk Drives	11.	Magnetic Tape Racks
4.	Line Printer	12.	Diskette Cabinets
5.	Diskette Drives	13.	Operators' and Engineers' Manuals
6.	PABX	14.	Guillotine/Decollators
7.	Console	15.	Stationery Cupboards
8.	Work Tables.		

The nature of several types of specialist work has been covered previously and do not merit repeating here:

(a) Systems analysis.

(b) Programming Operating.

Details of the work of other, specialist stall' are given in the following paragraphs.

The IT (DP manager In many commercial organisations).

The IT manager (sometimes called the computer manager) is a key figure in the - organisation. His or her job is to ensure that the computer department functions efficiently in the service of the company. He or she is responsible for ensuring that the computer needs of the organisation are met within the policy guidelines laid down.

The IT manager must be a good administrator as well as having a sound business knowledge. He or she must also have the knowledge and expertise necessary to enable him or her to control his or her teams of specialists in the various computing fields.

Status: It is important that his or her status be clearly defined especially with regard to his right of access to the board.

Ideally one would like someone who has previous experience of installing successful systems, and (lie appointment should be made very early on in the planning cycle to enable the company to gel the benefit of his or her specialist knowledge and experience.

If not recruited from outside, [he potential computer manager could be found in the management services division.

DATA-CONTROL STAFF IN A DP DEPARTMENT

These staff are responsible for the co-ordination of all machine-processing operations and for ensuring a smooth (low of work through the operations department. In order in explain their work we will trace a particular job through from beginning to end.

(a) Source documents accompanied by control totals are received from the clerical function (outside the department) and vetted visually.

(b) Control totals are agreed and documents passed under this control to the data preparation section.

(c) The data-preparation section prepares floppy disks or other media for source documents.

(d) Source documents are returned to the clerical function and the floppy disks now represent the input for a particular run (job), e.g. "invoice run".

(e) The job-assembly section "make up" the run and prepare a run authorisation document. This will detail the various tapes/disks required and how the output is (o he disposed of. This is then passed to the computer-room supervisor.

(f) Tape/disk library. The librarian will provide all the .tape reels and disk packs required for the particular job and will pass them to the computer room for the computer to process.

(g) On completion of the computer run/s the tapes/disks, etc, go back to the library and all the documentation goes to the data control section.

The control section now scrutinises the control log to ensure all action has been taken correctly and to initiate any possible corrective action indicated. Control totals will be reconciled. These will now include totals of items rejected on the run/s, but all totals will need to be reconciled back to those compiled ;it the beginning of the job.

The output is dealt with; invoices, for example, will be despatched to customers and warehouses, etc.

All necessary information will be fed back to the user output despatched, error lists, control totals.

DATA – PREPARATION STAFF

These staff are responsible for:

(a) Preparing floppy disks from source documents or preparation by other means, e.g. on-line data entry.

(b) Operation of ancillary machines such as floppy disk-units, paper bursters, etc.

WEEK 14

OBJECTIVES:

The students are expected to learn the following:

- **System evaluation procedures:**
- **Sost/benefits,**
- **Quality assurance,**
- **Control environment,**
- **Validation/ error control,**
- **Computer procedure and file control,**
- **Output procedures,**
- **Fullback and recovery,**
- **External requirement and recommendation.**

SYSTEM EVALUATION

The investigation should start by making contact with the manager of the user departments, not only to deal with the normal formalities but, in particular, to establish:

- whether or not the manager is satisfied with the performance of the system, and if not. what are the reasons;
- the use being made of the output reports; whether they are accurate: if they are timely; whether they contain insufficient or unwanted information;
- the operational aspects: whether the procedures are causing problems, and if any changes have been made;
- effectiveness: if there are many error reports; whether there are inaccuracies not being reported: turn-round and response times; the level of equipment utilization, reliability and service;
- changes in volumes of data, information, paper handling and their effect on the system;
- amendment requests: whether they have been implemented correctly; whether there are any pending.

The above facts should also be established in more detail from other levels of user staff employing the procedures.

When the personal interviews have been concluded, the auditor should quantify actual performance to establish any deviations from the planned performance, together with explanations: this is the main objective of the audit. Deviations, which may be classed as avoidable, can arise from incorrect estimates, e.g.:

- clerical and computer procedure timings;
- data volumes and growth rates;
- starring levels;
- error rates.

COSTS/BENEFITS

Here the actual costs and benefits are compared with those planned, showing any deviation from expectations. The causes of any deviation of costs or benefits from those planned should be established and stated. These may arise from

- unplanned pay increases: ,
- extra staffing, retention of initial temporary staff, inaccurate estimates;
- changed methods of computer charging;
- inaccurate estimates of data volume and timing;
- authorised, or unauthorised changes to procedures and documents.

Deviations may be either advantageous or disadvantageous, and all details should be reported. An increased cost may, of course, produce a better service, perhaps giving higher value.

Other types of deviation are usually environmental, arising from operational, trading or statutory changes, e.g.:

- pay methods, accounting policy, new equipment and techniques;
- production and selling methods;
- product and market standardisation or diversification;
- organizational extension or contraction;
- government statutory returns, new taxes.

Where changes already have been made to the-system, these should be summarised together with the causes. A check should be made that these have been officially approved and have followed the correct amending procedures, particularly that the changes have been recorded on the master system file and in the other appropriate reference manuals.

The performance statistics should first show the comparison of actual with planned, and only then should the effect on these of any amendments and improvements be shown. These comparison records can be fed back to the planning and estimating section, to system analysts and programmers, to improve future forecasting methods.

Management requires to know of any benefits which have not been realised, and the causes. To enable a realistic comparison to be made, the criteria used initially as a basis for estimating the financial benefits should be employed. Where these are found to>e ineffective in any way, reasons should be given, and alternative criteria then applied to both the estimates and the actual evaluation. This is particularly applicable to the criteria used to evaluate intangible benefits. Any additional benefits arising which were not expected should be noted; and any financial gains arising from changes made since the changeover. There should also be comment on the likelihood, of attaining any long-term benefits.

Quality Assurance

The level of control within the system deserves special attention. It is essential that adequate control procedures are built into the system as it is designed. These should be checked to ensure that they are working effectively, one being maintained, and that the system is secure. The following checklist gives a summary of the questions which should be asked.

CONTROL ENVIRONMENT

- is there clear segregation of control responsibilities?
- have all mandatory controls been specified?
- what standby procedures are there, and what is the cost of having to use them?

- can user involvement be adequately demonstrated?
- how will the user monitor system operation?
- are there procedures of authorisation to check the quality of data?
- are documentation standards maintained?

Source data collection

- have batch sizes been defined and maintained for maximum control?
- have batch control records been defined and maintained?
- are batch controls established as soon as possible?
- is the data collection environment suitable?
- are data collection and verification procedures clearly specified?
- are the following procedures defined and maintained correctly:
  - registration of receipts?
  - verification of receipts?
  - recording of work distribution?
  - data conversion controls?
  - error procedures,?.
  - procedures for special equipment?

Validation

- is all input verified to the required standard before processing?
- are all fields validated for range, format and size?
- are check-digits used where appropriate?
- are input records verified for completeness, content and field sequence?
- has sufficient use been made of possible field comparison tests: consistency, credibility, cross-checking, acceptability?

Error control

- are control reports adequate both for errors and successful runs?
- are any errors processed with acceptable data, and if so are suitable safeguards included?

- do error-override facilities exist, and are there special controls to prevent their misuse?
- has the most appropriate method been used for clearing a validation run?

COMPUTER PROCEDURE AND FILE CONTROLS

- does the program design include contingencies for overflow, timing and frequency variations, environment changes, and variations in machine conditions?
- are file controls adequate in the form of labels or special control records?
- are control totals kept for all significant fields, plus record counts and hash totals?
- could separate control tiles be usefully kept?
- is there a full reconciliation system linking input data to all output?
- is there appropriate provisioning of an audit trail?

Output procedures

- do output programs validate new fields created for output purposes?
- are key fields rechecked for credibility?
- are fields and reports edited according to standard?
- are full control reports provided?
- are reconciliations reported whether successful or not, with supporting control totals?
- are program performance figures reported?
- are output control reports produced?
- is all output approved and monitored by an output control section?
- is there an output control register?
- are control totals verified by output control section?
- are special procedures defined for confidential data?
- does the user receive a control report showing costs, volumes, and error-rates?
- are periodic in-depth checks performed on output data?

Use of terminals

- does terminal dialogue have built-in redundancy for error detection?
- is clerical data input minimised by the use of machine-generated data, or avoidance of fixed data keying?
- is input distinguished from output?
- are there adequate message controls (serial numbers, logging, hard copy, audit trail)?
- is the terminal design best for this application in terms of keyboard, screen format, security, etc?
- what data protection facilities are provided for data transmission?
- are error correction facilities provided through retransmission, reconstruction or using the principle of no acknowledgement?

FULLBACK AND RECOVERY

- are fallback clerical input procedures defined?
- is there a specified procedure for re-establishing controls?
- are all messages logged on receipt, and are these logs retrievable for recovery purposes?

External requirements

- have all appropriate external authorities been consulted?
- has the auditor approved the system controls?

Recommendations

Ways to improve system performance should be given: either to meet or exceed expectations. If additional work is needed, then the terms of reference should be formulated in detail.

WEEK 15

OBJECTIVES:

The students are expected to learn the following:

- **Maintenance and review concept:**
- **Amendment procedures,**
- **Maintenance group,**
- **System audit**

MAINTENANCE AND REVIEW

Provision must be made for environmental changes which may affect either the computer or other parts of computer-based systems: such activity is normally called 'maintenance'. It includes both the improvement of system functions and the correction of faults which arise during the operation of a system.

Maintenance activity may require the continuing involvement of a large proportion of computer department resources. For computer installations which have already developed the basic applications for the organisation, the main task may be to adapt existing systems in a changing environment. Perhaps a better term to describe this activity is ‘system evolution'. All systems are dynamic and subject to constantly changing requirements. Effort must be devoted to adapting them, and design should be flexibly specified so that such changes can be easily implemented.

Most changes arise in two ways:

- As part of the normal running of the system when errors are found, users ask for improvements, or external requirements change:
- As a result of a specific investigation and review of the system's performance.

This looks first at amendment procedures and then at formal system reviews.

AMENDMENT PROCEDURES

Systems should not be changed casually following informal requests. Changes in one area may affect others.

The Amendment Log, prepared initially with the Amendment Notification, records the estimated and actual completion of each of the activities defined by the author.

The Amendment List provides a permanent record of all the amendments made to any of the documents within a documentation file: it is a useful reference to keep at the end of each file. The serial number is the amendment number entered on the corresponding Amendment Notification. The original issue of any document can be identified by the letter 'A'; each time a document is replaced by an amendment the issue letter is changed sequentially, in the series B to Z, AA to ZZ.

MAINTENANCE GROUP

Responsibility for the maintenance of a particular system must be allocated before any requirement for changes arises. It is unwise for this responsibility to rest with the original designer after 'changeover'. If the number of amendments justifies a separate maintenance team, then this team usually reports either to the operations manager or to the programming manager. The maintenance team should be allowed to influence the original design of system,/programs insofar as it will affect their 'maintainability'.

SYSTEMS AUDIT

The systems audit is an investigation to review the performance of an operational system: to compare actual with planned performance; to verify that the stated objectives of the system are still valid in the present environment; and to evaluate the achievement of these objectives.

This investigation and evaluation may be carried out: by a systems analyst, preferably one who was not responsible for the original design; by representatives of users, computer operations, or internal auditors; or by a team composed of these representatives. A knowledge of systems design is essential for analysis of findings.

The initial review should take place when the system has had time to settle down, when any additional assistance by systems analysts and temporary staff is no longer required, when both equipment and people are operating satisfactorily, and before any major changes are made to the original design specification. This is unlikely to be less than three months after changeover. The initial systems audit provides the opportunity to check whether the objectives and benefits forecast in the feasibility study have been achieved. Subsequent audits, carried out as part of regular reviews of systems (perhaps annually) will be concerned with the continued achievement of benefits, any deviations from the master system specification, and opportunities for improvement.

The detailed tasks to be carried for this investigation are based on a checklist of the contents required for the Systems Audit Report. They are summarised under two main headings:

- system performance;
- cost benefit.

It must be emphasized that the over-riding reason for an audit is to verify that the stated objectives of the system are being achieved, or that they are still valid in the present environment. These objectives (for which management authorized the use of resources in the first place) must be established before attempting to evaluate the system performance. The objectives and cost benefits will be found in the management report of the feasibility study, and in subsequent reports. The expected performance of the system, in broad terms, should also b found there, but the System Specification should be referred to for details.

Review and Maintenance

Once the system has become operational it will need to be examined to see if it has met its objectives. For example, the costs and benefits will be compared with the estimates produced at the system's inception. This particular activity is often known as "post-audit".

The system will also need to be reviewed and maintained periodically for the following reasons:

(a) To deal with unforeseen problems arising in operation, e.g. programs may need to be modified to deal with unforeseen circumstance.

(b) To confirm that the planned objectives are being met and to take action if they are not.

(c) To ensure that the system is able to cope with the changing requirements of business.